



UNIVERSIDAD DE CASTILLA-LA MANCHA

Departamento de Sistemas Informáticos

**Modelo de Proceso para el Desarrollo de Interfaces
en Entornos CSCW Centrado en los Usuarios y
Dirigido por Tareas**

*Task-Oriented and User-Centred Process Model for Developing Interfaces
for Human-Computer-Human Environments*

Tesis Doctoral

Víctor Manuel Ruiz Penichet

Directores:
Dra. D^a. María Dolores Lozano Pérez
Dr. D. José Antonio Gallud Lázaro

Noviembre, 2007

ÍNDICE GENERAL

Índice General	iii
Índice de Figuras	ix
Índice de Tablas.....	xiii
Dedicatoria	xv
Agradecimientos	xvii
Capítulo 1. Introducción	1
1.1 Motivación.....	3
1.2 Objetivos	4
1.3 Organización del Documento.....	4
Capítulo 2. Origen y Estado del Arte de los Entornos CSCW	7
2.1 Origen	9
2.2 Definiciones.....	11
2.3 Los Fundamentos del CSCW.....	13
2.4 Clasificaciones	13
2.4.1 Primeras Clasificaciones	13
2.4.2 Problemática Actual	17
2.4.3 Solución Propuesta	18
2.4.4 Ejemplos de Clasificación	20
2.4.5 Análisis de la Clasificación	21
2.5 Ventajas e Inconvenientes de estos Sistemas	22
2.6 Investigación en CSCW. Aplicaciones y Herramientas	24
2.6.1 Investigaciones	24
2.6.2 Coordinación.....	29

2.6.3	Colaboración	32
2.6.4	Herramientas	35
2.6.5	Metodologías Colaborativas	37

Capítulo 3. Conceptos Fundamentales en Entornos CSCW39

3.1	Definiciones Terminológicas: la Necesidad de un Lenguaje Común.....	41
3.1.1	Taxonomías	41
3.1.2	Ontologías	42
3.1.3	Folcsonomías	43
3.1.4	Microformatos	44
3.1.5	Un Modelo Conceptual del Groupware en 1994.....	46
3.2	Estructuras Organizativas y Grupos de Usuarios.....	49
3.3	Colaboración, Cooperación, Comunicación y Coordinación	54
3.4	Características Espacio-Temporales de los Sistemas Groupware	56
3.5	Tareas y Tareas Cooperativas	57
3.5.1	Técnicas tradicionales del análisis de tareas en sistemas centrados en el usuario	58
3.5.2	ConcurTaskTrees (CTT)	60
3.5.3	Collaboration Usability Analisis (CUA)	63
3.5.4	Groupware Task Analysis (GTA)	68
3.5.5	Tareas y Coordinación: un Estrecho Vínculo.....	69
3.5.6	Algunos Conceptos Descritos en la WfMC	70
3.5.7	El Concepto de Tarea en un Estudio Típico de Workflow.....	73
3.5.8	Otras Definiciones en torno a Tareas	75
3.6	Awareness y Contexto Compartido.....	76
3.7	Diseño de Interfaces de Usuario Basado en Modelos	79
3.7.1	Enumeración de Trabajos para el Diseño de Interfaces de Usuario	80
3.7.2	De Objetos de Interacción Abstractos a Facetas	84
3.8	Implementación de Interfaces de Usuario	92
3.9	Metodologías para el Desarrollo de Entornos Colaborativos.....	94
3.9.1	AMENITIES	95
3.9.2	CIAM: una Propuesta Metodológica para el Desarrollo de la Interfaz de Usuario en Sistemas Groupware	99

Capítulo 4. Modelo Conceptual para Entornos CSCW..... 107

4.1	Una Ontología como Solución a la Diversidad de Vocabulario Empleado en el Desarrollo de Aplicaciones Groupware	109
4.1.1	Planteamiento del Problema	110
4.1.2	Selección de una Solución	110
4.1.3	Descripción de la Solución Propuesta	112
4.2	Un Modelo Conceptual para la Especificación de Entornos Colaborativos: Descripción de los Modelos	113
4.2.1	Modelo de Organización	114
4.2.2	Modelo de Tareas	135
4.2.3	Modelo de Objetivos.....	151
4.2.4	Modelo de Sesión.....	156

**Capítulo 5. TOUCHE: Task-Oriented and User-Centred
Process Model for Developing Interfaces for Human-
Computer-Human Environments..... 157**

5.1	Descripción general	159
5.2	Etapa 1.- Elicitación de Requisitos.....	160
5.2.1	Definición de Requisito	161
5.2.2	Modelo de Proceso de Ingeniería de Requisitos.....	162
5.2.3	Técnicas de Elicitación de Requisitos.....	162
5.2.4	Tareas en la Elicitación de Requisitos.....	164
5.2.5	Documento de Requisitos del Sistema (DRS)	166
5.2.6	Consideraciones para la Etapa de Elicitación de Requisitos en esta Metodología	167
5.2.7	Elicitación de Requisitos en Entornos CSCW	180
5.3	Etapa 2.- Análisis	184
5.3.1	Análisis del Sistema en Entornos CSCW	184
5.3.2	Identificación de Roles y Tareas	186
5.3.3	Estructura y Comportamiento	192
5.3.4	Diagramas Estructurales	193
5.3.5	Diagramas de Comportamiento	195
5.3.6	Trazabilidad Inter- e Intraetapa	196
5.4	Etapa 3.- Diseño	198
5.4.1	Diseño de la Interfaz de Usuario en Entornos CSCW.....	200
5.4.2	Metodología del Diseño.....	201
5.4.3	Modelo de Navegación	204
5.4.4	Modelo de Presentación	205
5.4.5	Trazabilidad Inter- e Intraetapa	206
5.5	Etapa 4.- Implementación	206
5.5.1	Concrete User Interface y Final User Interface	207
5.5.2	El Problema de la Falta de Estándares en los Elementos para las Aplicaciones Groupware	210
5.5.3	CIOs en los Sistemas Groupware	211
5.5.4	Elementos de los FUIs en los Sistemas Groupware.....	217
5.5.5	Trazabilidad	218

Capítulo 6. Caso de Estudio 219

6.1	Descripción del problema.....	221
6.2	Elicitación de requisitos	221
6.2.1	Adquisición del Conocimiento del Dominio del Problema	221
6.2.2	Identificar la Estructura Organizativa y los Actores del Sistema	222
6.2.3	Definir los Objetivos del Sistema.....	224
6.2.4	Elicitar los Requisitos.....	228
6.2.5	Priorizar Objetivos y Requisitos.....	232
6.3	Análisis	232
6.3.1	Identificación y Descripción de Roles	232
6.3.2	Identificación y Descripción de Tareas.....	234
6.3.3	Estructura	236
6.3.4	Comportamiento.....	237
6.3.5	Trazabilidad	239

6.4	Diseño	240
6.5	Implementación	245

Capítulo 7. TOUCHE CASE Tool: Una Herramienta CASE de Soporte a la Metodología247

7.1	Introducción	249
7.2	Herramientas CASE Actuales	249
7.3	Una Aproximación al Desarrollo mediante Herramientas Cooperativas: CE4Web.....	250
7.3.1	Arquitectura de CE4Web	252
7.3.2	Interfaz de Usuario	253
7.4	Implementación de TOUCHE CASE Tool	253
7.4.1	Modelo de Dominio	255
7.4.2	Elicitación de Requisitos.....	256
7.4.3	Análisis	261
7.4.4	La Vista.....	261
7.4.5	Conclusiones.....	264
7.5	Funcionamiento de TOUCHE CASE Tool.....	264
7.5.1	Creación de un Proyecto	264
7.5.2	Etapa de Elicitación de Requisitos.....	265
7.5.3	Análisis, Diseño e Implementación	274

Capítulo 8. Conclusiones, Aportaciones y Trabajos Futuros .279

8.1	Conclusiones	281
8.2	Aportaciones	282
8.3	Publicaciones	282
8.4	Trabajos Futuros	286

Referencias289

Anexo I: Descripción de los Metadatos de las Plantillas para Elicitación de Requisitos [Durán, 2000].....305

Plantilla general para requisitos.....	305
Extensión específica para requisitos de información.....	306
Extensión específica para requisitos funcionales.....	307
Extensión específica para requisitos no funcionales	308
Extensión CSCW de la plantilla general para requisitos	308
Plantilla para la Estructura Organizativa	308
Plantilla para Actores	308
Extensión específica para Grupos.....	309

Extensión específica para Individuos, Usuarios y Agentes.....309

Anexo 2: Actuaciones Experimentales311

Procedimiento de Sugerencias y Reclamaciones.....	311
Roles del sistema.....	314
Descripción del Workflow	314
Estados de los Comentarios.....	315
Bitácora de Proyectos	316
Descripción General de Requisitos	317
Elementos de memoria	319
Interfaz Web.....	324
Marcas de Tiempo.....	331
Ventajas del uso de la Bitácora de Proyectos.....	332

ÍNDICE DE FIGURAS

Figura 1.	Contextos de desarrollo e investigación en Estados Unidos para el CSCW y el groupware	10
Figura 2.	Esquema que diferencia el Trabajo Cooperativo Asistido por Computador (CSCW), del Trabajo Cooperativo (CW) y de los Agentes Cooperativos (CA)	13
Figura 3.	Matriz espacio/temporal de Johansen [Johansen, 1988].....	14
Figura 4.	Mapping de BSCW sobre la matriz de Johansen [Griffiths].....	15
Figura 5.	Matriz espacio/temporal de Johansen modificada por Grudin [Grudin, 1994] para elaborar una variante de clasificación de Groupware	15
Figura 6.	Tipos de herramientas de Groupware según Andriessen [Andriessen, 2003]	16
Figura 7.	Posibilidades según características CSCW	18
Figura 8.	Posibilidades espacio - temporales.....	19
Figura 9.	Clasificación no excluyente	19
Figura 10.	Ejemplos de funciones, herramientas y sistemas clasificados.....	20
Figura 11.	Clasificación según funcionalidad básica y original	21
Figura 12.	Clasificación según funcionalidad adicional posible	21
Figura 13.	Arquitecturas centralizada (a) y replicada (b).....	25
Figura 14.	Arquitectura híbrida	25
Figura 15.	AutoSpeakerID	26
Figura 16.	Ticket2Talk	27
Figura 17.	Aplicación CO2DE en la que están trabajando dos autores simultáneamente	28
Figura 18.	Relaciones entre la terminología básica de un Workflow.....	29
Figura 19.	Ejemplo de workflow como Red de Petri de alto nivel	30
Figura 20.	Notaciones especiales para construcciones comunes	31
Figura 21.	Clasificación de los Sistemas de Gestión del Conocimiento [Cobos, 2002].....	32
Figura 22.	Aspecto general de la intranet en un BSCW	34
Figura 23.	Vista de la página principal de un sistema Sharepoint Portal Server.....	34
Figura 24.	Ejemplo sobre la descripción en GOMS de la tarea de borrado de un archivo en DOS	59
Figura 25.	Diagrama cooperativo correspondiente a la tarea cooperativa de pintar en una pizarra virtual compartida en CTT	62
Figura 26.	Diagrama que representa las tareas que ha de realizar un usuario que desempeñe el rol 'painter' para pintar en una pizarra virtual compartida	62
Figura 27.	El método CUA	64
Figura 28.	Paso 1: Identificación de escenarios	64
Figura 29.	Paso 2: Los escenarios más interesantes / complejos se especifican textualmente con detalle.....	64
Figura 30.	Paso 3: Las tareas de los escenarios principales se representan en forma de diagramas	65
Figura 31.	Paso 4: Las tareas más importantes se describen por medio de diagramas (instancias de tareas y acciones)	66
Figura 32.	DUTCH method scheme	68

Figura 33.	The DUTCH design process.....	68
Figura 34.	Relaciones entre la terminología básica	70
Figura 35.	Relación entre <i>task</i> , <i>work item</i> , <i>case</i> y <i>activity</i>	74
Figura 36.	Representación de la relación de descomposición de tareas sin incluir el concepto de acción (a) e incluyéndolo (b) en [Molina, 2006]	76
Figura 37.	Marco de referencia de TADEUS	81
Figura 38.	Ejemplo de Diagrama de Interacción de Diálogos en IDEAS.....	86
Figura 39.	Ejemplo de Diagrama de Interacción de Diálogos en IDEAS.....	86
Figura 40.	Ejemplo de Diagrama de Especificación de Componentes en IDEAS	87
Figura 41.	Ejemplo de <i>Contrato</i> en IDEAS para poner en relación AIOs con los objetos del dominio.....	88
Figura 42.	Modelos de interfaz de usuario declarativos en UMLi	88
Figura 43.	(a) Ejemplo de diagrama de interfaz de usuario en UMLi en su versión preliminar. (b) Ejemplo del mismo diagrama refinado	89
Figura 44.	Detalle de la relación de los conceptos empleados en UsiXML en el modelado de la UI abstracta.....	91
Figura 45.	Marco de referencia Cameleon para el desarrollo de aplicaciones interactivas multi-contexto. Imagen obtenida de [Limbourg, 2004]	91
Figura 46.	Notación gráfica de IdealXML sobre objetos de UsiXML	92
Figura 47.	En Multimodal Teresa se pueden diseñar diferentes interfaces de usuario, para diferentes plataformas, a partir de un mismo modelo de tareas	93
Figura 48.	Una interfaz de usuario a partir de un esquema en UsiXML. Extraída de [Lepreux, 2006]	94
Figura 49.	Vistas conceptuales del modelo cooperativo de AMENITIES	96
Figura 50.	Etapas metodológicas de CIAM	103
Figura 51.	Relación entre modelos.....	114
Figura 52.	Metamodelo parcial de la <i>estructura organizativa</i>	116
Figura 53.	Notación empleada para representar el concepto de <i>rol</i>	117
Figura 54.	Metamodelo parcial del <i>rol</i>	118
Figura 55.	Notación empleada para representar el concepto de actor.....	119
Figura 56.	Metamodelo parcial del <i>actor</i>	120
Figura 57.	Notación empleada para representar el concepto de <i>grupo</i>	121
Figura 58.	Metamodelo parcial del <i>grupo</i>	121
Figura 59.	Notación empleada para representar el concepto de <i>individuo</i>	122
Figura 60.	Metamodelo parcial del <i>individuo</i>	122
Figura 61.	Notación empleada para representar el concepto de <i>usuario</i>	123
Figura 62.	Metamodelo parcial de <i>usuario</i>	123
Figura 63.	Notación empleada para representar el concepto de <i>agente</i>	124
Figura 64.	Metamodelo parcial de <i>agente</i>	124
Figura 65.	Metamodelo parcial de la <i>restricción</i>	125
Figura 66.	Metamodelo parcial de la <i>ley</i>	126
Figura 67.	Metamodelo parcial de la <i>capacidad</i>	127
Figura 68.	Esquema clasificadorio de las <i>relaciones organizativas</i> de un sistema.	128
Figura 69.	Notación empleada para representar el concepto de <i>relación de desempeño</i>	129
Figura 70.	Metamodelo parcial de la <i>relación de desempeño</i>	130
Figura 71.	Notación empleada para representar el concepto de <i>relación de agregación</i>	131
Figura 72.	Metamodelo parcial de la <i>relación de agregación</i>	131
Figura 73.	Notación empleada para representar el concepto de <i>relación de jerarquía</i>	132
Figura 74.	Metamodelo parcial de la <i>relación de jerarquía</i>	132
Figura 75.	Notación empleada para representar el concepto de <i>co-interacción</i>	134
Figura 76.	Metamodelo parcial de la <i>co-interacción</i>	134
Figura 77.	Metamodelo de Organización y sus relaciones con otros metamodelos	134
Figura 78.	Metamodelo parcial de <i>tarea</i>	137
Figura 79.	Notación empleada para representar el concepto de <i>tarea abstracta</i>	138
Figura 80.	Notación empleada para representar el concepto de <i>tarea de aplicación</i>	138
Figura 81.	Notación empleada para representar el concepto de <i>tarea de interacción</i>	139
Figura 82.	Notación empleada para representar el concepto de <i>tarea de usuario</i>	139
Figura 83.	Metamodelo parcial de <i>tarea compuesta</i>	141
Figura 84.	Metamodelo parcial de <i>Tarea Atómica</i>	142
Figura 85.	Metamodelo parcial de la <i>tarea de grupo</i>	144

Figura 86.	Posible metamodelo parcial del <i>Modelo de Objetivos</i> del sistema: primera opción.....	152
Figura 87.	Posible metamodelo parcial del <i>Modelo de Objetivos</i> del sistema: segunda opción	153
Figura 88.	Metamodelo parcial del <i>Modelo de Objetivos</i> del sistema.....	155
Figura 89.	Metamodelo parcial de Sesión.....	156
Figura 90.	Etapas de la Metodología	159
Figura 91.	Tareas de elicitación de requisitos en [Durán, 2000]	164
Figura 92.	Estructura del Documento de Requisitos del Sistema (DRS) a partir de [Durán, 2000]	166
Figura 93.	Pasos de la etapa de elicitación de requisitos	181
Figura 94.	Pasos de la etapa de Análisis	185
Figura 95.	Pasos en la identificación y descripción de roles.....	187
Figura 96.	Relación entre requisitos del sistema y tareas de usuario en el análisis	189
Figura 97.	Pasos en la identificación y descripción de tareas.....	189
Figura 98.	Elementos a tener en cuenta en los métodos HCI para el diseño de la interfaz de usuario de los sistemas CSCW.....	201
Figura 99.	Modelo conceptual basado en el modelo conceptual para interfaces de usuario abstractas de UsiXML para el modelado de la UI abstracta de los sistemas CSCW. Las nuevas inclusiones aparecen sombreadas.	202
Figura 100.	Pasos de la etapa de Diseño	204
Figura 101.	Pasos de la etapa de Implementación	207
Figura 102.	Ejemplos de Objeto de Interacción Concreto en XUL empleados en IDEAS....	208
Figura 103.	Modelo de presentación concreto para elementos del toolkit Java Swing en UMLi.....	208
Figura 104.	(a) Modelo CUI, (b) CUIs del tipo Graphical Container, (c) CUIs del tipo Graphical Individual Component, (d) CUIs del tipo auditivo	209
Figura 105.	Modelo conceptual de clasificación de CIOs [Limbourg, 2004].....	211
Figura 106.	CIO elemento compuesto en un sistema tipo meeting: cuadro de usuarios del sistema.....	213
Figura 107.	Modelo conceptual de clasificación de CIOs extendido de [Limbourg, 2004] para contemplar CIOs de aplicaciones groupware. Sombreada la propuesta añadida.	213
Figura 108.	Ejemplos de indicadores de acción extraídos de [Gutwin, 2004].....	215
Figura 109.	Ejemplos de CIOs como resultado de la reificación de AIOs caracterizados con la faceta <i>visibility</i> . Imagen extraída de [Gutwin, 2004]	216
Figura 110.	Captura de ejemplo de prototipo rápido creado con SDGT [Tse, 2004].....	217
Figura 111.	Diagrama de Casos de Uso del objetivo del sistema <i>OBJ-3: Elaboración sincrónica de documentos candidatos a ser publicados</i>	229
Figura 112.	Diagrama de la Estructura Organizativa del sistema del caso de estudio.....	236
Figura 113.	Tareas llevadas a cabo por los actores que juegan el rol <i>Writer</i> representadas por medio de un TD	237
Figura 114.	Diagrama de co-interacciones del sistema del caso de estudio	238
Figura 115.	Diagrama de Interacción de Contenedores Abstractos (ACID) del caso de estudio que representa la navegación por las distintas interfaces	241
Figura 116.	Diseño de la interfaz <i>Gestión Docs</i> del caso de estudio por medio del Diagrama de Interfaces de Usuario Abstractas (AUID).....	242
Figura 117.	Diseño de la interfaz <i>Ventana para la edición cooperativa</i> del caso de estudio por medio del Diagrama de Interfaces de Usuario Abstractas (AUID)	244
Figura 118.	Interfaz de usuario final para de la edición cooperativa de un documento. FUI a partir de la AUI <i>Ventana para la edición cooperativa</i>	245
Figura 119.	Captura de una sesión donde participan varios usuarios en CE4Web para elaborar un diagrama de clases.....	251
Figura 120.	Diagrama de paquetes del sistema	254
Figura 121.	Diagrama del modelo de dominio de la aplicación	254
Figura 122.	Diagrama que muestra la estructura y relación entre los elementos que componen la etapa de elicitación de requisitos en la aplicación	255
Figura 123.	Estructura de clases del requisito funcional.....	257
Figura 124.	Diagrama de clases que representa el modelo de la estructura organizativa de la aplicación.....	258
Figura 125.	Modelo que representa un <i>diagrama de casos de uso</i> y su estructura.....	259

Figura 126.	Estructura abstracta del documento de requisitos del sistema	260
Figura 127.	Modelo que representa los roles	261
Figura 128.	Modelo de los componentes principales de la interfaz gráfica	262
Figura 129.	Ejemplos del empleo del artefacto	262
Figura 130.	Representación para los diagramas de la aplicación	263
Figura 131.	Captura de la creación de un proyecto donde se pueden apreciar, en el menú de la izquierda, las cuatro etapas del modelo de proceso.....	265
Figura 132.	Captura de la pantalla principal en la <i>Elicitación de Requisitos</i>	266
Figura 133.	Insercción de recursos: información relacionada con el dominio del problema.....	266
Figura 134.	Despliegue en el árbol de la información relativa a la <i>estructura organizativa</i> del sistema	267
Figura 135.	Ejemplo de <i>plantilla</i> para un <i>actor</i> de tipo <i>grupo</i> . Nota: La figura está partida en dos para mejorar su visualización completa: la parte de la derecha es continuación de la de la izquierda	268
Figura 136.	Captura donde se muestra desplegada la información relativa a los objetivos del sistema. Concretamente está señalado el primero de ellos.....	269
Figura 137.	Captura de la información relativa a un objetivo, concretamente el primero de ellos.....	270
Figura 138.	<i>Diagrama de casos de uso</i> para el <i>objetivo</i> especificado generado de forma automática a partir de la información facilitada en los <i>requisitos funcionales</i> que se asocian con la <i>secuencia normal de ejecución</i> del mismo y la asociación de los <i>requisitos funcionales</i> asociados con el <i>objetivo</i>	271
Figura 139.	Árbol con los datos específicos para la <i>elicitación de requisitos</i>	272
Figura 140.	Captura de ejemplo de la descripción de un requisito funcional. Nota: La figura está partida en dos para mejorar su visualización completa: la parte de la derecha es continuación de la de la izquierda	273
Figura 141.	Matriz de rastreabilidad Objetivos-Requisitos	274
Figura 142.	Captura de la etapa de <i>análisis</i> en el árbol. Desplegado el <i>rol</i> "Reader"	275
Figura 143.	Captura con la información relativa a un rol. Nota: La figura está partida en dos para mejorar su visualización completa: la parte de la derecha es continuación de la de la izquierda.....	275
Figura 144.	Diagrama de tareas de un rol en CTT a falta de introducir las relaciones entre las tareas	276
Figura 145.	Diagrama de Interacción de Contenedores Abstractos (ACID) generado desde TOUCHE CASE Tool	277
Figura 146.	Diagrama de Interfaces de Usuario Abstractas (AUID) generado desde TOUCHE CASE Tool	277
Figura 147.	Captura del Procedimiento de Sugerencias y Reclamaciones en funcionamiento en la Web Municipal	312
Figura 148.	Flujo completo del Procedimiento de Sugerencias y Reclamaciones	313
Figura 149.	Ejemplo de organización de usuarios para un proyecto concreto	319
Figura 150.	Prototipo de interfaz para la introducción de un nuevo elemento de memoria.....	320
Figura 151.	Prototipo de interfaz para visualización de <i>elementos de memoria</i>	324
Figura 152.	Prototipo de interfaz para la generación de DDPs: Portada del DDP	326
Figura 153.	Prototipo de interfaz para la generación de DDPs: Secciones del DDP	327

ÍNDICE DE TABLAS

Tabla 1.	Conceptos de workflow y análisis de tareas.....	53
Tabla 2.	Ejemplo de tabla que describe la tarea de enviar un archivo a la papelera de reciclaje en UAN.....	59
Tabla 3.	Operadores temporales utilizados en CTT	61
Tabla 4.	Tipos de tarea en CTT.....	61
Tabla 5.	Mecanismos de colaboración en CUA	67
Tabla 6.	Elementos del workspace awareness.....	77
Tabla 7.	Elementos del workspace awareness relacionados con el pasado.....	78
Tabla 8.	Tipos de Objeto de Interacción Abstracto y ejemplos de elementos	84
Tabla 9.	Conceptos definidos en el marco conceptual de AMENITIES.....	99
Tabla 10.	Definición de conceptos de la <i>Vista Organizacional</i> en [Molina, 2006]	100
Tabla 11.	Definición de conceptos de la <i>Vista de Inter-Acción</i> en [Molina, 2006]	101
Tabla 12.	Definición de conceptos de la <i>Vista de Datos</i> en [Molina, 2006]	102
Tabla 13.	Definición de conceptos de la <i>Vista de Interacción</i> en [Molina, 2006].....	102
Tabla 14.	Etapas de CIAM y las técnicas empleadas en cada una	104
Tabla 15.	Ventajas e inconvenientes en técnicas para la categorización y/o definición de conceptos.....	109
Tabla 16.	Ejemplo de matriz de rastreabilidad [Durán, 2000]	167
Tabla 17.	Plantilla general.....	173
Tabla 18.	Extensión específica para objetivos.....	173
Tabla 19.	Extensión específica para requisitos de información	174
Tabla 20.	Extensión específica para requisitos funcionales	174
Tabla 21.	Extensión específica para requisitos no funcionales	174
Tabla 22.	Extensión CSCW de la plantilla general	175
Tabla 23.	Plantilla para la Estructura Organizativa de los participantes del sistema	177
Tabla 24.	Plantilla para <i>Actores</i>	178
Tabla 25.	Extensión específica para Grupos	179
Tabla 26.	Modelo para la extensión específica para <i>individuos, usuarios y agentes</i>	180
Tabla 27.	Plantilla para la descripción de los Roles del sistema: modelo organizativo	188
Tabla 28.	Plantilla para la descripción de las Tareas del sistema: modelo de tareas	190
Tabla 29.	Elementos organizativos	193
Tabla 30.	Relaciones entre los elementos organizativos	194
Tabla 31.	Esquema de matriz actores-roles	197
Tabla 32.	Esquema de matriz tareas-requisitos	197
Tabla 33.	Esquema de matriz tareas-roles	198
Tabla 34.	Notación para los elementos en el diseño de la UI de aplicaciones groupware	203
Tabla 35.	Descripción de la Estructura Organizativa de los participantes del sistema del caso de estudio	222
Tabla 36.	Descripción del usuario Author a partir de la plantilla para actores	223
Tabla 37.	Descripción del grupo AUTHORS a partir de la plantilla para actores y su extensión para grupos	224

Tabla 38.	Descripción del objetivo “elaboración síncrona de documentos candidatos a ser publicados” a partir de la plantilla para objetivos	225
Tabla 39.	Descripción del objetivo “revisión de documentos candidatos” a partir de la plantilla para objetivos	226
Tabla 40.	Descripción del requisito funcional “Editar documento” a partir de la plantilla para requisitos.....	230
Tabla 41.	Descripción del requisito no funcional “Percepción en la edición” a partir de la plantilla para requisitos	231
Tabla 42.	Descripción del rol Writer a partir de la plantilla para la descripción de los Roles del sistema.....	234
Tabla 43.	Descripción de la tarea <i>Editar_doc</i> a partir de la plantilla para la descripción de las tareas del sistema	235
Tabla 44.	Trazabilidad entre actores-roles	239
Tabla 45.	Trazabilidad entre tareas-requisitos para el objetivo <i>OBJ-3: Elaboración síncrona de documentos candidatos a ser publicados</i>	239
Tabla 46.	Trazabilidad entre tareas-roles.....	240

DEDICATORIA

*A mi familia,
que me ha ayudado constantemente a superar los
momentos más difíciles de estos últimos años.
Siempre han estado ahí de manera incondicional,
confiando en mí, apoyándome y dándome lo que
saben dar mejor que nadie: su cariño.*

AGRADECIMIENTOS

A veces el lenguaje se queda corto para expresar cosas que uno siente. Este es uno de esos momentos. Es mucho tiempo, mucha dedicación y muchas las personas que han estado ahí. Es un tópico decir que si trato de nombrar a alguien seguro que me dejo a otros que también debiera nombrar. Como casi siempre, los tópicos son verdad.

En primer lugar me gustaría mencionar a mis directores de tesis, José Antonio y María. Nuestra relación cruza la línea de la profesionalidad y hace mucho tiempo que son más amigos que compañeros. Han sabido estar y llenar mi lado personal.

El trabajo lo he dedicado a mi familia: mis padres, mis hermanos y María; pero no puedo dejar pasar la oportunidad de agradecerles todo lo que hacen por mí en estas tres líneas.

Richard, Fernando Andrés, Pascual, Miguel y los miembros del grupo LoUISE, cada uno desde su papel como compañero de viaje ha contribuido de alguna manera a que llegara al final.

Otros muchos han aportado desde su opinión profesional hasta su apoyo moral: Bárbara Leporini y los demás miembros del laboratorio HIIS, Paula González y demás miembros del grupo GRIHO, José Luis Garrido y demás miembros del grupo GEDES, M^a José Aguilar de la UCLM y mis compañeros de doctorado con los que he podido compartir batallitas.

A mi grupo de Montañeros Santo Ángel y a mis amigos, que hacen sentirme más persona y me ayudan a poner los pies en la tierra y a ordenar mis prioridades.

Gracias a todo aquel que se cruzó en mi camino y, consciente o inconscientemente me ayudó.

Modelo de Proceso para el Desarrollo de Interfaces en Entornos CSCW Centrado en los Usuarios y Dirigido por Tareas

Noviembre 2007

Víctor Manuel Ruiz Penichet

CAPÍTULO 1. INTRODUCCIÓN

Las sociedades modernas se distinguen por la velocidad de crecimiento en sectores altamente sofisticados como el de las tecnologías de la información y las comunicaciones. Dicho crecimiento se expresa en términos económicos como el impacto del sector tecnológico sobre el producto interior bruto (PIB) de un país, aunque también es posible experimentarlo en el día a día de esas sociedades, donde la tecnología es protagonista principal.

Cada progreso tecnológico tiene repercusión casi inmediata tanto en el modo de trabajar como en la forma de divertirse, en todas las facetas de la vida, de los miembros de esas sociedades avanzadas.

Dentro de las tecnologías de la información y gracias al progreso de las infraestructuras de comunicaciones, los sistemas software han ido aumentando su complejidad para satisfacer las demandas no ya de usuarios individuales sino de grupos de trabajo. Este es el contexto en el que los sistemas cooperativos soportados por ordenador (CSCW) tienen una relevancia mayor y su diseño y desarrollo son tareas que merecen atención.

El objetivo principal que persigue la tesis es la definición precisa de un modelo de proceso y una metodología que faciliten el desarrollo de sistemas CSCW atendiendo explícitamente a las características específicas de este tipo de sistemas.

En este capítulo de introducción se describe brevemente el trabajo realizado. Se muestra la motivación, los objetivos y la organización del documento.

1.1 Motivación

Los sistemas informáticos han ido evolucionando de manera que, en un corto periodo de tiempo, se ha pasado de hablar del PC u Ordenador Personal a las comunidades software. Es decir, la tecnología ha avanzado cubriendo las necesidades de las personas que son fundamentalmente las de comunicarse o colaborar entre ellas para alcanzar un objetivo común.

En un principio la tecnología resolvía problemas que se planteaban personas de forma individual. Así surgieron procesadores de texto, hojas de cálculo, etc. Pronto, el matiz que introduce el concepto de grupo de personas realizando algo en común a través de las redes de computadores se convierte en el centro del desarrollo. Son numerosas las aplicaciones basadas en comunidades de usuarios que realizan tareas en conjunto por medio de Internet y su éxito es indiscutible.

El premio Nobel de medicina de 1959, Severo Ochoa, decía que "*en principio la investigación necesita más cabezas que medios*". El CSCW o *Computer-Supported Cooperative Work* es un área de investigación relativamente nueva dentro del mundo de las ciencias de la computación que surgió precisamente como resultado de la reunión de un buen número de "*cabezas pensantes*", expertos, de diferentes áreas de investigación.

El origen interdisciplinario del CSCW permite extender ideas sobre la teoría del conocimiento humano, filosofía de la mente, sociología, psicología social, psicología de grupos, antropología, etc., hacia el mundo de la computación. El objetivo es, por tanto, crear sistemas para grupos conociendo cómo se comportan los grupos, desarrollar sistemas para usuarios que no están solos, que forman parte de una organización, conociendo su comportamiento dentro de la organización.

La Ingeniería del Software debe guiar el desarrollo de estas aplicaciones de soporte para el trabajo en grupo (también conocido como *groupware*), cada vez más necesarias por la forma de trabajo que se impone en la sociedad, tal y como lo ha venido haciendo con todas las aplicaciones por medio de modelos de proceso y metodologías bien definidos. Sin embargo, las aplicaciones groupware tienen una serie de características especiales que, debidamente consideradas, pueden lograr un desarrollo de mayor calidad puesto que se tienen en cuenta las necesidades del usuario desde el principio, necesidades específicas propias de estos sistemas, y que hasta ahora no han sido debidamente cubiertas como son la colaboración, comunicación, coordinación, compartición de información, etc.

En esta tesis se propone un modelo de proceso y una metodología, centrados en el usuario y dirigido por tareas, para el desarrollo de aplicaciones groupware desde la elicitación de requisitos hasta su implementación, considerando las características y particularidades de estos sistemas desde el inicio. Para ello, previamente se ha realizado un estudio de los sistemas CSCW que, adicionalmente, ha dado como resultado un método para la clasificación de estas aplicaciones, solucionando inconvenientes en propuestas anteriores. El estudio de los conceptos específicos ha concluido con un modelo conceptual de términos y relaciones sobre el que se asienta el modelo de proceso propuesto.

Por último se muestra un caso de estudio de una aplicación elaborada siguiendo la metodología propuesta y la herramienta CASE desarrollada para dar soporte a este modelo de proceso.

1.2 Objetivos

El objetivo principal de la tesis es **la definición precisa de un modelo de proceso y la metodología a seguir en cada una de sus etapas, para el desarrollo de interfaces de usuario de sistemas CSCW centrado en los usuarios y dirigido por tareas**. Para ello se han definido una serie de objetivos específicos que se describen a continuación:

- Conocer el estado del arte de los sistemas CSCW: qué son, cuáles son sus fundamentos, qué tipo de herramientas existen, ventajas e inconvenientes del uso de estos sistemas.
- Conocer los conceptos espacio-temporales, conceptos de colaboración, comunicación, cooperación y coordinación. Estudiar el modelado de tareas, el concepto de awareness y el diseño de interfaces de usuario basado en modelos.
- Definir un modelo conceptual para emplear un vocabulario común y concretar el significado de los términos que se van a emplear a lo largo del proceso de desarrollo. Conocer y emplear los términos frecuentes y necesarios.
- Definir un modelo de proceso que permita desarrollar sistemas CSCW de calidad, centrándose en el usuario como parte de un grupo y cuidando especialmente la interfaz de usuario.
- Definir la metodología a emplear en cada etapa del modelo de proceso definido, incluyendo la definición de nuevos modelos que nos sirvan para capturar las características específicas necesarias a tener en cuenta en el desarrollo de sistemas colaborativos.
- Establecer la trazabilidad correspondiente entre los diferentes modelos definidos en distintos niveles de abstracción.
- Aplicar el proceso a un caso de estudio concreto que permita validar su utilidad y muestre un ejemplo de su uso.
- Construir el prototipo de una herramienta CASE que dé soporte al modelo de proceso y metodología desarrollados.

1.3 Organización del Documento

La memoria de Tesis Doctoral se organiza en ocho capítulos cuya estructura y contenidos se describen a continuación:

El **Capítulo 1**, capítulo actual, describe brevemente el trabajo realizado y contiene la motivación, los objetivos y la organización del documento.

El **Capítulo 2** habla acerca del origen del CSCW, de los estudios llevados a cabo por los diferentes investigadores en el área. Así mismo incluye la propuesta de un método para la clasificación de aplicaciones groupware.

El **Capítulo 3** describe algunos conceptos de los trabajos más relevantes relacionados con la terminología que se suele emplear en entornos colaborativos. Se concretiza el estado del arte genérico estudiado en el capítulo anterior.

El **Capítulo 4** presenta la definición del modelo conceptual utilizado como base para este trabajo y que recoge los términos de uso frecuente definidos y relacionados de acuerdo con lo que los investigadores de más prestigio han propuesto.

El **Capítulo 5** describe las etapas del modelo de proceso propuesto en esta tesis, así como la metodología seguida en cada una de ellas: TOUCHE: Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments. Las etapas del modelo de proceso son cuatro: *Elicitación de requisitos, Análisis, Diseño e Implementación*. Cuatro etapas clásicas en el desarrollo de aplicaciones desde la Ingeniería del Software, pero donde se ha prestado especial atención a los aspectos característicos de los sistemas CSCW.

El **Capítulo 6** describe un caso de estudio que proporciona un ejemplo de aplicación del modelo de proceso definido en la tesis, y que demuestra la aplicabilidad y reproducibilidad de la propuesta.

El **Capítulo 7** presenta la herramienta CASE implementada para dar soporte a la metodología, automatizando ciertas tareas que hacen más sencilla la especificación del sistema.

El **Capítulo 8** muestra las conclusiones, aportaciones, publicaciones y trabajos futuros de la tesis.

Por último las referencias empleadas en la tesis se agrupan ordenadas alfabéticamente en el apartado **Referencias** y se ha considerado la introducción de dos documentos **anexos**: el primero con una breve descripción de unos metadatos empleados en la etapa de elicitación de requisitos del modelo de proceso, puesto que han sido adoptados de otra tesis; y el segundo con la descripción de dos de las aplicaciones desarrolladas e implantadas durante la fase de estudio de los sistemas CSCW.

CAPÍTULO 2.

ORIGEN Y ESTADO DEL ARTE DE LOS ENTORNOS CSCW

El poeta británico Lord Byron solía decir que “*el mejor profeta del futuro es el pasado*”. En este capítulo se habla acerca de los orígenes del término CSCW, de lo que comentaban los primeros investigadores en este campo, investigadores que, debido a la corta historia de este campo de investigación, en su mayoría siguen en activo. También se repasarán algunos conceptos relacionados con CSCW. Son conceptos anteriores a su origen, pero que han sido absorbidos hasta formar parte de él y, por tanto, resulta importante considerarlos.

El estudio de estos sistemas ha propiciado el conocimiento necesario para abordar los objetivos de la tesis: un modelo de proceso y la metodología que permita el desarrollo de interfaces de usuario en sistemas CSCW.

Además, para llevar a cabo un estudio minucioso y más de cerca se han desarrollado algunas aplicaciones descritas en el Anexo II. Cabe destacar que estas herramientas no han sido sólo prototipos experimentales, sino que se han desplegado y están actualmente en explotación además de haber dado como resultado diferentes publicaciones internacionales.

Como resultado adicional se destaca la propuesta de un sistema de clasificación para estos sistemas que ya ha sido aceptada por la comunidad nacional e internacional como demuestran las publicaciones obtenidas [Penichet, 2005b; Penichet, 2007e].

2.1 Origen

Hacia 1984 tuvo mucho éxito un fenómeno conocido como *Office Automation*, es decir, lo que podríamos llamar el *pre-CSCW*, una primera aproximación del soporte para grupos. El *Office Automation*, que literalmente se traduce como "automatización de la oficina" nace a mediados de los setenta, cuando las minicomputadoras empiezan a dar un soporte para grupos algo más sofisticado e interactivo. Aplicaciones pensadas para ser empleadas por un único usuario como los procesadores de texto u hojas de cálculo tuvieron mucho éxito y Office Automation trató de extender esas aplicaciones a filosofías de trabajo en grupo.

A mediados de la década de los ochenta, Paul Cashman e Irene Greif [Greif, 1988] organizaron un workshop de profesionales de diferentes disciplinas interesados en cómo trabajaba la gente y en cómo la tecnología podría ayudarles. Es entonces cuando acuñaron el término "*Computer-Supported Cooperative Work*" para describir este interés común. Las primeras conferencias se dieron en Estados Unidos pero rápidamente se extendió a Europa y Asia, donde ya existían trabajos relacionados y un serio interés en el tema.

El término *groupware* ya se empleaba con anterioridad al CSCW por Peter y Trudy Johnson-Lenz, antes de 1984. Su definición original es "*intentional group processes plus software to support them*", es decir, software que soporta procesos de grupo. Apareció en 1981 en el artículo "*Consider the Groupware: Design and Group Process Impacts on Communication in the Electronic Medium*" [Johnson-Lenz, 1981].

Este término fue rápidamente adoptado por la comunidad CSCW. Así, el término *groupware* se emplea para hacer referencia a tecnologías comerciales, al software, mientras que el término CSCW hace referencia a la investigación que hay detrás de todo eso, a sistemas experimentales, a la naturaleza de los espacios de trabajo y las organizaciones. Es decir, *CSCW describe la investigación y el groupware la tecnología* [Grudin, 1994 (b)]. Como decía Greenberg [Greenberg, 1991], "*CSCW también puede verse como una disciplina científica que guía el diseño y el desarrollo de Groupware cuidadosa y apropiadamente*".

Según Douglas Engelbart, una eminencia del mundo de la informática e interesado en la colaboración asíncrona entre equipos de trabajo distribuidos geográficamente [Engelbart, 2004], se necesitaba saber cómo trabaja la gente en grupo, cómo funcionan las organizaciones y cómo afecta la tecnología a todo esto. CSCW empieza entonces como un esfuerzo por parte de los tecnólogos por aprender de *economistas, psicólogos sociales, antropólogos, teóricos organizacionales, educadores* y cualquiera que pudiera tener algo que ver con actividades de grupo.

Por un lado resulta altamente interesante el hecho de que CSCW surja del interés de investigadores provenientes de campos tan diversos como los mencionados anteriormente: psicología social, economía, antropología, etc. pero esto conlleva un problema muy importante, y es que en cada campo se habla un "idioma", es decir, se usa una terminología diferente y esto da lugar una verdadera "Torre de Babel".

No es lo mismo la palabra "usuario" en el campo del HCI (Human-Computer Interaction o Interacción Persona-Ordenador), la persona que está frente a los dispositivos, introduciendo información y comandos; que en el campo de los sistemas de información, donde el "usuario" podría ser una persona que ni siquiera toque una tecla.

En la Figura 1 se representan los contextos de desarrollo e investigación del CSCW y del *groupware* en los Estados Unidos [Grudin, 1994 (a)].

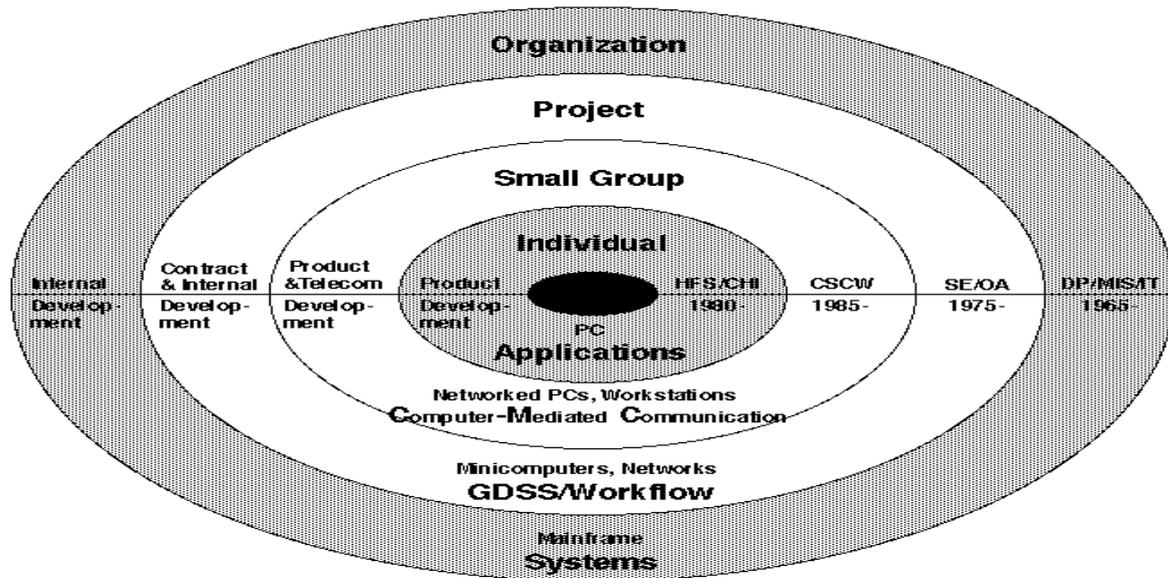


Figura 1. Contextos de desarrollo e investigación en Estados Unidos para el CSCW y el groupware

Hasta la primera década del CSCW, casi toda la actividad se centraba en los anillos que aparecen sombreados, en los sistemas mainframe y aplicaciones individuales.

El anillo exterior representa los grandes sistemas y aplicaciones para mainframe, grandes sistemas diseñados para cubrir las necesidades de las organizaciones: transacciones, control de inventario, fabricación controlada por computador, etc.

El anillo interior representa aplicaciones especialmente diseñadas para usuarios individuales, usuarios de PCs y estaciones de trabajo: procesadores de textos, debuggers, hojas de cálculo, juegos, etc.

Pero poco a poco la actividad se va centrando en proyectos y pequeños grupos.

El tercer anillo desde dentro representa a grandes proyectos incluyendo salas de reuniones electrónicas, sistemas de automatización de workflow, etc. aplicaciones en definitiva para grupos relativamente grandes, de 6 personas o más.

El segundo anillo desde dentro representa a pequeños grupos de usuarios, de no más de 3 ó 4 personas: aplicaciones de escritura colaborativas, etc.

En general se puede decir, y en particular para el caso de los Estados Unidos, que el software desarrollado en los anillos centrales no sombreados, es lo que se conoce como groupware.

En la parte izquierda de la Figura 1 se representan los contextos de desarrollo de software. Por un lado los *convenios y contratos con el Gobierno* estimulan el desarrollo del software a nivel de grandes proyectos (tercer anillo), y por otro lado, las *compañías de telecomunicaciones* y los *desarrolladores de productos comerciales*, empiezan a encontrar enormemente interesante el software que se podría enmarcar dentro del anillo segundo, para grupos pequeños.

En la parte derecha de la figura se representan las áreas de investigación asociadas al desarrollo y uso de los sistemas unidos a cada contexto de desarrollo. El campo de los Sistemas de Información (IS) se centra en la Organización, anillo exterior, pero también concierne a los grandes proyectos (*anillo tercero*). A mediados de los 70 emergen los campos de la Ingeniería del Software y la Automatización de la Oficina (Software Engineering, SE, y Office Automation, OA). Se centran en dar soporte a grandes grupos y proyectos. Aunque OA no se mantuvo como un campo de investigación, muchos tópicos sobre sistemas de gestión de proyectos

distribuidos dieron lugar a lo que hoy se conoce como workflow, que también es anterior a CSCW.

HCI (Human-Computer Interaction o Interacción Persona-Ordenador) y CSCW tienen mucho en común, tanto es así, que CSCW proviene del campo de la interacción persona-ordenador. El cambio de las compañías de realizar desarrollos para personas individuales -single-user applications- a realizar sistemas para pequeños grupos -small-group applications-, provoca la aparición del CSCW como nueva filosofía que resurge a partir del HCI. La organización CHI ha patrocinado en algunas ocasiones, a partir de 1986, algunas conferencias CSCW.

En [Horn, 2004] se dice que CSCW surge como un campo de investigación debido a la especialización de los contextos de grupo y de organización en *human-computer interaction* (HCI). HCI es anterior a CSCW y es un campo en el que se ha estado investigando durante años sobre la interacción de las personas con los sistemas. Pero el estudio del comportamiento persona-ordenador-persona se hace cada vez más importante, tanto, que esta especialización es lo que da lugar al CSCW.

Las aplicaciones enmarcadas en los entornos CSCW permiten compartir experiencias, informar y permanecer informados, restringir accesos a información, realizar conferencias y videoconferencias, aplicaciones de autoría colaborativas, correo electrónico, salas de reuniones electrónicas, sistemas que den soporte a grupos, etc.

Y es que, con el paso del tiempo, los sistemas informáticos están pasando de ser sistemas individuales, que resolvían problemas de usuarios particulares, a ser sistemas ideados para resolver problemas de grupo. Sistemas que aplican teorías, técnicas y métodos desde la antropología y la sociología, hasta la psicología social o la filosofía [Grudin, 1994; Martin, 2004].

2.2 Definiciones

El filósofo griego Aristóteles dijo una vez que "*una definición es una frase que significa la esencia de una cosa*". Todos los investigadores tratan de buscar la esencia de aquello que investigan y cuando creen tenerlo claro intentan explicar lo que significa por medio de definiciones.

Una colección de definiciones, proporcionará un conjunto de ideas que giran, todas ellas, sobre la esencia troncal de esa "*cosa*", enriqueciendo finalmente la percepción global de la misma.

Así, se muestran a continuación un conjunto de definiciones acerca del *groupware* en primer lugar:

- Peter y Trudy Johnson-Lenz, 1978: "Un grupo de procesos calculados, más el software necesario para soportarlos".
- R. Johansen, 1988 [Johansen, 1988]: "Soporte por ordenador para equipos de trabajo".
- Doug Englebart, 1988: "Un sistema de desarrollo colaborativo para la interacción hombre-máquina".
- Clarence Ellis, 1991 [Ellis, 1991]: "Sistemas basados en computadoras para grupos de personas que realizan tareas comunes (o metas) y que proporcionan un interfaz para un entorno compartido".
- Beacker, 1993: "Es un software multiusuario que soporta el CSCW".

- David Coleman, 1997 [Coleman, 1997]: "Colaboración guiada por computador que incrementa la productividad o funcionalidad de los procesos persona-persona".
- Orlikowski y Hofman, 1997 [Orlikowski, 1997]: "La tecnología Groupware proporciona redes de computadores que soportan la comunicación, coordinación y colaboración a través de facilidades como el intercambio de información, la compartición de repositorios, los foros de discusión y la mensajería".

Podemos decir que groupware es una aplicación o conjunto de herramientas, un sistema al fin y al cabo, que cubre ciertas necesidades, como la comunicación, la cooperación y la coordinación, surgidas en el trabajo colaborativo de dos o más personas, estén o no próximos en el espacio o en el tiempo, proporcionando un interfaz común a un entorno compartido [Penichet, 2003].

Por otro lado, CSCW es toda la filosofía que enmarca a estos sistemas groupware, es su base. Los investigadores también han propuesto diferentes definiciones en torno al término CSCW:

- Greif, 1988 [Greif, 1988]: "CSCW ha surgido de la investigación sobre el rol del computador en el trabajo en grupo. Surgen preguntas acerca de cómo los grupos grandes y pequeños pueden colaborar usando la tecnología de los computadores: ¿Cómo deben las personas planificar el trabajo en conjunto para aprovecharse de este medio? ¿Qué tipo de software debe desarrollarse? ¿Cómo se definirá el trabajo en grupo para extraer el potencial de las personas y las tecnologías? Las respuestas se encontrarán en la investigación de disciplinas que incluyen a la informática, inteligencia artificial, psicología, sociología, teoría de la organización y antropología. CSCW es el marco de esta investigación interdisciplinaria".
- Bannon y Schmidt, 1989 [Bannon, 1989] definen CSCW como "un esfuerzo por comprender la naturaleza y características del trabajo cooperativo con el objetivo de diseñar tecnologías adecuadas basadas en computador".
- Ellis, 1991 [Ellis, 1991]: "CSCW es un campo nuevo y multidisciplinar, que utiliza la experiencia y colaboración de muchos especialistas, incluidos profesionales de la computación y de las ciencias sociales. CSCW observa cómo trabajan los grupos y cómo puede ayudarlos la tecnología a realizar mejor el trabajo".
- Greenberg, 1991 [Greenberg, 1991]: "CSCW también puede verse como una disciplina científica que guía el diseño y el desarrollo de Groupware cuidadosa y apropiadamente".
- Beacker, 1993: "El CSCW es la actividad coordinada y asistida por computador, tal como la comunicación y la resolución de problemas, llevada a cabo por un grupo de individuos que colaboran entre sí".

Tal y como comentaba el profesor Marcos Borges (U. Federal de Río de Janeiro) en una conferencia mantenida en 2004 en la Universidad de Castilla-La Mancha, conviene distinguir entre Trabajo Cooperativo (CW) y Trabajo Cooperativo Asistido por Computador (CSCW). El primero existe desde hace muchos años y ha sido ampliamente estudiado: interacción de grupos y otras teorías sobre comportamientos, etc. Sin embargo, CSCW es eso mismo pero asistido por computador. Y es que la tecnología cambia la forma de pensar de la gente, la forma de actuar, la forma de comunicarse. La participación de seres humanos en un sistema CSCW es fundamental y es lo que los diferencia de otros sistemas. En la Figura 2 se puede ver un esquema que representa esto mismo.

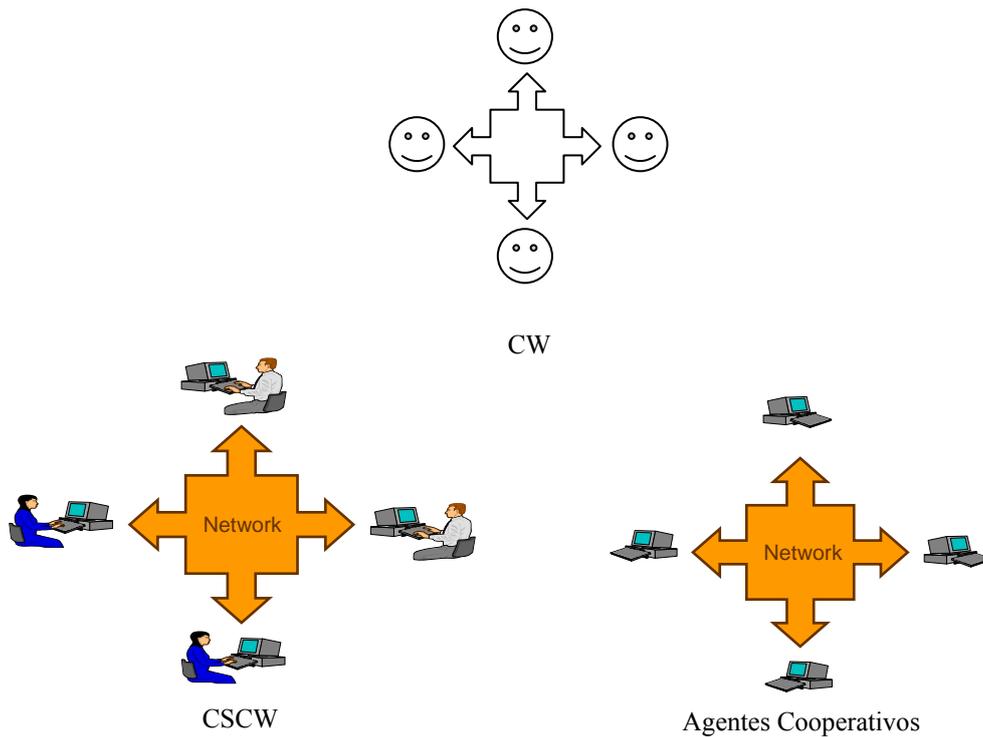


Figura 2. Esquema que diferencia el Trabajo Cooperativo Asistido por Computador (CSCW), del Trabajo Cooperativo (CW) y de los Agentes Cooperativos (CA)

2.3 Los Fundamentos del CSCW

Los conceptos de *comunicación, colaboración, cooperación, intercambio de información y coordinación* en cuanto a características propias de los sistemas CSCW se han considerado desde siempre como su fundamento.

A esto hay que añadir pequeños matices y discrepancias en el uso de la terminología.

Por su estrecha relación con los objetivos del siguiente capítulo, el desarrollo de este apartado se abordará en detalle en el apartado 3.3.

2.4 Clasificaciones

El uso de clasificaciones ha permitido ordenar y categorizar las diferentes aplicaciones según sus características, lo que las hace especialmente útiles a la hora de buscar una determinada aplicación que resuelva un problema concreto.

2.4.1 Primeras Clasificaciones

Las herramientas groupware se han clasificado hasta ahora de muchas maneras, pero lo cierto es que todas ellas se basan en una matriz inicial ideada por Johansen en el 88 (Figura 3) [Johansen, 1988].

2.4.1.1 Matriz Espacio-Temporal de Johansen

En esta primera clasificación se categorizan las herramientas según el *espacio* y el *tiempo*. La interacción persona-ordenador-persona se puede dar en el mismo espacio físico, como podría ser una sala de reuniones, el salón de una conferencia o un espacio de trabajo común; pero también se puede dar en la distancia y algunos ejemplos serían las salas de video conferencia, el uso de editores colaborativos o las pizarras compartidas. Así mismo, la dimensión temporal de estos sistemas facilita una distinción entre aquéllos en los que la interacción se da en tiempo real como la, tan de moda, telefonía IP, o los chats; y aquellos otros en los que el tiempo no juega un papel relevante, o al menos no en exceso, caso del correo electrónico, control de versiones, agendas, etc.

Por tanto, las herramientas se pueden clasificar de cuatro maneras: *síncronas / en el mismo espacio*, *síncronas / en espacios diferentes*, *asíncronas / en el mismo espacio* o *asíncronas / en espacios diferentes*. El correo electrónico sería una herramienta asíncrona y, en principio, destinada a personas situadas en espacios diferentes.

	Tiempo real	Tiempos diferentes
En el mismo sitio	Interacción cara a cara	Interacción asíncrona
En la distancia	Interacción síncrona distribuida	Interacción asíncrona distribuida

Figura 3. Matriz espacio/temporal de Johansen [Johansen, 1988]

Pero los sistemas se han ido complicando tanto que incluso la matriz se podría utilizar para poder clasificar la funcionalidad de un solo sistema groupware, entendiendo el sistema como un conjunto de herramientas groupware, ofreciendo cada una de ellas una funcionalidad que resuelve un determinado problema, y que permite clasificarla.

Pero los sistemas se han ido complicando tanto que incluso la matriz se podría utilizar para poder clasificar la funcionalidad de un solo sistema groupware, entendiendo el sistema como un conjunto de herramientas groupware, ofreciendo cada una de ellas una funcionalidad que resuelve un determinado problema, y que permite clasificarla.

Por ejemplo, el sistema de gestión del conocimiento BSCW [BSCW, 1995], una herramienta de trabajo colaborativo basado en Web desarrollada en Python, que permite a los diferentes miembros de un grupo cooperar de forma síncrona o asíncrona a través de Internet o desde una Intranet. Como gran sistema, BSCW resuelve varios problemas en torno a la gestión documental, problemas que podrían ser resueltos por una herramienta cada uno de ellos: versionado, check-in y check-out, publicación, etc. Podríamos tratar a BSCW como un sistema que aglutina varias herramientas groupware, y de este modo, su clasificación en la matriz no está clara.

La Figura 4 muestra la correspondencia o mapping de algunas funciones de BSCW con la matriz espacio temporal de Johansen [Griffiths].

2.4.1.2 Matriz Espacio-Temporal más Conciencia de Espacio y Tiempo

Otra clasificación, muy similar a la anterior, es la que presenta Grudin apoyándose en la matriz espacio/temporal de Johansen [Johansen, 1988] para elaborar una variante de clasificación de groupware (Figura 5) [Grudin, 1994; DeSanctis, 1987].

	Tiempo real	Tiempos diferentes
En el mismo sitio	El objeto "Meeting" asiste en la realización de reuniones cara a cara.	Utilizando el Monitor applet. Se puede integrar con sistemas de videoconferencia (por ejemplo CU-SeeMe).
En la distancia	Idem que a la derecha -->	Se almacenan todos los documentos en espacios de trabajo. Informa de modificaciones en el espacio de trabajo. Control de versiones. Soporte por email. Tablón de anuncios.

Figura 4. Mapping de BSCW sobre la matriz de Johansen [Griffiths]

		TIEMPO		
		mismo	diferente pero conocido	diferente y desconocido
LUGAR	mismo	Meeting facilitation	Work shifts	Team rooms
	diferente pero conocido	Teleconferencing Videoconferencing Desktop conferencing	Electronic mail	Collaborative writing
	diferente y desconocido	Interactive multicast seminars	Computer boards	Workflow

Figura 5. Matriz espacio/temporal de Johansen modificada por Grudin [Grudin, 1994] para elaborar una variante de clasificación de Groupware

De nuevo, la actividad se puede realizar en el mismo espacio o en espacios diferentes, y en el mismo tiempo o a lo largo del tiempo. La aportación está en si el usuario conoce o no esos espacios y/o tiempos que son diferentes. Es decir, se introduce una nueva característica según el conocimiento de los usuarios.

- Por su característica *espacial*, las aplicaciones groupware pueden ser aplicaciones que se realizan en el *mismo sitio* físico, orientadas a que todos los usuarios las utilicen en el mismo lugar de trabajo (primera fila). También es posible que se den en *lugares diferentes pero conocidos* por los participantes (segunda fila) o que se den en *lugares diferentes y además desconocidos* por los usuarios (tercera fila).

Por otro lado, según sus particularidades *temporales*, las aplicaciones groupware pueden ser en *tiempo real o síncronas* (primera columna), en *diferido o asíncronas* con *tiempos conocidos* por parte de los usuarios (segunda columna) o darse de forma *asíncrona o en tiempos diferentes y desconocidos* (tercera columna).

En la Figura 5 se muestra la matriz aumentada con algunas aplicaciones de ejemplo que puedan ser representativas para cada celda.

De nuevo, las aplicaciones no siempre se tienen por qué clasificar claramente en una de esas celdas, y en algunos casos, una aplicación puede clasificarse en una, y en otras en otra diferente. Esta clasificación puede definir para qué son esas aplicaciones pero tampoco es restrictiva.

2.4.1.3 Matriz de Andriessen

Erick Andriessen extiende todavía más la clasificación inicial de Johansen. Se basa en la combinación de la clasificación de ésta y la clasificación de los cinco posibles grupos de procesos en las ICTs (Information and Communications Technologies) [Andriessen, 2003]:

- Procesos de intercambio entre personas: la *comunicación*.
- Procesos orientados a las tareas: cooperación, coordinación y compartición de información y aprendizaje.
- Procesos orientados a grupos: *interacciones sociales*.

	Soporte entre comunicantes: Comunicación Asíncrona Sitios diferentes / tiempo diferente	Soporte para reuniones electrónicas síncronas Sitios diferentes / mismo tiempo	Soporte para reuniones cara a cara síncronas Mismo lugar / mismo tiempo
Sistemas de comunicación	Fax E-mail Video-mail Voz-mail	Teléfono, móvil Sistemas de audio Sistemas de vídeo Chats	
Sistemas de compartición de información	Sistemas de compartición de documentos Tableros de noticias digitales Foros	Sistemas de teleconsulta Co-Navegador	Sistemas de presentación
Sistemas de cooperación	Documentos co-autor Gestión documental Discusiones	CAD compartidos, pizarras compartidas, , etc.	Sistemas que dan soporte a decisiones de grupo
Sistemas de coordinación	Calendarios de grupo Planificación compartida Sistemas de gestión de workflow compartidos Gestor de eventos Agenda	Sistemas de notificación	Sistemas de soporte de control central y comandos
Sistemas de interacción social		Espacios media Realidad virtual	

Figura 6. Tipos de herramientas de Groupware según Andriessen [Andriessen, 2003]

Las *herramientas de comunicación* facilitan la comunicación entre personas separadas geográficamente y la hacen fácil, rápida y económica.

- Las *herramientas de compartición de información* y consulta facilitan almacenar y recuperar grandes cantidades de información de forma rápida, confiable y económica, a grupos de usuarios que pueden acceder a la misma incluso de forma remota.
- Las *herramientas de colaboración* son utilidades que mejoran el trabajo en grupo proporcionando documentos compartidos y herramientas de co-authoring (que permiten gestionar documentos de forma conjunta por parte de varios usuarios).
- Las *herramientas de coordinación* proporcionan coordinación a un grupo de trabajo distribuido por medio de sincronizadores, como por ejemplo calendarios de grupo o sistemas de gestión workflow.
- Por último, las herramientas de *interacción social*, que proporcionan la posibilidad de realizar reuniones virtuales por medio de cámaras, monitores y otros dispositivos para poder llevar a cabo esas reuniones de forma similar a como se harían físicamente.

Combinando la matriz espacio/temporal con los cinco tipos de proceso se obtiene la clasificación que se puede ver en la Figura 6, que incluye algunos ejemplos de herramienta ya clasificados.

2.4.2 Problemática Actual

Como se ha comentado con anterioridad y como se puede ver en la literatura, todas estas clasificaciones realizan una aproximación en la que se podrían encajar algunas herramientas. No siempre es sencillo poder hacerlo y con frecuencia no se sabe dónde ubicar una herramienta sencilla y menos aún un sistema complejo.

Si se habla de una aplicación sencilla, que tiene una funcionalidad concreta para resolver un problema determinado -como un chat-, la complejidad puede ser menor. Pero cuando es un sistema que implementa varias funciones, como podría ser un gestor documental que facilita herramientas, o funciones para el versionado, la compartición, la protección y desprotección, publicación, aprobación, etc. el problema de la clasificación del sistema es más serio.

¿Cómo clasificar un gestor documental? La función de versionado no está orientada a usuarios separados en espacio necesariamente. Incluso puede incluir un mecanismo de colaboración para la realización de los documentos en tiempo real. O, tener una agenda compartida para un grupo de usuarios, ¿implica que no estén trabajando codo con codo?

La siguiente pregunta que cabe hacerse es si esas clasificaciones que existen son válidas o, en caso contrario, no son acertadas.

Realmente las matrices de clasificación existentes hasta la fecha han resultado útiles, pero la complejidad de las herramientas y la elaboración de los grandes sistemas actuales han hecho que la idea quede en cierto modo obsoleta.

Estas taxonomías son adecuadas para herramientas de propósito determinado o más o menos sencillas, pero no para herramientas o sistemas más complejos.

2.4.3 Solución Propuesta

Para salvar esta situación se puede dar un giro en la idea de la clasificación. En lugar de intentar clasificar las herramientas según puedan encajar en esas matrices, se puede mostrar la relación de una función, o de una aplicación, o de un sistema, con las *características espacio-temporales* y con las *características propias de los sistemas CSCW* (*colaboración, comunicación y coordinación*) [Penichet, 2005b; Penichet, 2007].

De esta manera no se restringe la posibilidad de que una función, pero especialmente una aplicación o un sistema, puedan tener características a la vez.

2.4.3.1 Características propias del CSCW

Como *características propias de CSCW* se pueden dar hasta siete posibilidades, de la *a* a la *g*, como muestra la Figura 7. Lógicamente, la primera de las opciones no es válida, porque un sistema que no tenga al menos una de las tres, no es un sistema CSCW.

Tipo	Característica CSCW		
	Colaboración No = 0, Sí = 1	Comunicación No = 0, Sí = 1	Coordinación No = 0, Sí = 1
X	0	0	0
A	0	0	1
B	0	1	0
C	0	1	1
D	1	0	0
E	1	0	1
F	1	1	0
G	1	1	1

Figura 7. Posibilidades según características CSCW

2.4.3.2 Características Espacio-Temporales

Por otro lado, como características *espacio-temporales* se pueden dar hasta nueve posibilidades. Como una aplicación debe ser *síncrona, asíncrona* o *ambas* y lo mismo con el *espacio*, los estados en los que se de que no pueda ser síncrona ni asíncrona o que no se dé en el mismo espacio ni en espacio diferente, son *estados prohibidos* porque no se pueden dar. Los únicos estados posibles, como muestra la Figura 8, son: 5, 6, 7, 9, 10, 11, 13, 14 y 15.

2.4.3.3 Matriz de Clasificación Resultante

Por lo tanto, una aplicación podría clasificarse en la tabla que se muestra en la Figura 9, una *clasificación no excluyente*, que indica con claridad qué tipo de función, o herramienta, o sistema se trata.

Siguiendo estas mismas directrices, en lugar de utilizar la tabla, se puede emplear un código asociado, el *tipo general*, uniendo la letra que representa las características CSCW con el número de la propiedad *espacio-temporal*.

Tipo	Tiempo		Espacio	
	Síncrono No = 0, Sí = 1	Asíncrona No = 0, Sí = 1	Mismo No = 0, Sí = 1	Diferente No = 0, Sí = 1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Figura 8. Posibilidades espacio - temporales

Tipo	Herramienta	Característica CSCW			Tiempo / Espacio			
		Colaboración No = 0, Sí = 1	Comunicación No = 0, Sí = 1	Coordinación No = 0, Sí = 1	Síncrono N=0, S=1	Asíncrona N=0, S=1	Mismo N=0, S=1	Diferente N=0, S=1
B-9	Chat	0	1	0	1	0	0	1

Figura 9. Clasificación no excluyente

Por ejemplo, una aplicación que únicamente permitiera chatear sería del *tipo B-9*, puesto que no tiene nada que ver directamente con *colaboración* o con *coordinación*, sino con *comunicación*, y en cuanto a *características espacio-temporales*, queda completamente definido apuntando mediante el código 9 que es una *herramienta síncrona*, y *no asíncrona*, que se puede emplear en la *distancia*. También se podría emplear en el mismo lugar pero no tendría sentido, en principio.

Las preguntas que se deberían hacer para determinar a qué corresponde cada funcionalidad son las siguientes:

- ¿Se ayuda a que los usuarios puedan colaborar en la consecución de un objetivo? ¿Comparten información? ¿La trabajan?
- ¿Se puede utilizar como medio de comunicación? ¿Se les comunica algo a los usuarios? ¿Se informan por medio de esta herramienta?
- ¿Permite la coordinación de procesos, de personas?
- ¿La herramienta se utiliza en tiempo real?
- ¿Tiene sentido su uso en diferido?

- ¿Tiene sentido su uso en el mismo espacio físico?
- ¿La herramienta se puede emplear en espacios diferentes?

2.4.4 Ejemplos de Clasificación

En la Figura 10 se muestran algunos ejemplos, tanto de funcionalidades, como de herramientas, como de sistemas completos.

Tipo	Herramienta	Característica CSCW			Tiempo / Espacio			
		Colabora N=0, S=1	Comunica N=0, S=1	Coordina N=0, S=1	Sincrono N=0, S=1	Asíncrona N=0, S=1	Mismo N=0, S=1	Diferente N=0, S=1
B-5	Fax	0	1	0	0	1	0	1
B-5	E-mail	0	1	0	0	1	0	1
B-9	Telefonía IP	0	1	0	1	0	0	1
B-9	Chat	0	1	0	1	0	0	1
F-7	S. Gestión Documental	1	1	0	0	1	1	1
F-7	News	1	1	0	0	1	1	1
F-7	Foro	1	1	0	0	1	1	1
F-7	Discusiones	1	1	0	0	1	1	1
G-7	Aprobación	1	1	1	0	1	1	1
C-7	Calendarios de grupo	0	1	1	0	1	1	1
C-7	Planificación compartida	0	1	1	0	1	1	1
A-7	Gestión de Workflow	0	0	1	0	1	1	1
C-7	Gestión de Eventos	0	1	1	0	1	1	1
C-7	Agenda	0	1	1	0	1	1	1
D-11	Co-Navegador	1	0	0	1	0	1	1
D-11	Pizarra compartida	1	0	0	1	0	1	1
B-7	S. de Notificación	0	1	0	0	1	1	1
F-13	S. de Presentación	1	1	0	1	1	0	1
G-11	GDSS	1	1	1	1	0	1	1
G-15	BSCW	1	1	1	1	1	1	1
F-7	Sharepoint Portal Server	1	1	0	0	1	1	1
F-10	Meeting Room	1	1	0	1	0	1	0
F-9	Videoconferencia	1	1	0	1	0	0	1

Figura 10. Ejemplos de funciones, herramientas y sistemas clasificados.

Además, según la apreciación que se quiera hacer, la tabla de clasificación no excluyente puede dar unos resultados u otros.

Por ejemplo, si alguien necesita clasificar herramientas según la funcionalidad original y básica para la que han sido implementados puede obtener un resultado como el que se muestra en la Figura 11, sin embargo, si lo que desea es obtener una clasificación de aquellas funciones para las que podría emplearse adicionalmente, obtendría una clasificación como la que se muestra en la Figura 12.

En el ejemplo se puede apreciar la distinta clasificación de una herramienta sencilla como el e-mail. Si la intención es determinar las características básicas y originales de esta clase de herramientas como resultado se obtiene el tipo *B-5*, lo que quiere decir que se trata de herramientas de comunicación asíncrona para personas situadas físicamente distantes.

Si lo que se quiere mostrar es las posibilidades del e-mail, el resultado será el tipo G-7. Una herramienta que puede fomentar la colaboración entre usuarios y mediante la que se puede coordinar a diferentes personas.

No son programas síncronos por naturaleza puesto que el mensaje enviado puede llegar pasado un tiempo y a pesar de que están orientados a que se transmitan mensajes desde espacios diferentes, se pueden emplear dentro de una misma oficina como recordatorios, para informar, para coordinar, etc.

B-5	E-mail	0	1	0	0	1	0	1
-----	--------	---	---	---	---	---	---	---

Figura 11. Clasificación según funcionalidad básica y original

G-7	E-mail	1	1	1	0	1	1	1
-----	--------	---	---	---	---	---	---	---

Figura 12. Clasificación según funcionalidad adicional posible

2.4.5 Análisis de la Clasificación

De esta clasificación se pueden sacar algunas conclusiones interesantes acerca de la configuración típica de las herramientas CSCW.

Las aplicaciones menos frecuentes son las que encajan en un patrón par, es decir, las herramientas tipo -6, -10, -14 no son muy frecuentes porque no es habitual que un sistema se desarrolle para trabajar únicamente en el mismo lugar, y no para poder hacerlo en la distancia, como sería el caso de una *meeting room*. Las aplicaciones CSCW, generalmente, encajan con códigos espacio-temporales impares (patrones -5, -7, -9, -11, -13 y -15), puesto que están ideadas para poder ser utilizadas desde espacios diferentes, aunque muchas de ellas se puedan utilizar en el mismo espacio y siga teniendo sentido su uso (de hecho en la mayoría de los casos esto es así). Esta conclusión es lógica, puesto que el medio de comunicación para estos sistemas es Internet (u otras redes de comunicaciones).

Los patrones espacio-temporales en los que encajan la mayoría de las herramientas son -7, -9 y -11; herramientas síncronas o asíncronas independientemente del lugar desde el que se empleen, o herramientas síncronas orientadas a un uso desde espacios diferentes.

Según las características propias del CSCW, las aplicaciones suelen incluir, al menos, comunicación como una de sus funciones, ya sea entre miembros de la organización directamente, o comunicación de información como noticias, eventos, etc. hacia los miembros de la organización. Como puede apreciarse en la Figura 10, como en esta pequeña clasificación, más del 80% de las herramientas encajan en patrones -B, -C, -F o -G que son los propios de la comunicación.

La taxonomía es *no excluyente* en el sentido de que una función se podría llegar a clasificar con diferentes atributos a la vez. Es decir, una herramienta se podría clasificar de manera que se mostrara su capacidad de ser usada tanto en sitios diferentes como en el mismo espacio.

La matriz de clasificación no excluyente es *sólida* puesto que las herramientas sólo pueden clasificarse de una única manera si se basa en la definición del autor acerca de su origen y objetivos. Las herramientas son creadas para realizar una serie de funciones originalmente y éstas son las empleadas para realizar la clasificación.

Otra cosa es que una herramienta implementada para realizar una función se pueda emplear para realizar otras. Un correo electrónico, en su origen, se emplea para la comunicación entre espacios diferentes, pero ya se ha visto en el apartado anterior

que se podría utilizar para una comunicación en el mismo espacio. En este sentido, la matriz de clasificación no excluyente es *flexible* de acuerdo con las necesidades que surjan en cada momento, de manera que se puede adaptar a otras posibles necesidades de clasificación manteniendo la misma estructura. Por lo tanto, se puede decir de forma sólida que una herramienta de correo electrónico es de tipo *B-5*, pero en cualquier caso se podría dar un enfoque flexible si lo que se quiere mostrar son las posibilidades adicionales de esta herramienta, en cuyo caso se podría corresponder con una herramienta de tipo *G-7*.

2.5 Ventajas e Inconvenientes de estos Sistemas

El despliegue de un sistema CSCW, como todas las tecnologías y todas las filosofías, implica una serie de ventajas y de inconvenientes [Coleman, 1997; Andriessen, 2003; Khoshafian, 1995; Penichet, 2003; Grudin, 1988].

Algunas de las ventajas que supondría la implantación de un sistema groupware son las siguientes:

- Facilita la comunicación. La hace más clara, rápida y persuasiva.
- Posibilita una comunicación que de otro modo no sería posible.
- Posibilita la comunicación a distancia, sin importar el lugar físico donde se encuentren los distintos miembros del grupo, reduciendo el número de reuniones.
- Integración total de grupos separados físicamente.
- Hay un mejor control de costes.
- Reducen gastos por desplazamientos, viajes, dietas, alojamiento, etc.
- Mantiene informados y formados a todos los miembros del grupo sin necesidad de tener un gran número de personas dedicadas a ello para hacerlo de forma personal. Fácil acceso a información nueva y/o de última hora por parte de todos los miembros.
- Mejora notablemente la coordinación de los grupos.
- Se aumenta la productividad.
- Automatización de procesos.
- Facilita la resolución de problemas en el grupo.
- Posibilita nuevos modos de comunicación como la aprobación de documentos e incluye y potencia otras ya existentes como los chat, los foros, las discusiones o los grupos de noticias.
- Mejora el servicio al cliente.
- Da soporte para TQM (Total Quality Management).

Y algunos de los inconvenientes recogidos en la literatura y experimentados con el tiempo son los que se muestran a continuación:

- El bajo nivel de formación en groupware.
- Desconfianza en el nuevo sistema.
- Se trata de grupos de usuarios, siempre difíciles de coordinar.
- La reacción que un grupo vaya a tener con un sistema es difícil de analizar.

- El propio dinamismo de los grupos puede resultar problemático. Los grupos no están siempre formados por las mismas personas, las personas no siempre tienen los mismos roles, etc.
- Rechazo de las organizaciones por el esfuerzo económico y de tiempo que hay que realizar inicialmente sin garantías tangibles de que vaya a funcionar.
- Sensación de dependencia del sistema de Groupware y más aún de los fabricantes del mismo.
- Resistencia al cambio por parte de los usuarios y más aun de las organizaciones.
- Demasiada organización para algunos, o demasiado trabajo.
- Falta de estándares en el mercado del Groupware.
- Se ha demostrado que cuando los sistemas proporcionan procesos muy sofisticados como, por ejemplo co-authoring, los usuarios finales del sistema no los utilizan con frecuencia a causa de la complejidad, las interfaces poco amigables, falta de entrenamiento, etc.
- Como consecuencia de la implantación de estos sistemas, algunas tareas se pueden automatizar, el modo de trabajo en general cambia, y esto puede llevar a que haya un reparto desigual de carga de trabajo.
- A pesar de que las aplicaciones CSCW facilitan la comunicación entre personas, también pueden perjudicar las relaciones personales y cercanas en una organización, entre los miembros que trabajan juntos.
- Diferenciar entre quién trabaja en grupo y quién se beneficia del trabajo en grupo.

En cualquier caso, habrá que tener en cuenta el cambio que supone la nueva filosofía de trabajo en el grupo donde se quiere implantar el sistema y tomar medidas, planificar, para que estos cambios sean lo menos traumáticos posibles.

Los usuarios del futuro sistema no cambiarán su modo de trabajo ni su forma de pensar en sólo unos días, es algo que supondrá un esfuerzo y algo de tiempo, y más aún la organización en su conjunto.

Según la ley de Coleman, "la gente se resiste a los cambios y las organizaciones se resisten a los cambios en un grado exponencialmente mayor". Y un corolario a la misma dice que "cuanto mayor sea la organización, más grande sea el cambio o más complejo el proyecto, mayor será la resistencia al cambio".

De todos modos, la necesidad de los sistemas colaborativos es cada vez mayor y entra en la sociedad de un modo natural, por las propias necesidades e inquietudes de la gente.

La sociedad evoluciona y precisamente lo hace hacia la globalización. Los proyectos no los realiza una sola persona y la persona se entiende, no como un ser individual, sino como un miembro activo de un grupo, que se relaciona dentro de él, y que se relaciona con el resto de grupos.

La tecnología debe entender cómo son este tipo de relaciones, cómo funciona el ser humano en grupo, para adaptar el software a las necesidades reales de los trabajadores, de las personas en general, de manera que las herramientas sean realmente una ayuda y no un inconveniente. Y ese es, precisamente, uno de los objetivos principales del CSCW.

2.6 Investigación en CSCW. Aplicaciones y Herramientas

Las bases del CSCW se han mantenido a lo largo de la breve historia de esta materia. Sólo han pasado veinte años desde su origen, pero ya se han construido multitud de herramientas y se han ideado multitud de soluciones.

El estudio del estado actual de cualquier materia se puede alcanzar desde el estudio de los congresos actuales en la misma, así como el seguimiento del trabajo realizado por grupos y expertos dedicados a ella. En el caso de CSCW el congreso por excelencia es el CSCW para los años pares y ECSCW para los impares. Se trata del mismo congreso, solo que los años pares se realiza en América y en los años impares en Europa.

Debido al carácter interdisciplinario del CSCW, el número de campos sobre el que se puede aplicar es enorme. Tan sólo observando el número de campos diferentes del último congreso CSCW mantenido en noviembre de 2004 en Chicago, puede verse esta magnitud. Había sesiones dedicadas a arquitecturas, grandes displays, compartición del conocimiento, evaluación, aplicaciones médicas, sistemas, conciencia y disponibilidad social, comunidades, interacción con dispositivos compartidos, diseño de sobremesa, asuntos organizacionales, componentes, juegos, equipos distribuidos, gesticulación, movimiento y habla, interrupciones y colaboración síncrona.

En este apartado se muestran algunas investigaciones interesantes, muchas de las cuales facilitan nuevos modos de comunicación. Se describen asuntos relativos a coordinación, otros a colaboración, concretamente a gestión documental. Se enumeran algunas herramientas con objeto de proporcionar una idea global de las posibles aplicaciones del CSCW.

2.6.1 Investigaciones

En relación con las arquitecturas, las dinámicas son apropiadas para estos entornos. Éstas determinan la naturaleza de los componentes del sistema y en qué máquinas se han de colocar esos componentes.

En general, las máquinas de los usuarios que participan en la sesión colaborativa ejecutan un componente de interfaz de usuario que se comunica con otro componente local o remoto dependiendo precisamente de la arquitectura del sistema, es decir, de si es *replicada* o *centralizada* (Figura 13) [Chung, 2004].

En una *arquitectura centralizada* (a) todos los usuarios interaccionan con un único programa que reside en la máquina de alguno de ellos.

En una *arquitectura replicada* (b) el programa reside en la máquina de cada usuario y se sincronizan todos ellos por medio de *broadcasting*.

Todas las arquitecturas tienen pros y contras. Por ejemplo, una arquitectura centralizada ofrecería un rendimiento mejor cuando (a) la máquina donde se ejecuta el programa en cuestión sea más potente que el resto de las máquinas que participan del sistema colaborativo, (b) el sitio central está conectado al resto por una conexión de red rápida, (c) el usuario del sitio central genera muchas más entradas al sistema que el resto y (d) el número de usuarios es muy grande.

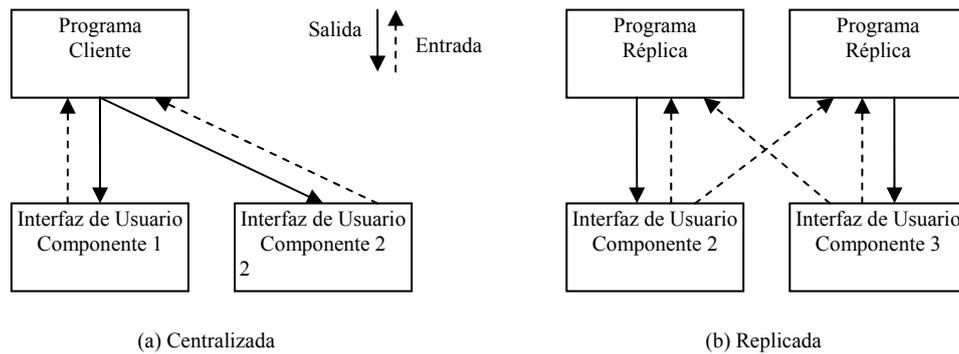


Figura 13. Arquitecturas centralizada (a) y replicada (b)

Los diseñadores de sistemas colaborativos comerciales emplean arquitecturas según sus necesidades.

También es posible encontrar *arquitecturas híbridas*, formadas por subarquitecturas centralizadas y subarquitecturas distribuidas. Así, al iniciar la sesión colaborativa, se tiene la flexibilidad de poder elegir entre una u otra.

Pero más flexible todavía es que el tipo de arquitectura pueda cambiar en tiempo de ejecución, dinámicamente, en respuesta a cambios en la configuración de los usuarios que interactúan.

Como acabamos de ver, la arquitectura depende de las conexiones de red, de las computadoras, de las entradas que genera el usuario y del número de usuarios. Y todo esto puede cambiar con la entrada o salida de un usuario del sistema colaborativo. Por ejemplo, si se trata de una arquitectura centralizada y el usuario que tiene la máquina donde se ejecuta el sistema debe cerrar su sesión, no podría si quiere que el sistema, y los demás usuarios por consiguiente, siga en marcha. Las *transiciones dinámicas entre arquitecturas* centralizadas, replicadas e híbridas podrían resolver este problema (Figura 14).

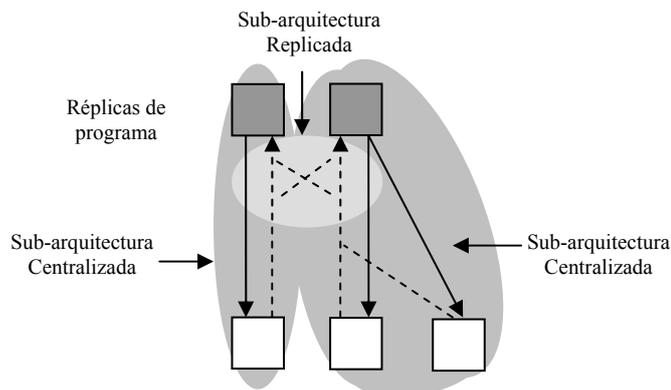


Figura 14. Arquitectura híbrida

También es muy actual encontrar esfuerzos dedicados a emplear grandes dispositivos como pantallas planas para conseguir aplicaciones diferentes en grandes salones, en grandes espacios, de manera que un gran número de personas puedan beneficiarse o interactuar con ella.

Desde el NASA Ames Research Center, se ha elaborado un sistema colaborativo para la misión de la NASA Mars Exploration Rovers (MER) 2003 [Tollinger, 2004].

La herramienta se conoce como MERBoard y proporciona un espacio de trabajo colaborativo para los científicos de la misión MER 2003. En principio fue diseñada como una pizarra compartida (digital whiteboard), pero con la ventaja y la diferencia de que los datos se almacenan de forma persistente, recuperable y distribuida.

En junio de 2003 se lanzaron al espacio dos robots exploradores (rovers) con destino a Marte. Llegaron con éxito a su destino en enero de 2004 con el objetivo de realizar diversos experimentos sobre la superficie del planeta rojo.

Cada uno de los rovers pueden tener una actividad de unas horas por cada sol (día marciano) durante esas horas se van recogiendo datos que se envían posteriormente a los más de 120 científicos, que disponen de 18 horas para procesarlos, analizarlos y usarlos para decidir qué actividades científicas se pedirán para el sol siguiente.

Los científicos se dividen en cinco grupos temáticos. Entre todos los grupos deben consensuar un conjunto de actividades prioritarias que se entregarán al equipo de ingenieros. A este conjunto se le conoce como Science Operations Working Group (SOWG). Tienen sólo unas horas para procesar toda la información recibida y saber lo que hacer posteriormente, antes de que se empiecen a recibir datos de nuevo.

La aplicación MERBoard permite compartir en tiempo real diagramas inteligentes que emplean estos científicos, pero además pueden compartir, ver y discutir otro tipo de información. Básicamente es un área de trabajo representada en una pantalla táctil de plasma de grandes dimensiones que proporciona una digital whiteboard para facilitar la colaboración informal de diversos grupos de científicos.

Otro ejemplo de herramienta que hace uso de pantallas grandes se ha desarrollado entre el Intel Research y la Universidad de Washington. Elaboraron en 2003 unas herramientas que facilitaban la colaboración y la comunicación en espacios sociales como conferencias académicas [McCarthy, 2004]: *AutoSpeakerID (ASID)* y *Ticket2Talk (T2T)*.

Las conferencias académicas proporcionan un tiempo y un espacio ideales para personas con intereses comunes, para que puedan intercambiar opiniones e información mutuamente, para aprender del trabajo de cada uno. Incluso se dan intercambios informales de información en momentos como la recepción o los breaks.



Figura 15. AutoSpeakerID

Una de las formas posibles consiste en un *proactive display*: un gran dispositivo de computación que está equipado con sensores que detectan personas cercanas y les responde en el contexto adecuado.

AutoSpeakerID (ASID) muestra la foto, el nombre y la afiliación de la persona que está realizando la pregunta (o respuesta) durante el turno de preguntas y

respuestas que sigue a la presentación de un artículo o un panel. El micrófono está "aumentado" (augmented) con una antena RFID, de manera que cuando la persona que va a realizar la pregunta se acerca el micrófono, si ésta lleva un RFID tag en su placa distintiva, se reconocerá y mandará la señal para que sea esta la persona que salga por pantalla con los datos mencionados.



Figura 16. Ticket2Talk

En *Ticket2Talk* (*T2T*) el mecanismo es parecido al de la aplicación *ASID*, pero en este caso se trata de una herramienta para un entorno más informal, como podría ser un descanso en la conferencia. Habría antenas distribuidas por la zona que identificarían las etiquetas RFID de las placas distintivas de las personas y mostrarían información de la persona (como por ejemplo sus intereses) en una pantalla activa cercana.

El grupo CHORD de la Universidad Federal de Río de Janeiro ha desarrollado un editor cooperativo que permite trabajar de un modo interactivo y en tiempo real a varias personas. De este modo, con CO2DE [Meire, 2003], que así se llama la herramienta, se pueden crear diagramas entre varios miembros de un equipo a pesar de que estén separados en la distancia. La herramienta, además, posee un sistema de versionado que permite ver los avances entre las diferentes versiones. En la Figura 17 se pueden apreciar dos punteros correspondientes a dos personas que están trabajando a la vez en la creación de un diagrama.

Otros grupos de investigación centran sus estudios en el *awareness*, conciencia o percepción. En la literatura de CSCW se ha insistido en que la eficiencia y la efectividad dependen, en gran medida, de lo bien informados que estén los individuos de una organización de las actividades que realizan sus compañeros [Gross, 2004]. Esto es lo que se conoce como *awareness*. También se puede ver como *group awareness* (percepción de grupo) o *workspace awareness* (percepción de espacio de trabajo).

Por ejemplo, en la Universidad de Saskatchewan (Saskatoon, Canadá) realizan estudios sobre la conciencia de grupo en el desarrollo de software distribuido [Gutwin, 2004b]. Los proyectos de desarrollo de software abierto son casi siempre colaborativos y distribuidos. A pesar de las dificultades impuestas por la distancia, se crean proyectos de gran envergadura con éxito (el sistema operativo Linux, el servidor Web Apache, las OpenOffice).

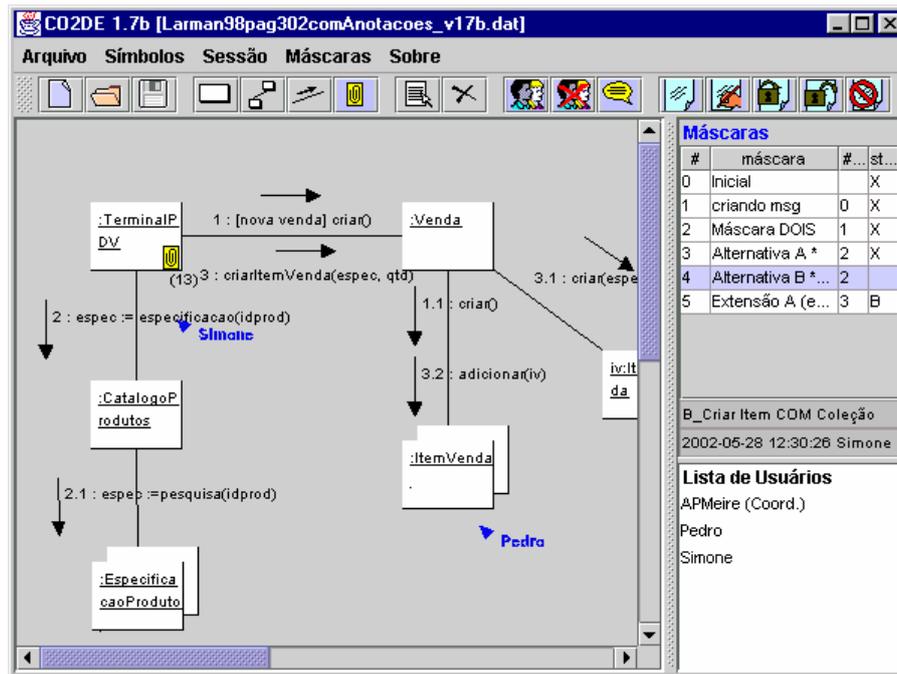


Figura 17. Aplicación CO2DE en la que están trabajando dos autores simultáneamente

Esta conciencia de grupo incluye conocimiento sobre quién está en el proyecto, en qué parte del código trabaja, qué hace, cuáles son sus planes, etc.

Se mantienen informados y saben coordinar sus esfuerzos, añadir código sin problemas, hacer cambios que afectan a otros módulos o evitar realizar un mismo trabajo varias veces.

Las herramientas colaborativas también pueden ayudar directamente en el desarrollo software. En la Universidad British Columbia (Canadá) se está elaborando un prototipo llamado Hipikat [Čubranić, 2004] que facilita aprender de los proyectos software de programación que están en marcha, de manera que una persona que entra nueva en el proyecto puede utilizarlo para aprender de él en lugar de tener que recurrir sólo a los compañeros. Se trata de conseguir recursos humanos productivos de forma inminente al entrar en proyectos que ya estaban en marcha.

Por un lado se crea la *memoria de proyecto* a partir de los diferentes artefactos y comunicaciones creados durante la historia del desarrollo del proyecto y luego se crean unas *recomendaciones* para el desarrollador a partir de esta memoria y relacionadas con la tarea que está o va a llevar a cabo.

Cuando el desarrollador hace alguna consulta relacionada con algún artefacto, inmediatamente el sistema sigue los enlaces establecidos y le sugiere diferentes posibilidades. Por ejemplo, si el desarrollador quiere realizar una tarea de cambio de una característica, el sistema sugerirá que mire otros cambios que se realizaron con anterioridad y que están relacionados con éste. Así probablemente pueda ver trozos de código que ya se generaron y que le puedan servir ahora, o ver mensajes asociados, etc.

Otro tema interesante es la evaluación de todo este tipo de sistemas colaborativos. En la Universidad de Penn State (Pensilvania) se centran en la evaluación por escenarios de uso [Haynes, 2004], lo que les permite focalizar los esfuerzos de evaluación e identificar los factores técnicos, humanos, organizacionales, etc. que hacen que un sistema tenga éxito.

Estos escenarios representan instancias concretas del sistema completo abarcando espacio, tiempo, personas, características del sistema, etc. proporcionando a diseñadores, desarrolladores y a otros actores partes concretas del sistema global donde centrar sus análisis, lo que desemboca en unos resultados más óptimos.

2.6.2 Coordinación

También la coordinación, de grupos humanos que trabajan con un fin común o de procesos, juega un papel fundamental. De hecho, la coordinación es uno de los pilares de un sistema CSCW. Los sistemas workflow se han convertido en una herramienta fundamental desde incluso antes de que se conociera el CSCW como tal.

La WfMC (*Workflow Management Coalition* [WfMC, 1999]) ha publicado un modelo de referencia arquitectónico (*Architectural Reference Model*) que describe la estructura y los interfaces de un *Workflow Management System*. Para avanzar en el desarrollo de unos estándares y una terminología comunes ha descrito unos conceptos básicos, la estructura y conceptos de proceso y una terminología y conceptos de workflow algo más avanzados (ver Figura 18).

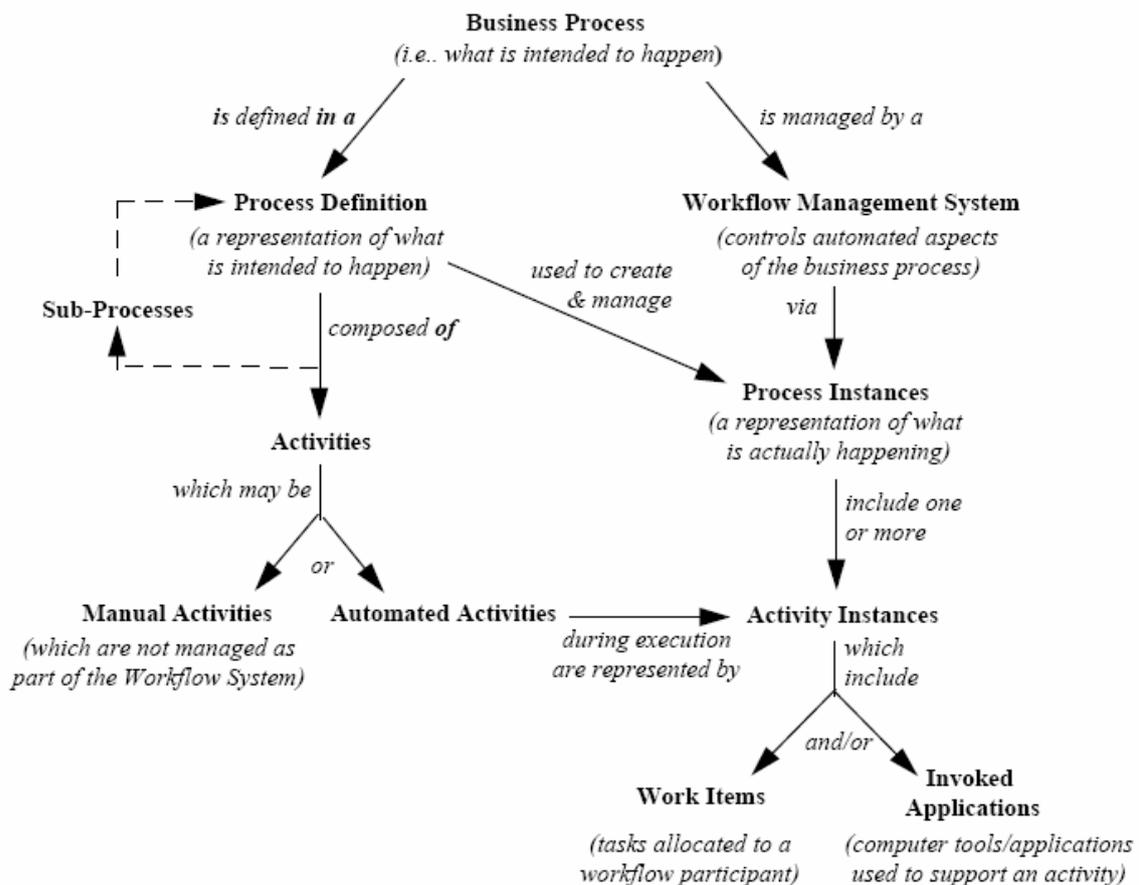


Figura 18. Relaciones entre la terminología básica de un Workflow

El profesor Wil van der Aalst de la Eindhoven University of Technology (EUT) es una referencia internacional en este campo. En concreto modela sistemas workflow como redes de petri de alto nivel (CPN, etc.) (se puede ver un ejemplo en la Figura 19).

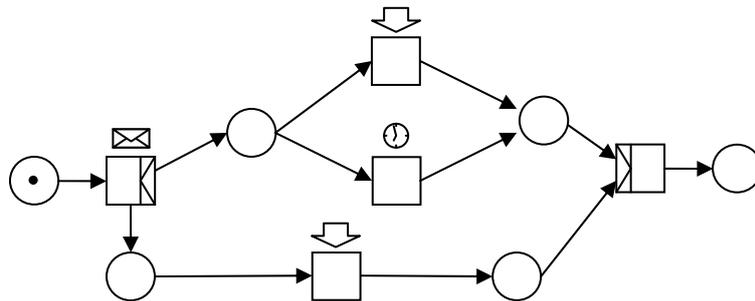


Figura 19. Ejemplo de workflow como Red de Petri de alto nivel

Según el Dr. van der Aalst, las personas trabajan para vivir, o al menos eso es lo que debería ser. De hecho, trabajamos para poder producir todo lo que necesitamos, desde las necesidades más básicas, como la comida, la casa, etc. hasta las necesidades de puro entretenimiento. Pero lo que también es cierto, es que una sola persona no puede realizar todo por ella misma porque además no sería eficiente. Por eso, las personas se dividen en "unidades de negocio" para poder realizar trabajos particulares y conseguir productos particulares. Se consiguen de esta manera productos de un modo más eficiente. Los trabajos están relacionados. Un producto puede no tener sentido sin otros, otros trabajos se crean para poder producir productos que ayuden en otros trabajos, etc.

En este sentido somos tan "eficientes" que nos podemos permitir el lujo de tener tiempo libre y "gastarlo" en algunas "unidades de negocio" creadas por otros trabajadores. Es decir, la organización ha posibilitado que se creen "unidades de negocio" que ni siquiera tienen un objetivo vital o sirven para complementar a otras unidades, sino que son meramente lucrativos.

Para hacer todo ese trabajo controlable y para animar a la comunicación entre trabajadores aparecen los sistemas de gestión de workflow *WMS*. Son una nueva clase de información que hacen posible, de un modo contundente, construir un puente entre el trabajo de las personas y las aplicaciones.

Los procesos se modelan y analizan mediante un mecanismo formal que, entre otras ventajas, proporciona una definición formal: las Redes de Petri (PN, ideadas por Carl Adam Petri en 1962 para modelar y analizar procesos). Además de proporcionar una definición formal evita ambigüedades, incertidumbres y contradicciones y se pueden mostrar los procesos gráficamente. Concretamente, como se comentó con anterioridad, se propone el uso de *Redes de Petri de Alto Nivel* (High-Level Petri Nets).

En la Figura 20 se puede ver un ejemplo de unas notaciones especiales para construcciones comunes, una transformación de los conceptos de workflow como AND-split, AND-join, OR-split, etc. hacia las redes de petri.

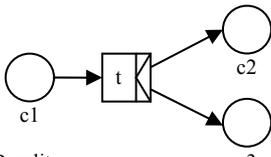
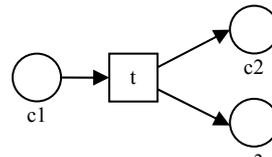
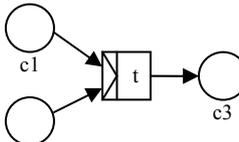
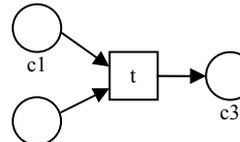
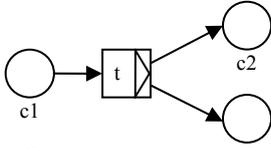
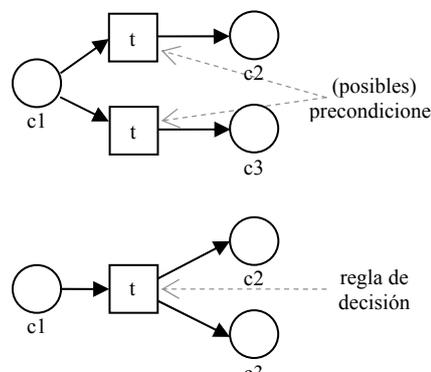
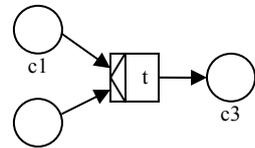
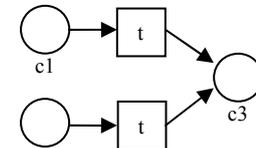
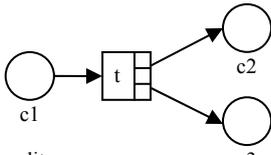
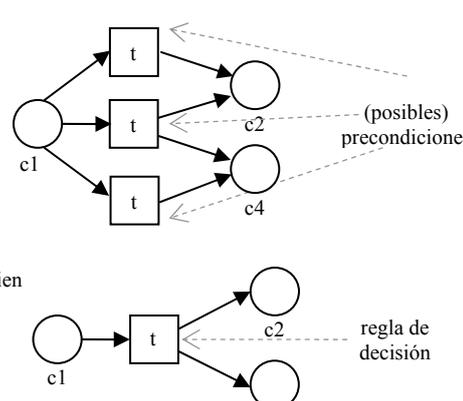
NOTACIÓN	SIGNIFICADO
 <p>AND-split</p>	
 <p>AND-join</p>	
 <p>OR-split</p>	<p>o bien</p> 
 <p>OR-join</p>	
 <p>AND/OR-split</p>	<p>o bien</p> 

Figura 20. Notaciones especiales para construcciones comunes

2.6.3 Colaboración

Un campo de aplicación de la colaboración es la gestión del conocimiento, donde un grupo de usuarios genera ese conocimiento y lo comparte. Los sistemas de *gestión del conocimiento* en general, y los sistemas de *gestión documental* en particular, juegan un papel importante en los entornos colaborativos. La gestión del conocimiento es "la habilidad de las personas para entender y manejar la información utilizando la tecnología y la compartición del conocimiento" [Dutta, 1997], es decir, la unión de personas e información por medio de la tecnología, potenciada por la capacidad de compartir el conocimiento. Se entiende por herramienta de *gestión documental* como toda aquella que está orientada al trabajo con documentos, centralizando y organizando la información y facilitando el acceso y el trabajo con la misma a los miembros de una organización.

En la Figura 21 se puede ver una clasificación de diferentes herramientas de gestión del conocimiento.

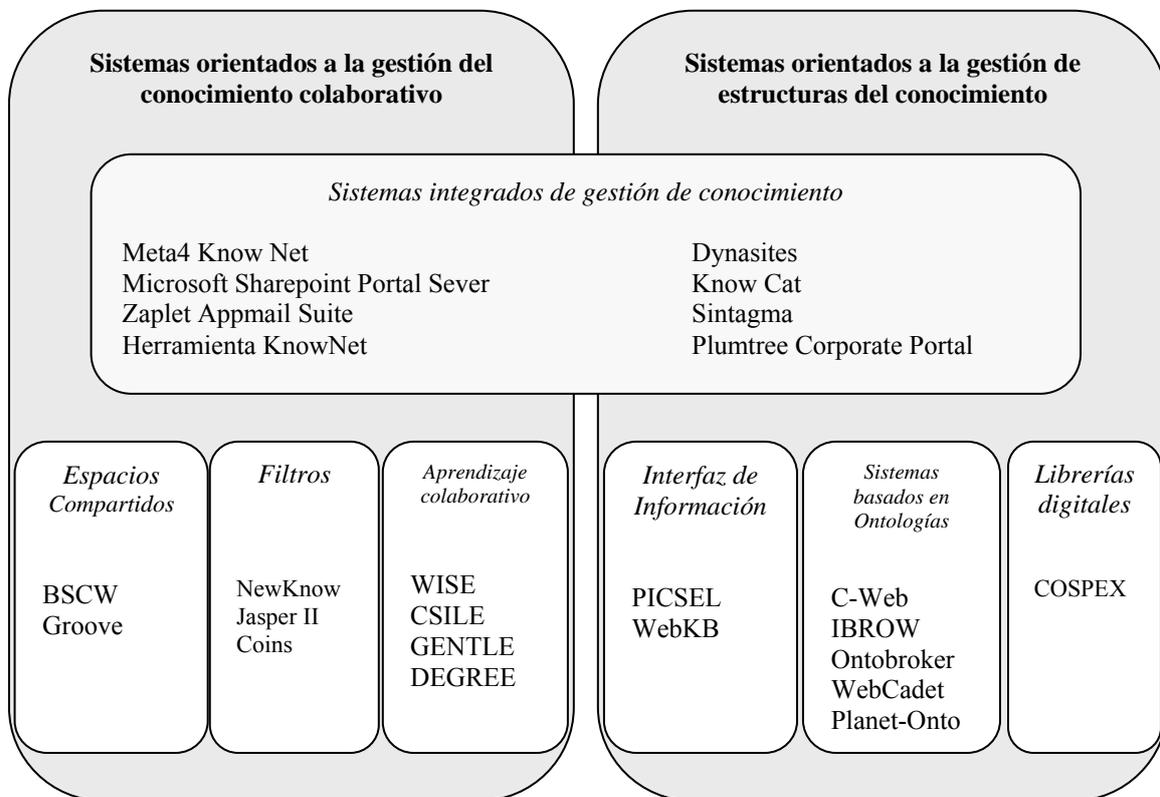


Figura 21. Clasificación de los Sistemas de Gestión del Conocimiento [Cobos, 2002]

En la Universidad Autónoma de Madrid se elaboró un sistema de gestión del conocimiento conocido como KnowCat [Alamán, 1999].

KnowCat es un sistema distribuido que, sin necesidad de supervisión, permite la creación incremental de conocimiento estructurado. La motivación subyacente es permitir la generación de conocimiento colectivo de alta calidad como resultado de la interacción de los usuarios con dicho conocimiento. El nombre, KnowCat, es el acrónimo de "Knowledge Catalyser" o "catalizador de conocimiento", haciendo referencia a la propiedad principal que exhibe el sistema: la catalización del proceso de cristalización del conocimiento.

En KnowCat hay unos nodos de conocimiento que se va alimentando de documentos, de información. Según las opiniones de los usuarios y el grado de utilización de esos documentos, se puede producir el proceso de cristalización del conocimiento cuando se consolida algún documento. La información es importante si la opinión sobre ella es importante y se usa con frecuencia. Un documento sin opiniones positivas o sin ser usado estará en espera de su posible cristalización o incluso a la espera de ser eliminado.

Este proceso de cristalización se basa en el trabajo colaborativo de comunidades virtuales de expertos, que son grupos de usuarios considerados expertos en uno o más temas. Estos expertos son los que pueden opinar con sus votaciones sobre los documentos existentes en lo que se conoce como *fase activa*. Alguien ha llegado a ser experto en un tema cuando documentos suyos se han cristalizado.

Pero para empezar en un tema, no hay una comunidad virtual de expertos inicial que permita que el sistema funcione "solo". Por eso existe una fase inicial o *fase supervisada* en la que usuarios con rol de coordinadores toman la decisión de pasar a *fase activa* cuando lo crean oportuno y se encargan de actuar como expertos en todos los temas de forma temporal hasta que se pasa a *fase activa*.

Otros sistemas de gestión del conocimiento más conocidos son BSCW o Microsoft Sharepoint Portal Server.

BSCW [BSCW, 1995; Penichet, 2003] es una herramienta de trabajo colaborativo basado en Web desarrollada en Python, que permite a los diferentes miembros de un grupo cooperar de forma síncrona o asíncrona a través de Internet o desde una Intranet.

De forma asíncrona, ofrece un espacio de trabajo compartido, accesible a través del navegador de Internet que se utilice. Mediante esta herramienta se tendrá informado a cada miembro del grupo que acceda, sobre ciertos eventos. Además se pueden subir documentos a esta área de trabajo, de manera que el resto de usuarios puedan verlos o no según los derechos de que se provean.

De forma síncrona, se proporcionan una serie de herramientas para planificar y organizar reuniones virtuales, etc.

La Figura 22 muestra el aspecto general de la intranet de un usuario en un sistema BSCW.

Así mismo, Microsoft Sharepoint Portal Server SPS 2001 [Microsoft, 2003a; Microsoft, 2003b; Penichet, 2003] es un entorno colaborativo de Microsoft basado en Web para facilitar a los componentes de una organización la búsqueda, organización y administración de sus datos.

Integra funciones de administración de la información como protección, desprotección, perfiles y publicación de documentos, identificación de versiones, enrutamiento de aprobación, notificaciones, seguimiento del historial de los documentos y recuperación de versiones anteriores. Posibilita una búsqueda eficiente.

En la Figura 23 se muestra el portal Web de un sistema Sharepoint Portal Server (versión 2001). En concreto se trata de la página principal del administrador vista desde la Web.

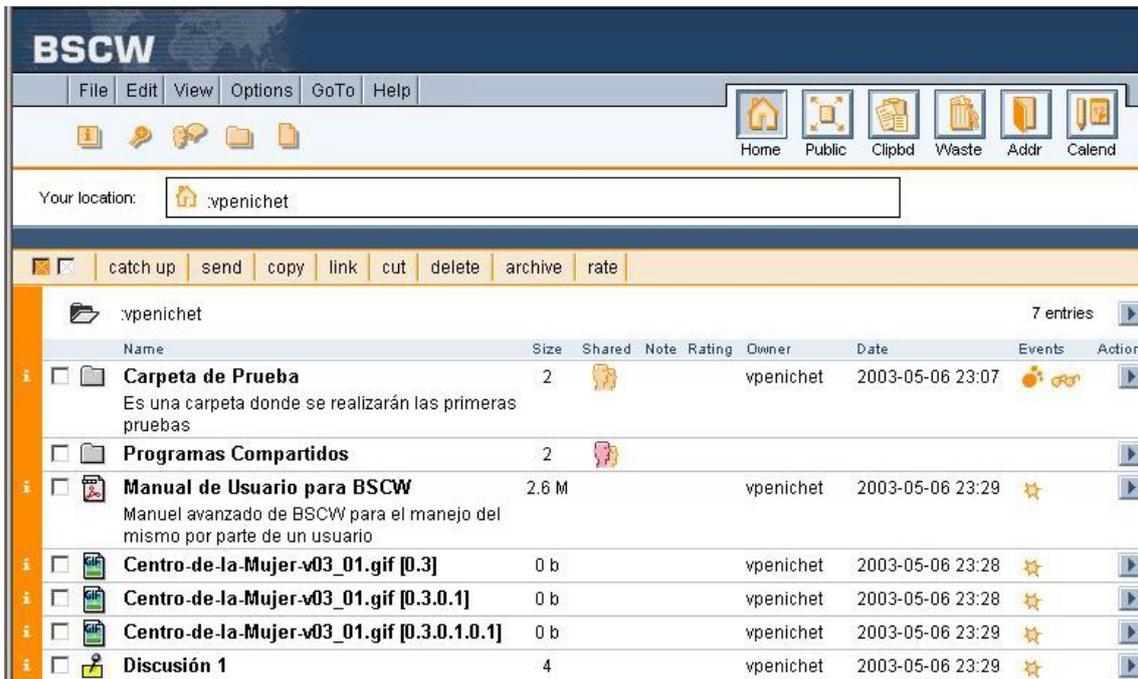


Figura 22. Aspecto general de la intranet en un BSCW

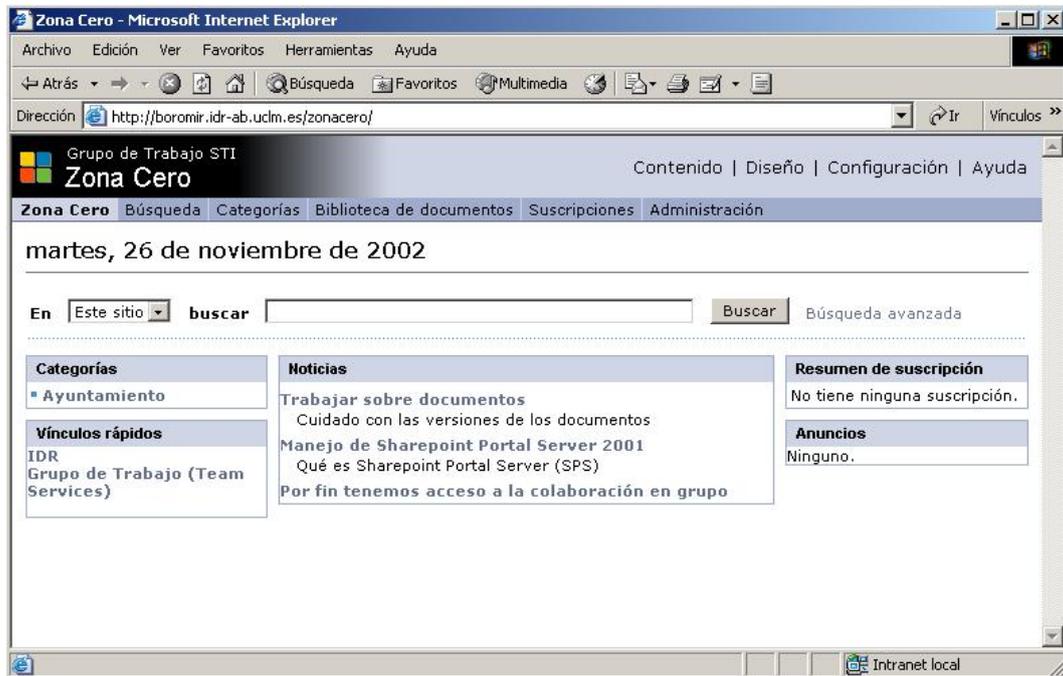


Figura 23. Vista de la página principal de un sistema Sharepoint Portal Server

2.6.4 Herramientas

Hoy día se pueden encontrar multitud de herramientas que dan solución a las necesidades groupware de una organización. Muchas de ellas resuelven varios problemas a la vez como por ejemplo CommBits (basado en phpGroupware), que contiene un calendario en grupo, libro de direcciones Web, Webmail, encuestas, foro de discusión, etc.

Coleman en [Coleman, 1997] lista un grupo de herramientas que actúan sobre diferentes plataformas, sistemas operativos, sistemas de correo electrónico y arquitecturas de red. Algunas de las que destaca son:

- GroupWise-Novell Lotus
- TeamOffice-ICL/Fujitsu
- GoldMetal Workgroup-Decathlon
- Notes-Lotus/IBM
- OpenDoc-Apple/IBM
- OpenMind-Attachmate
- BAI-5000 Distribution Management
- Patient Tracking System-Management
- CustomerFirst-Repository
- System-Business Automation HelpDesk-Trellis
- Directions ProTEAM-Scopus
- Technologies CenterPoint-Bank of Montreal
- InterNotes Publisher-IBM/Lotus
- RoundTable-The ForeFront
- SamePage-WebFlow
- PCS 50-PictureTel
- Group Metro-Action Technologies

A continuación se muestran una serie de herramientas que se pueden catalogar como herramientas groupware porque cubren una o varias de las características del mismo:

- Herramientas de Correo electrónico, agenda, calendario, etc.
 - Eudora es capaz de ordenar el correo entrante y saliente, soporta múltiples cuentas de usuario y lanza aplicaciones de Internet desde el propio mensaje.
 - CCMail: programa de Lotus Corp. Es un sistema de mensajería basado en Lan que incluye formularios electrónicos integrados, calendario, agenda y un buscador.
 - Groupwise es una herramienta de Novell que gestiona tareas, calendarios personales, agendas de grupo e e-mail.
 - BeyondMail: programa de Banyan Systems, Inc., que permite a grupos de personas responder a preguntas y automatiza la gestión del correo electrónico.

- Outlook: herramienta de Microsoft que simplifica la comunicación por e-mail, permite la planificación de grupos, agendas, libro de direcciones y contactos, etc.
- CommBits.com es un conjunto de herramientas basadas en PhpGroupware que ofrecen a los usuarios un entorno colaborativo con mail, calendario, libro de direcciones, notas, mensajes, encuestas, etc.
- Sitios de WebMail.
- Herramientas de Información Compartida / Productos para Intranet y gestión documental
 - Collabra Server, de Netscape Corporation, que facilita a los usuarios participar en encuentros virtuales, grupos de discusión, compartir información, herramientas de notificación, etc.
 - Microsoft Exchange Server proporciona calendarios de grupo y permite compartir varios calendarios personales, así como tener carpetas públicas a las que pueden acceder los miembros de la organización, agendas, grupos de discusión, mensajes instantáneos, datos en tiempo real y videoconferencia, etc.
 - FirstClass, de SoftArc Inc. es un sistema de comunicación multiplataforma que facilita correo electrónico, discusiones electrónicas, compartición de archivos, etc.
 - Lotus Notes y Domino: sistema de mensajería y colaboración, calendario, agenda, compartición de documentos, etc.
 - TeamWare Office proporciona mail, calendario, foro, discusiones, librería de documentos compartidos. Basado en servicios Web, puede incluso ser utilizado en PDA
 - Microsoft SharePoint Portal Server es un portal flexible que permite buscar, compartir, publicar, etc. información fácilmente. Realizar desprotecciones y protecciones, versionado y búsquedas de documentos. Se integra con otros productos de la misma casa.
 - BSCW es una herramienta para la gestión documental y el trabajo en grupo mediante un entorno basado en Web.
- Herramientas de Videoconferencia y pizarra compartida
 - Microsoft NetMeeting: videoconferencia, Chat, transferencia de archivos, teléfono Web y pizarra compartida.
 - ICQ Pro 2003: chat, voz, pizarra de dibujo, transferencia de ficheros.
 - ThruCam: videoconferencia, Chat, compartición de archivos y pizarra.
 - PRIVARIA Secure Networking: intercambio de archivos, pizarra y videoconferencia.
 - CU-SeeMe video y audio conferencia.
 - Horizonlive: comunicaciones interactivas.
 - WebEx permite realizar reuniones virtuales, presentaciones en tiempo real, compartir documentos o programas, etc.
- Herramientas de comunicación síncrona
 - MSN Messenger: mensajería instantánea, transferencia de ficheros, etc.
 - Yahoo! Messenger: mensajería instantánea, transferencia de ficheros, etc.

- Mirc: chat, transferencia de ficheros, etc.
- Connecta 2000: chat, videoconferencia, telefonía IP, compartición de archivos, mail, mensajes instantáneos, etc.
- ICQ GroupWare: mensajes instantáneos, chats, transferencia de archivos, etc.
- Skype es un servicio de telefonía IP que permite la comunicación en tiempo real por medio de la de voz entre diferentes dispositivos.
- PeopleCall es un servicio de telefonía IP que permite la comunicación en tiempo real por medio de la de voz entre diferentes dispositivos.

Esta es sólo una muestra de la inmensa cantidad de posibilidades del CSCW, y en su mayoría se trata de aplicaciones recientes. Debido a que los sistemas en general tienden a ser colaborativos, a facilitar la comunicación entre las personas y a ser una herramienta útil en la coordinación de procesos y de personas, es de esperar que el número de aplicaciones, ideas, formas, etc. crezca en los años venideros. De hecho, las aplicaciones más o menos sencillas en cuanto al objetivo para el que se crean, pero con una fuerte carga colaborativa, como Picasa, Blogger, Calendar, Skype, Del.icio.us, Flickr o YouTube, son las que realmente están teniendo éxito de masas.

2.6.5 Metodologías Colaborativas

A la hora de querer implantar un sistema de información de cierta envergadura se ha de seguir una metodología que permita realizar los pasos de su desarrollo con una cierta lógica. Existen multitud de metodologías para cubrir el ciclo de vida del software que son perfectamente válidas como por ejemplo Métrica v3 [MAP, 2000] o RUP [Jacobson, 2000].

Muy a groso modo, el groupware trata de ayudar a realizar el trabajo de un grupo de personas que se comunican, cooperan entre ellas y que necesitan coordinarse o coordinar tareas, etc. por medio de Tecnologías de la Información y de la Comunicación, a través de redes de ordenadores.

Un sistema colaborativo, a pesar de las peculiaridades que pueda tener, es un sistema que se podría obtener aplicando una metodología genérica, pero puesto que se trata de sistemas especializados, no sólo en resolver problemas particulares, sino en facilitar las bases del CSCW en el desarrollo de aplicaciones para usuarios en grupo, se plantea la necesidad de emplear una metodología especializada que tenga en cuenta estos detalles concretos, que resultan de especial importancia en estos sistemas.

Existen algunas aproximaciones metodológicas relacionadas con interacción y con trabajos cooperativos. Estas aproximaciones se detallan en el apartado 3.9 en el capítulo siguiente.

CAPÍTULO 3.

CONCEPTOS FUNDAMENTALES EN ENTORNOS CSCW

Los esfuerzos realizados desde diferentes áreas de investigación han quedado patentes en el capítulo anterior donde se ve el CSCW como un campo de investigación que surge de la necesidad de encontrar en la tecnología un nexo de unión de diferentes líneas como la antropología, la filosofía, la sociología, la etnografía o la psicología.

Las herramientas Groupware han de solucionar problemas que van más allá de la interacción de la persona con la computadora. Deben tener en cuenta las necesidades de los usuarios como grupo que interacciona entre sí por medio de las máquinas y a través de la infraestructura de red.

En este capítulo se describirán algunos conceptos de los trabajos más incipientes relacionados con la terminología que se suele emplear en entornos colaborativos. Presentar una relación de todos los términos y todas las variantes en sus definiciones según los distintos autores sería una tarea muy compleja que haría perder el sentido de este capítulo, que pretende dar una visión amplia pero general del uso tradicional de estos conceptos.

Así mismo, resulta imposible separar completamente los términos en diferentes grupos, porque su grado de interrelación es muy elevado. Es decir, no sería posible hablar de las *tareas* sin comentar algo de *coordinación*, o describir los *roles* que desempeñan los usuarios del sistema sin mentar las *tareas* asociadas a los mismos.

Este capítulo concretiza el estado del arte genérico estudiado en el anterior. En él se muestran los diferentes esfuerzos realizados en campos de estudio que han logrado definir qué son las tareas cooperativas, cómo se relacionan los grupos o incluso qué significa el concepto de agrupación.

3.1 Definiciones Terminológicas: la Necesidad de un Lenguaje Común

Tras la revisión del estado del arte en torno a los sistemas colaborativos del capítulo anterior y tras la revisión de los campos de investigación relacionados que se aluden en este capítulo, es posible tener una idea clara de la gran cantidad de conceptos que surgen y que hay que manejar con soltura para poder hablar con propiedad dentro de este, y cualquier, ámbito.

Por este motivo, se considera de vital importancia mencionar las técnicas que se emplean para hacer un buen uso del vocabulario. Técnicas hay muchas, pero en este apartado se muestran aquéllas que, por su propia definición, están más cerca de los entornos colaborativos o que se acomodan más a aquello que se necesita para resolver el problema planteado en esta tesis.

3.1.1 Taxonomías

El origen del término taxonomía proviene del griego: *taxis* (ordenación) y *nomos* (norma). Por lo tanto, una primera definición del concepto taxonomía a partir de su origen podría ser *ciencia de la clasificación*. De hecho, la Real Academia Española [RAE, 2005] la define como *ciencia que trata de los principios, métodos y fines de la clasificación. Se aplica en particular, dentro de la biología, para la ordenación jerarquizada y sistemática, con sus nombres, de los grupos de animales y de vegetales*.

Efectivamente, el concepto de taxonomía fue empleado por primera vez en el entorno de la Biología para realizar la clasificación (que curiosamente es la segunda acepción de la RAE: clasificación) de las especies de un modo jerárquico agrupados por géneros, familias, órdenes, clases, filos y reinos.

En informática, también conocidas como *directorios*, es un concepto asociado a Web 1.0.

Bajo este sencillo paradigma, los elementos, unidades taxonómicas o taxones, se clasifican en una estructura jerárquica tipo padres-hijos.

Se podría considerar que una taxonomía es una versión muy reducida y tosca de una ontología.

El problema de las taxonomías, además de lo excesivamente sencillas que pueden llegar a ser, radica en la rigidez de la clasificación, así como en lo difícil que puede llegar a ser identificar el nodo más adecuado donde clasificar los elementos [Gruber, 2005]. En este sentido, otras soluciones como las folcsonomías resultan más cómodas y flexibles. Al contrario que en las taxonomías y esquemas de clasificación, en las folcsonomías no hay relaciones explícitas entre los términos.

En la actualidad se tiende al abandono de estas taxonomías a favor de las folcsonomías. El uso de las folcsonomías baja barreras a la cooperación [Mathes, 2004]: *“los grupos de usuarios no tienen que ponerse de acuerdo en la jerarquía de conceptos bajo la que debería estar clasificado un elemento en una taxonomía. Sólo necesitan ponerse de acuerdo en las etiquetas que utilizarán para caracterizarlos”*. En este sentido es más cómodo y flexible el uso de etiquetas para describir y/o caracterizar un elemento, en lugar de tener que “encajarlo” en una clasificación estática, en la que, a menudo, podría haber diferentes posibilidades. Estas diferentes posibilidades crean ambigüedad.

3.1.2 Ontologías

El uso de las ontologías es una posibilidad para obtener un conjunto de términos definidos que conformen un vocabulario de uso común sobre un tema determinado.

El término ontología proviene de las palabras griegas *ontos* (ser) y *logos* (razón, palabra, conocimiento). Es una rama de la filosofía relacionada con el estudio del ser, la realidad y la sustancia en su forma fundamental. Sin embargo, este término se ha venido utilizando a lo largo de la historia en otras muchas áreas del conocimiento de muy diversas formas. En [Deborah, 2002] se presenta un estudio interesante acerca de los diferentes usos de este concepto. Probablemente, la definición de ontología más aceptada sea la de Gruber [Gruber, 1994]: *“una ontología es una especificación de una conceptualización”*. En el mismo trabajo se presenta una explicación mucho más extensa que, brevemente, expresa que para compartir y reutilizar el conocimiento entre sistemas es necesario definir un vocabulario común que lo represente y que una ontología es una especificación de un vocabulario figurativo para un dominio de discurso compartido.

En la tesis de [Fernández, 2003] titulada *Un Entorno de Integración de Ontologías para el Desarrollo de Sistemas de Gestión de Conocimiento*, hay un amplio trabajo realizado donde estudia y explica la evolución de las ontologías y sus clasificaciones. En este trabajo se muestra una redefinición de ontología hecha por [Borst, 1997] sobre la definición planteada previamente por [Gruber, 1994]: *“una ontología es una especificación formal de una conceptualización compartida”*. *Formal* se refiere a la necesidad de disponer de ontologías comprensibles por las máquinas. *Compartida* se refiere al tipo de conocimiento contenido en las ontologías, esto es, conocimiento consensuado y no privado.

Un tercer autor, [Studer, 1998], pretende explicar en su conjunto las definiciones dadas por Gruber y Borst: *“conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo a través de la identificación de los conceptos relevantes de dicho fenómeno. Explícita significa que el tipo de conceptos y restricciones usados se definen explícitamente. Formal representa el hecho de que la ontología debería ser entendible por las máquinas. Compartida refleja la noción de que una ontología captura conocimiento consensual, esto es, que no es de un individuo, sino que es aceptado por un grupo”*.

La comunidad de la Inteligencia Artificial plantea un reto sobre la posibilidad de acceder fácilmente al conocimiento compartido [Neches, 1991] en términos de comunicación inter-aplicaciones a través de la red.

Aparece entonces la idea de que estos “glosarios” deben ser comprendidos de alguna manera por las máquinas y de que este conocimiento no ha de ser privado, sino público y compartido, aceptado por un grupo.

La Web Semántica [Bernes-Lee, 2001] aparece como un marco común que permite la compartición y reutilización de la información [W3C, 2006]. La Web Semántica es una Web de información. Se basa en RDF (Resource Description Framework) para representar la información e intercambiar el conocimiento en la Web; y en OWL (Web Ontology Language) para publicar y compartir conjuntos de términos, es decir, ontologías, que dan soporte a búsquedas avanzadas en la Web, agentes software y gestión del conocimiento.

OWL [W3C-OWL, 2004] es un lenguaje para definir ontologías Web. Una ontología definida con OWL incluye descripciones de clases, propiedades y sus instancias. Se trata de una representación del conocimiento, no sólo un formato de mensaje como podría representarse por medio de un XML Schema.

3.1.3 Folcsonomías

Una folcsonomía es un novedoso tipo de clasificación de la información. Se trata de un modo informal de categorización social de la información.

Informal porque las categorías se establecen por opiniones de los usuarios y no suelen llevar un control de sinónimos en el uso de las etiquetas que se utilizan para clasificar la información. *Categorización* porque se emplean una serie de etiquetas que permiten clasificar la información, de manera que se puedan realizar búsquedas de acuerdo a los criterios establecidos. Se establece una especie de red de etiquetas que van enlazando los conceptos que forman parte de la folcsonomía. Y *social* porque son los propios usuarios de los sistemas los que se encargan de clasificar la información de acuerdo a sus propias opiniones, fomentando así la participación grupal y las redes de conocimiento sociales.

El término folcsonomía (folksonomy) se atribuye a Thomas Vander Wal [Vander Wal, 2004], si bien el concepto de etiquetas para clasificar la información existía con anterioridad. Es un proceso colaborativo en el que diferentes usuarios categorizan contenido como páginas Web, fotografías, enlaces, etc. por medio de etiquetas. De esta manera, se facilita la recuperación de información y las búsquedas.

Folcsonomía es un concepto asociado a Web 2.0 [O'Reilly, 2005] donde se tienen más en cuenta aspectos sociales y de contexto y "sustituiría" al concepto de directorios o taxonomías. *"El mercado con etiquetas permite la clase de asociaciones múltiples, y solapadas que el propio cerebro humano utiliza, en lugar de categorías rígidas. En el ejemplo canónico, una foto de Flickr de un cachorro puede ser marcada con la etiqueta tanto 'cachorro' como 'lindo' – permitiendo la recuperación siguiendo los mismos ejes naturales generados por la actividad del usuario"*.

La idea de abandonar las taxonomías a favor de listas de palabras clave no es nueva. La diferencia fundamental en este tipo de sistemas es el feedback que aportan los usuarios [Udell, 2004].

"Of course, that idea's been around for decades, so what's special about Flickr and del.icio.us? Sometimes a difference in degree becomes a difference in kind. The degree to which these systems bind the assignment of tags to their use –in a tight feedback loop– is that kind of difference"
[Udell, 2004].

Ejemplos del uso de esta técnica son Furl [LookSmat, 2003], Flickr [Yahoo-Flickr, 2005] y Del.icio.us [Yahoo-del, 2005]:

- Furl es un servicio que permite al usuario guardar elementos que encuentra en la Web para poder encontrarlos nuevamente de una forma más eficaz y veloz. Con la información que los usuarios van guardando, Furl recomienda páginas que podrían ser del interés del usuario. Comparte esos contenidos guardados y establece redes de enlaces por medio de newsletters o de la integración de estos enlaces en Webs existentes.
- Flickr es una herramienta que permite la compartición y la gestión de fotografías. Facilita esa compartición a usuarios que estén interesados en determinados tipos de fotografías y posibilita un nuevo método de organización colaborativo por medio de comentarios, etiquetas y notas.
- del.icio.us se define como un *"social bookmarking website"*. Un portal donde los usuarios pueden preservar para uso futuro y compartir los enlaces, artículos, blogs, etc. que consideren favoritos. Debido al aspecto social, permite descubrir elementos nuevos de interés. Lógicamente, también se basa en el uso de etiquetas.

El uso de los metadatos empleados en las folcsonomías podrían ser la clave para el desarrollo de la Web Semántica, puesto que podría mejorar la precisión de la recuperación de la información.

Gruber hace una reflexión sobre ontologías y folcsonomías [Gruber, 2005]. Ontología es la tecnología que permite compartir información en la Web, es decir, es la tecnología base para la Web Semántica que permite identificar una conceptualización común de la información. Folcsonomía es el etiquetado de elementos por parte de los usuarios en un contexto social, que no una taxonomía [Vander Wal, 2004]. Se trata de conceptos diferentes con ventajas e inconvenientes. Por ejemplo, el uso de ontologías para realizar búsquedas es demasiado rígido y menos eficaz que mediante el uso de las etiquetas de las folcsonomías, etiquetas que, por otro lado, introducen "inteligencia humana" al sistema. Sin embargo, la especificación de los conceptos es más amplia por medio de las ontologías.

3.1.4 Microformatos

Los microformatos [Tantek, 2005] son un nuevo método de clasificación y definición de la información de la Red, basado en el marcado de los elementos.

Se trata de un conjunto de datos abierto pensado para funcionar sobre estándares existentes. En su definición se dice que:

- Son una manera de pensar sobre los datos.
- Son principios de diseño para formatos.
- Están adaptados a patrones de uso y comportamientos actuales.
- Están altamente relacionados con XHTML y AKA.

Así mismo se dice que no son:

- Un nuevo lenguaje.
- Infinitamente extensibles y abiertos.
- Un intento por que todo el mundo cambie su comportamiento y reescriba sus herramientas.
- Una nueva aproximación que "eche por tierra" todo lo que existe y funciona en la actualidad.
- La panacea para las taxonomías, ontologías y otras abstracciones.
- Un intento por definir el mundo completo.

La definición oficial de microformato, elaborada por Dan Cederholm y Tantek Çelik en junio de 2005, es la siguiente:

"Diseñados en primer lugar para las personas y en segundo lugar para las máquinas, los microformatos son un conjunto de formatos de información abierto y sencillo construido sobre estándares existentes ampliamente aceptados".

Existe una lista de microformatos muy amplia, entre los que destacan:

- *hCard* para representar gente, compañías, organizaciones y lugares usando una representación 1:1 de las propiedades y valores del estándar vCard en XHTML semántico.

Por ejemplo, una vCard sería:

```
BEGIN:VCARD
VERSION:3.0
N:Çelik;Tantek
FN:Tantek Çelik
URL:http://tantek.com/
ORG:Technorati
END:VCARD
```

Y la correspondiente hCard utilizando algunos :

```
<div class="vcard">
  <a class="url fn" href="http://tantek.com/">Tantek Çelik</a>
  <div class="org">Technorati</div>
</div>
```

El fragmento de hCard se visualizaría así (siendo el primero de los elementos un enlace a la URL indicada):

Tantek Çelik
Technorati

- *hCalendar* es un formato de eventos y calendarios distribuidos abierto y sencillo, basado en el estándar iCalendar, apropiado para ser embebido en (X)HTML, Atom, RSS y XML.
- *VoteLinks* es un microformato que extiende la etiqueta <a> (hipervínculo) de HTML con el atributo *rev* que podría tomar uno de estos tres valores: "vote-for", "vote-abstain" y "vote-against". Son autoexcluyentes y representan acuerdo, abstención y desacuerdo respectivamente.
- *XFN* es un método sencillo de representar las relaciones humanas a través de los hipervínculos. Resulta especialmente interesante en blogs y paginas personales.

Es una extensión de la etiqueta <a> (hipervínculo) de HTML con el atributo *rel* que podría tomar numerosos valores.

```
<a href="http://jeff.example.org" rel="friend met">...
```

Algunos de esos valores representan amistad (*friend*, *acquaintance*, *contact*), situaciones físicas (*met*), o profesionales (*co-worker*, *colleague*), relaciones familiares (*child*, *parent*, *sibling*, *spouse*, *kin*) o románticas (*muse*, *crush*, *date*, *sweetheart*), etc.

- *Rel-license* es un formato sencillo y abierto para indicar el contenido de las licencias. Apropiado para ser embebido en (X)HTML, Atom, RSS y XML.
- *rel-tag* es un microformato que añade el atributo *rel="tag"* a la etiqueta <a> (hipervínculo) de HTML, indicando de esta manera que la página de destino está hecha por el autor, o es parte, de la página actual.

Popularmente se entiende que los microformatos son un conjunto de "convenciones simples para integrar semántica en HTML y permitir un desarrollo descentralizado":

"Los microformatos se caracterizan por:

- ser convenciones simples
- diseñados para integrar semántica en etiquetas para solucionar problemas específicos
- se pueden usar en formatos diseñados para humanos: (X)HTML/XML, Atom/RSS, feeds, etc.
 - 23.1 normalizando el contenido existente usando patrones
 - 23.2 usando nombres de clases cortos y descriptivos
 - 23.3 basándose en estándares que ya funcionan
- permiten un desarrollo descentralizado de recursos, herramientas y servicios".

3.1.5 Un Modelo Conceptual del Groupware en 1994

Ellis y Wainer [Ellis, 1994] propusieron un modelo conceptual del groupware a mediados de los noventa. Lo plantean como solución a dos preguntas: (a) cómo se pueden *comparar* dos sistemas groupware distintos y (b) cómo se puede *describir* un sistema particular.

Comentan que un sistema groupware se puede caracterizar desde tres aspectos o modelos (como ellos mismos indican):

- una descripción *de los objetos y las operaciones* sobre esos objetos que están disponibles para los usuarios,
- una descripción de los aspectos dinámicos del sistema, es decir del *control y el flujo de datos*;
- y una descripción de la *interfaz* entre el sistema y los usuarios y entre los propios usuarios.

A cada uno de esos modelos los llaman *modelo ontológico*, *modelo de coordinación* y *modelo de interfaz de usuario* respectivamente.

No capturan aspectos de implementación. Ese nivel de detalle lo dejan para los responsables de implementación.

Con esta aproximación de los tres modelos, consiguen reflejar la intuición que un diseñador tiene respecto a algunos aspectos importantes de un sistema. En este sentido, los modelos conceptuales propuestos tienen fuertes similitudes con la ingeniería del software: el modelo ontológico está relacionado con la especificación estática del sistema, por ejemplo, los diagramas entidad relación; el modelo de coordinación se correspondería con la especificación funcional, diagramas de flujo de datos, redes de Petri u otras formas de descripción dinámica de sistemas. Sin embargo, la principal diferencia con estos modelos y sus correspondencias en la ingeniería del software sigue siendo que no están pensados para representar detalles de implementación. Por ejemplo, el modelo de coordinación describe la organización de actividades que realizan los usuarios, pero no la organización de procesos que realizaría el sistema.

Modelo Ontológico

El *modelo ontológico* de un sistema groupware es la descripción de las clases de objetos y las operaciones que se proporcionan sobre ellos a los usuarios. Dos sistemas groupware cuyo objetivo sea el mismo tienen dos modelos ontológicos diferentes que describen cómo son las operaciones en cada uno. Por ejemplo, dos sistemas de revisión de documentos científicos tienen el mismo objetivo de conseguir comentarios por parte de terceros, sin embargo, la manera de introducir los comentarios es diferente en ambos. En el ejemplo que muestran, los dos sistemas tienen objetos similares, entre ellos el objeto "comentario". Pero las operaciones sobre el objeto (modificar, insertar, etc.) son semánticamente diferentes. En uno de ellos los comentarios se añaden por frases, mientras que en el otro se añaden por párrafos.

Los *objetos* son estructuras de datos sobre las que operan los usuarios. Los objetos se modelan haciendo uso de conceptos estándares de modelado como los *atributos* y los *valores*: Los objetos tienen *atributos* y los atributos pueden tomar uno o más *valores*, valores que podrían ser atómicos o bien otros objetos. No está absolutamente relacionado con conceptos de metodologías orientadas a objetos. Por ejemplo, no se usan los términos "método", "paso de mensajes" o "herencia". El término objeto se emplea como sinónimo de estructura de datos.

Las **operaciones** son transformaciones que actúan sobre los objetos. Clasifican las operaciones en:

- **operaciones de vista**, que permite a un usuario conocer la existencia de un objeto y el valor de sus atributos,
- **operaciones de creación**, para crear nuevas instancias de objetos,
- **operaciones de modificación**, para cambiar o añadir valores de algunos atributos de una instancia de objeto.
- y operaciones de destrucción.

Distinguen entre **clases de objeto**, que representan objetos genéricos, prototipo; e **instancias de objeto**, que son entidades sobre las que se pueden aplicar las operaciones.

La **semántica deseada** (intended semantic) es la descripción del uso de los objetos, mientras que la **semántica operacional** (operational semantic) hace referencia a las restricciones de la clase con otras clases y al conjunto de operaciones que se pueden aplicar a instancias de esta clase. La **semántica deseada** de las clases de objetos es, normalmente, lo más importante en los sistemas groupware.

Otros conceptos a tener en cuenta en el modelo ontológico de un sistema groupware son los siguientes:

- **Derechos de operación y derechos de acceso**. Es decir, derechos de los usuarios para poder ejecutar ciertas operaciones o acceder a ciertos objetos. Hay un gran número de derechos que se deben considerar, como por ejemplo privacidad, protección y control.
- **Valores generados automáticamente**. No todos los valores de los atributos de los objetos los establecen los usuarios. Algunos valores los establece el sistema de forma automática.

Concluyen la descripción del modelo ontológico apuntando que no se trata de un modelo propio de los sistemas groupware, sino de todos los sistemas. Sin embargo, algunos aspectos son más relevantes en los primeros, como por ejemplo los derechos de operación y acceso.

Modelo de Coordinación

El modelo de coordinación describe las actividades que cada participante debe llevar a cabo y cómo se coordinan para que el grupo pueda lograr realizar su trabajo.

En torno a este modelo giran una serie de conceptos importantes como son los de actividad, rol y actor, entre otros.

Una **actividad** es un conjunto potencial de operaciones que un actor que desempeña un **rol** puede ejecutar. Una actividad tiene asociada una **meta** (goal). En general, un **actor** podría ser un usuario, un sistema informático o un grupo. El actor que realiza la actividad se dice que es el **ejecutor** de la misma. Un conjunto de actividades ordenadas dan lugar a un **procedimiento** (procedure).

Varios clientes que realizan una compra de un artículo precisarían del procedimiento "procesar orden". A cada una de esas órdenes, a cada una de esas instancias del procedimiento se las conoce como **endeavor**. Un procedimiento está formado por actividades, mientras que un endeavor está formado por instancias de actividades. A esas instancias de actividad se las conoce como **tareas**. Si la tarea

no ha comenzado todavía o ya ha sido finalizada se dice que es una tarea **inactiva** (inactive). En otro caso sería una tarea **activa** (active).

Las tareas se inician por medio de acciones de iniciación o **s-actions** y finalizan por medio de acciones de terminación o **t-actions**, ejecutadas por los actores del sistema.

Un aspecto relevante del modelo de coordinación es la precedencia temporal de las actividades. Algunas no pueden comenzar hasta que hayan finalizado otras, otras podrían ejecutarse en paralelo, etc.

Otros conceptos a tener en cuenta en el modelo de coordinación de un sistema groupware son los siguientes:

- Coordinación a **nivel de actividad** o a **nivel de objeto** (activity-level and object-level coordination)

En el nivel de actividad el modelo de coordinación describe la secuencia de actividades que constituye un procedimiento.

En el nivel de objeto el modelo de coordinación describe cómo trata el sistema el acceso secuencial o simultáneo de varios participantes al mismo conjunto de objetos.

- Multiple endeavor y single endeavor

Algunos sistemas no permiten tener múltiples endeavors. Sí es cierto que se pueden lanzar diferentes instancias del programa, pero no estarían relacionadas. Si las instancias del programa tienen conciencia las unas de las otras, se dice que permite tener múltiples endeavors.

- **Escenario e inspección de escenarios** (stages and inspection of stages)

Cada momento, cada instante de la colaboración se denomina escenario. Es una descripción global de las instancias de actividades que están activas.

Hay diferentes maneras de inspeccionar un escenario de un sistema que no es single-stage y que tiene múltiples endeavors. La **inspección basada en el participante** (participant-based inspection) proporciona información de todas las instancias activas de actividad de todos los endeavors de las que un participante en particular es responsable. La **inspección basada en endeavors** (endeavor-based inspection) proporciona la información de escenario correspondiente a un endeavor particular. La **inspección total** (total inspection) es una combinación de las dos anteriores. La **second-order inspection** proporciona información estadística de varios endeavors como el tiempo medio en completar un actividad, el tiempo medio en completar un procedimiento, etc.

- **Niveles de concurrencia** (levels of concurrency and currency)

Si las actividades se realizan una tras otra se dice que es **secuencial**. Si las instancias de actividad se pueden ejecutar simultáneamente en diferentes endeavors (no puede haber interferencia entre objetos por lo tanto) se dice que es en **paralelo**. Se habla de **additive concurrent** cuando hay tareas que se realizan simultáneamente en el mismo endeavor, pero sobre diferentes objetos. En este tercer nivel de concurrencia aparece el problema de la actualización de objetos que están siendo accedidos por diferentes usuarios: **currency**. Los cambios realizados por un participante en un objeto no se han propagado al resto de participantes del sistema. Surgen entonces términos como **tiempo de notificación**, actualización **automática** o actualización **manual**. Por último, cuando dos o más tareas simultáneas acceden al mismo conjunto de objetos se habla de **fully concurrent**.

Modelo de Interfaz de Usuario

La aparición de los sistemas groupware cambia la perspectiva de los sistemas informáticos. Se pasa de un sistema para resolver problemas a un sistema en el que además se facilita el trabajo entre los usuarios. Por lo tanto, este cambio se debe ser reflejado en los interfaces de usuario, que al fin y al cabo son el medio por el que los usuarios hacen uso de la lógica de negocio. A estos interfaces los llaman **interfaces de usuario de grupo** (group user-interface). El modelo de interfaz de usuario en este caso debe facilitar la interacción entre humanos, para lo que identifican tres componentes:

- Las vistas de objetos y operaciones locales

Los interfaces de usuario deben representar los objetos y operaciones comentados en este modelo ontológico teniendo en cuenta que diferentes participantes podrían tener diferentes capacidades o perspectivas. Estas vistas mostrarían, por tanto, los mismos objetos de formas diferentes.

- Las vistas de participantes

Son vistas que proporcionan a los participantes información y conciencia sobre otros participantes. En un sistema groupware se trata de un asunto fundamental. Por ejemplo, si en un sistema groupware de comunicación se sabe que una persona está presente en un momento determinado, el resto de personas producirán mensajes diferentes a los que producirían si no estuviera.

- Las vistas de contexto

Estas vistas muestran información sobre el contexto en el que se da la colaboración. Es importante tener en cuenta que el usuario no forma parte de ese contexto. El contexto lo forman los objetos y condiciones que están en el background. Los participantes están en el foreground. La elección de qué información contextual presentar y cómo presentarla es un reto. Sin embargo, han categorizado esta información en *estructural*, *social* u *organizacional*.

La **información de contexto estructural** se refiere al qué y al dónde de esa información.

La **información de contexto social** incluye elementos como normas de grupo, métricas de grupo o historia social del grupo.

La **información de contexto organizacional** está relacionada con las estructuras organizativas de los usuarios y otros elementos como las reglas de organización, etc.

3.2 Estructuras Organizativas y Grupos de Usuarios

Conceptos como los de *grupo*, *rol* o *actor* se han utilizado con mucha frecuencia en el análisis y diseño de sistemas complejos en los que existen muchos tipos de usuarios que interaccionan entre sí. Por ejemplo, en [Traetteberg, 1999] se emplean en el modelado de las tareas de los participantes de un flujo de trabajo de acuerdo con las cuatro dimensiones para describir el trabajo presentadas en [Marshak, 1997].

En algunas técnicas de modelado para el desarrollo de sistemas groupware [Paternò, 1999; Pinelle, 2003; Van der Veer, 2000] se consideran términos como *rol* o *actor* al modelar la colaboración existente entre los usuarios del sistema.

En el campo de los agentes, los *roles* permiten obtener modelos más expresivos. En [Odell, 2005] se afirma que sin considerar elementos organizativos, los diseñadores

no podrían aprovechar algunos beneficios tales como la escalabilidad. En [Mellouli, 2002] se propone un diagrama de agentes para mostrar los *roles*, sus *tareas* y *relaciones*. Estos y otros conceptos se emplean en [Van Dyke Parunak, 2001] para describir *estructuras organizativas* basadas en UML [Rumbaugh, 1999] para la especificación, el análisis y el diseño de sistemas multiagente.

Trabajos como FOAF (Friend-Of-A-Friend) vocabulary [Brickley, 2005] definen una serie de conceptos para expresar información sobre la *gente* y sus *relaciones*. Esta ontología RDFS/OWL se centra en describir a las *personas* y sus propiedades básicas como nombre, e-mail, etc. pero no en describir la *estructura organizativa* de los *usuarios* de un sistema y las tareas cooperativas que realizan.

Algunos entornos metodológicos para el diseño de sistemas cooperativos, como AMENITIES [Garrido, 2003], también consideran y definen conceptos relevantes para un sistema colaborativo. Esta metodología está basada en modelos de tareas y contempla aspectos dinámicos sobre cómo los usuarios pueden pasar de desempeñar un rol a desempeñar otro por medio de una serie de leyes y capacidades.

En CIAM [Molina, 2006], un marco metodológico para el diseño de la interfaz de usuario de sistemas groupware, se definen numerosos conceptos relacionados con la estructura organizativa de los usuario del sistema: *organización*, *rol*, *grupo*, *agente software*, *actor* y *equipo de trabajo*. Ver tablas de los conceptos de la metodología en el apartado 3.9.2 de este capítulo.

Así mismo, establece una diferenciación entre los conceptos de *grupo* y *equipo de trabajo*. Un *grupo* se entiende como un conjunto de personas (actores que desempeñan roles) que pertenecen a una misma organización o que participan en la realización de tareas cooperativas o colaborativas. Mientras que un equipo de trabajo es un grupo de trabajo que cumple ciertas características:

- Trabajan en una tarea común.
- Interactúan en un entorno compartido.
- Comparten e intercambian información.
- Deberían complementarse en sus habilidades.
- Existe una cierta disciplina o reglas de comportamiento entre ellos.
- Conocen al resto de miembros del grupo: sus destrezas, competencias y roles.
- Existe una necesidad de coordinación en la comunicación y actividades del grupo.

Todos estos métodos facilitan la comprensión del sistema y permiten a los diseñadores conocer quién está haciendo qué, o qué tipo de características y funciones puede desempeñar un usuario. En todos ellos se han definido términos como grupo, rol, tarea, colaboración, etc. de acuerdo con las necesidades y la lógica de la técnica que emplean.

Wil van der Aalst habla de las estructuras organizativas desde el prisma de la coordinación [Aalst, 2004]. Uno de los aspectos más importantes en las estructuras organizativas es la separación de responsabilidades y poderes. Los miembros de la organización tienen *roles* diferentes y además pueden tener roles diferentes a lo largo de su vida.

Las tres formas de estructura organizativa, o mecanismos de coordinación, son:

- Organización jerárquica (hierarchical organization)

Es la más conocida de las tres y se basa en estructuras arbóreas conocidas como diagramas organizativos, en los que las hojas (leaf, leaves) representan

grupos o departamentos, no roles; y las divisiones en las ramas (branches) muestran las relaciones de poder: una persona por encima de otra en el árbol, está autorizada a mandar trabajos a las que tiene por debajo.

- Organización matricial (matrix organization)

Se estructura de acuerdo a dos dimensiones: la funcional y la jerárquica. La parte jerárquica funciona de forma parecida a la que ya se ha mencionado anteriormente: las personas con la misma habilidad, experiencia... pertenecen al mismo grupo. La parte funcional se basa en las tareas que se tienen que ejecutar. Esas tareas que se tienen que realizar se llaman proyectos y los jefes encargados de que se lleven a cabo son los líderes de proyecto.

Por ejemplo, de arriba abajo podrían ser supervisores, carpinteros, albañiles pintores, enyesadores y de izquierda a derecha proyecto 1, proyecto 2, proyecto 3.

- Organización de red (malla) (network organization)

Actores autónomos colaboran proporcionando productos o servicios dando la sensación de que existe una organización. Es una organización virtual.

Hablando en términos de workflow, para la WfMC (se verán más términos de esta referencia cuando se hable de *workflow* y de *tarea*) [WfMC, 1999] un participante en el flujo de trabajo, *participante* o *actor* es un recurso que ejecuta el trabajo representado por una *instancia de actividad*. El trabajo se manifiesta como uno o más *elementos de trabajo* asignados al *participante* vía *lista de trabajo*. Normalmente se piensa en *recursos humanos* pero podríamos estar también hablando de otros *recursos* como por ejemplo *agentes inteligentes*. Se puede identificar directamente con una *definición de proceso de negocio* o, como es más habitual, como referencia a un rol de la entidad organizacional en la *definición del proceso*. Así puede haber más de un recurso disponible para el sistema de workflow para operar con ese rol durante la *ocurrencia del proceso*. Se identifican cuatro tipos de *participantes*: (1) *human*, (2) (machine) *resource*, (3) *role* y (4) *organizational_unit*. Algunos sinónimos de participante para la WfMC son: Actor, Agent, Player, User, Role Player, Work Performer.

La WfMC considera el concepto de *rol organizativo* (*organizational role*). Se trata de un grupo de *participantes* con un conjunto de atributos, capacidades y habilidades. Las *actividades* o los *elementos de trabajo* pueden requerir un recurso, un *participante* para llevar a cabo la tarea, con un determinado conjunto de atributos. Algunos sinónimos utilizados como rol organizativo son: Role, User Groups, Organisational Groups.

Para la WfMC un *rol de proceso* es un mecanismo que asocia *participantes* a una colección de *actividades* de workflow. Un *participante* asume un rol para acceder y procesar un trabajo de un *WMS*. El *rol de proceso* define el contexto en el que los *usuarios* participan en un *proceso* o una *actividad*. Sinónimos: Role, Activity Group, Workflow Performer Definition.

Incluso "afina" un poco más en la definición de algunos conceptos explicando que un *administrador* o *supervisor* (administrator) es un *usuario* del sistema workflow con privilegios especiales. Puede configurar el sistema y controlar y gestionar funciones. Puede haber varios *administradores* para compartir esas tareas, tomando cada uno de ellos la responsabilidad de un área de administración. Algunas funciones de un *usuario administrador* serían:

- Configurar y gestionar nombres de *usuarios*, passwords y *roles*.
- Asignar o reasignar *elementos de trabajo*.
- Procesar condiciones de excepción.

- Controlar *definiciones de procesos* o versiones de las mismas.
- Monitorizar el trabajo o el progreso de la *instancia de proceso*.
- Funciones de auditoría de sistemas.
- Etc.

La WfMC también contempla el concepto de *modelo organizativo* (*organizational model*) como un modelo que representa las entidades de una organización y sus relaciones. Incorpora conceptos como *jerarquía*, *autoridad*, *responsabilidades* o *atributos* asociados a *roles*. Algunos sinónimos son Role Model, Organisational Directory.

Otro investigador relevante en el área de la coordinación [Traetteberg, 1999] define algunos de estos conceptos:

- Los *actores* (actors) son seres intencionados que llevan a cabo acciones (acción es el concepto fundamental que describe el trabajo). El actor se caracteriza por un *estado* mental, que guía su comportamiento (*state*), *responsabilidades* (responsabilities) y *compromisos* (commitments). Los actores no sólo ejecutan acciones, sino que deciden qué acciones ejecutar. Técnicamente se pueden ver como recursos con ciertas características y habilidades que son necesarias para que se puedan ejecutar las tareas. Listando características como derechos, privilegios, conocimientos y habilidades, las acciones especifican de forma implícita quién las puede ejecutar.
- Los *grupos* están formados por una composición de actores. Un grupo completo podría llevar a cabo una acción compuesta, es decir, el grupo sería el conjunto de actores que ejecutan las subacciones de la acción compuesta.
- Los *roles* son el conjunto de características comunes de un conjunto de actores. Estas características son las típicas que necesitan las acciones para se ejecutadas. Se podrían los roles como el conjunto de recursos que necesitan las acciones, características que tienen que cumplir unos actores para que se les permita realizar la acción. Los roles llevan entonces a cabo las acciones. Un actor podría desempeñar diferentes roles.
- Los *objetos* son aquellas cosas materiales, mentales o representacionales que constituyen el dominio en el que se llevan a cabo las acciones. Los objetos se emplean para modelar información en torno a las acciones.
- Una *herramienta* es un caso especial del término general recurso. Las herramientas son aplicaciones o componentes del sistema integrado. Pueden ser concretas (Eudora) o abstractas (cliente de correo electrónico).

En la Tabla 1 se puede observar un resumen de la relación de algunos de estos conceptos definidos con otros del entorno del workflow y análisis de tareas.

Tabla 1. Conceptos de workflow y análisis de tareas

Workflow concept			Task interpretation
Generic	Abstract	Composite	
Action: basic unit of work, data flow or speech act based	Delayed definition, action template parameter	Provides hierarchical composition, also called process, job, activity	Task, task/subtask hierarchy • Data availability defines necessary pre-conditions • Work practice as additional constraints
Actor: intentional beings performing actions	Role: Intentionally defined set of actors, often based on actor characteristics	Group: Extensionally defined set of actors	User, user group, stereotype • Multi-user task models • Multiple work practices • Design targeted at different user groups
Object: part of (representation of) domain		Part-of hierarchy	Object • Operated upon • Contextual information
Tool: software needed for performing actions	Application class	Application suite, integrated components	Dialog element • Link between specification and design • Task based enactment • Component composition

En el Capítulo 5 se emplean unas plantillas para la recolección de datos relevantes sobre un proyecto que se quiere desarrollar. Es ahí donde se enumeran algunas de las características fundamentales que se han de tener en cuenta acerca de roles, tareas, etc. Sin embargo, existe una serie de metadatos (mostrados bajo estas líneas) que también describen a los roles, pero que están relacionados con el contexto y el entorno del sistema [Constantine, 1999]. Estos metadatos también se han empleado en otras metodologías como en Wisdom [Nunes, 2001]:

- *Características sociales.* En sociología el término empleado en inglés es *incumbents*. Personas tienen una serie de características, que actúan con un determinado rol social, se dice que ese es su *role incumbent*. Se trata de una serie de características propias de la persona (del actor) que va a desempeñar el rol y que se presuponen en un actor que desempeñe este rol. Por ejemplo, en el rol *Profesor_de_Matemáticas* se espera que las personas que vayan a desempeñarlo tengan conocimientos de notaciones matemáticas.
- *Destreza.* La destreza (*proficiency*) describe el nivel de habilidad en operación con el uso del sistema que tiene el usuario que desempeña este rol. Constantine define tres niveles de destreza: novel, intermedio o experto. El diseño del sistema, y más concretamente su interfaz, podría variar notablemente según el actor objetivo.

- *Patrón de uso*. El patrón de uso (*role interaction profile*) es un metadato que proporciona información sobre el comportamiento de los usuarios en ese rol. Es un metadato que se describe por medio de otros ocho.
- *Frecuencia*: cuán a menudo el usuario desempeña el rol.
- *Regularidad*: ¿hay un periodo regular de utilización o es más o menos esporádico?
- *Continuidad*: ¿la interacción en este rol es continua o intermitente?
- *Intensidad*: ¿cuál es el ratio de interacción?
- *Complejidad*: ¿cómo son de complejas las interacciones en este rol?
- *Predecibilidad*: ¿las interacciones en esta rol son predecibles o muy variables?
- *Foco del control*: ¿la interacción está dirigida por el proceso o por el usuario?
- *Información*. Dónde se origina la información y cómo fluye entre el usuario y el sistema tiene importantes implicaciones para el diseño de la interfaz del usuario. De nuevo se trata de un metadato que se describe por medio de varios.
- *Entrada*: ¿de dónde proviene la información?
- *Sentido del flujo*: ¿la información proviene del actor o sale de él?
- *Volumen*: ¿cuánta información está disponible y es del interés del actor?
- *Complejidad*: ¿cómo es de compleja la información disponible y útil para el actor?
- *Criterios de usabilidad*. Es una pequeña especificación de objetivos de usabilidad específicos con respecto al rol.
- *Soporte funcional*. Funciones o características necesarias para que el actor desempeñe el rol.
- *Riesgo operacional*. Nivel de riesgo asociado a la interacción con el sistema por parte de un actor que desempeñe este rol.
- *Restricciones de dispositivo*. Limitaciones del equipo físico que maneja el actor que juega este rol.
- *Entorno*. Factores relevantes del entorno físico donde un usuario que desempeña este rol interactúa con el sistema.

3.3 Colaboración, Cooperación, Comunicación y Coordinación

Los términos *comunicación*, *coordinación* y *cooperación* se han utilizado y asumido como pilares del CSCW casi desde sus inicios [Grudin, 1994; Poltrock, 1994; Target, 1997; etc.]. Sin embargo, su uso y definición no han estado exentos de polémica y ambigüedades en los diferentes trabajos. Concretamente los conceptos de ***cooperación*** y ***colaboración***.

Según la Real Academia Española [RAE, 2005], *colaborar* es trabajar con otra u otras personas en la realización de una obra, y se suele emplear *cooperar* con la misma semántica. *Cooperar* es obrar juntamente con otro u otros para un mismo fin. Aunque en principio parecen sinónimos, sin embargo, la realidad es que podría haber una pequeña diferencia cuando hablamos de CSCW.

Steven Poltrock y Jonathan Grudin son dos experimentados investigadores, referentes en el área. Durante muchos años han realizado tutoriales en numerosos congresos para aclarar conceptos relacionados con CSCW y groupware. Grudin es autor de dos de los artículos más referenciados [Grudin, 1994; Grudin, 1994 (b)]. En [Grudin, 1994], se matiza esta diferencia. Grupos pequeños de usuarios que comparten objetivos clave, se asume que *cooperan* entre ellos. En grandes organizaciones donde los objetivos, las metas normalmente difieren o incluso entran en conflicto, *colaboran* entre sí.

Se podría decir que en la cooperación el trabajo realizado se lleva a cabo de una forma más cercana, con más implicación, más en tiempo real.

"*Sólo se aguanta una civilización si muchos aportan su colaboración al esfuerzo. Si todos prefieren gozar el fruto, la civilización se hunde*" (José Ortega y Gasset). Sumando esfuerzos se avanza de un modo más eficiente y efectivo que de forma individualista y en las organizaciones se progresa como grupo en el desarrollo de todo proyecto e incluso en el desarrollo de la propia organización.

Curiosamente se ha definido CSCW, Computer-Supported *Cooperative Work*, en términos de comunicación, coordinación y *cooperación*. La pregunta siempre ha sido: ¿por qué no *colaboración* en lugar de *cooperación*?. Y más aún, parece que se trata de definir un concepto haciendo uso de él mismo. Quizás sea por este motivo por lo que del tutorial [Poltrock, 1994] en 1994 al tutorial [Poltrock, 1999] de 1999 o el [Poltrock, 2005] en 2005, Poltrock y Grudin dejaron de emplear estos términos cuando hablaban de los pilares del CSCW, para hablar de **compartición de información** (*information sharing*).

El groupware facilita la compartición de la información, por ejemplo, a través de sistemas de gestión documental: un repositorio donde se almacenan los documentos y pueden ser accedidos por los usuarios fácilmente, se controla el acceso a la información mediante privilegios, llevan a cabo un control de versiones, procesos de publicación, protección y desprotección de documentos, búsquedas avanzadas sobre múltiples tipos de formatos, posibilidad de aprobación de documentos por diferentes miembros, etc.

Otro concepto importante es la **comunicación**. Comunicar, según la RAE [RAE, 2005] es la acción y efecto de hacer a otro partícipe de lo que uno tiene; de descubrir, manifestar o hacer saber a alguien algo; de conversar, tratar con alguien de palabra o por escrito.

La comunicación entre las personas, entre los miembros de una organización, se ha de dar para enviar y recibir información, solicitudes, instrucciones, para estar al corriente del estado de la empresa, para estar al tanto de noticias de última hora que pueden afectar directamente a los trabajadores, etc.

Toda entidad que interaccione y trabaje en grupo, basa el éxito de sus negocios, entre otros factores, en una buena comunicación: "*la comunicación es la llave del éxito*" (Eduardo Aguilar Kublí).

Como se apunta en [Poltrock, 1999], los ordenadores se están convirtiendo en dispositivos de comunicación muy potentes. Cada vez más, la gente se comunica por medio de los ordenadores en lugar de usar dispositivos de comunicación tradicionales como el teléfono. Los investigadores se afanan por entender cómo se puede realizar una comunicación más efectiva mediante nuevos sistemas como la video-conferencia. El correo electrónico es una herramienta que hoy en día se asume como completamente implantada en todo negocio y que mejora los flujos de información y reduce los costes de coordinación.

Por último, la **coordinación** es una tarea fundamental en una organización. Concertar medios, esfuerzos, etc., para una acción común. Según Wil van der Aalst,

los trabajos se realizan de forma conjunta entre diferentes miembros de la organización, que participan en procesos estructurados. Es lo que se conoce como **workflow** [Aalst, 2004]. El centro del workflow son los procesos de negocio, procesos estructurados o semiestructurados y son estos procesos los que hay que coordinar.

La Workflow Management Coalition (WfMC) es una organización sin ánimo de lucro que pretende hacer avanzar el workflow a través del desarrollo de unos estándares y una terminología comunes. En el documento Terminology & Glossary [WfMC, 1999] define estos conceptos. También se retomará concienzudamente el concepto de workflow en el apartado siguiente, debido a su relación con términos importantes del CSCW como *tarea* o *actividad*.

Para la WfMC, el workflow, traducido como *flujo de trabajo* es la automatización de un proceso de negocio, completo o parte de él, donde documentos, información o tareas pasan de un participante a otro para realizar acciones de acuerdo con un conjunto de reglas de procedimiento. La automatización de un proceso de negocio se define en una definición de proceso, que identifica varias actividades de proceso, reglas de procedimiento e información de control asociada que se usa para gestionar el workflow durante el proceso. Identifican como sinónimos *Workflow Management*, *Workflow Computing* y *Case Management*.

Un *Sistema de Gestión de Workflow* define, crea y gestiona la ejecución de workflows a través del uso del software, ejecutando uno o más de un motor de workflow, que es capaz de interpretar la definición del proceso, interactuar con los participantes del workflow e invocar el uso de herramientas y aplicaciones del ámbito de las IT cuando fuese necesario.

Poltrock y Grudin [Poltrock, 1999] indican que los modelos de negocio tratan de ser más competitivos y para ello se han de aplicar técnicas que incrementen la calidad y disminuyan los costes de los procesos internos de una organización. Los sistemas groupware de coordinación pueden capturar y coordinar esos procesos. Así pues, la dimensión coordinativa del groupware puede hacer estos procesos más eficientes por medio del workflow. En este sentido, flujos de tareas y subtareas, roles, etc. se orquestan, se *coordinan*. Los sistemas de gestión de workflow modelan la secuencia de subtareas de un proceso y los roles que desempeñan los participantes del mismo y cada vez que una subtarea finaliza, el trabajo se encamina automáticamente hacia la persona responsable de la siguiente subtarea.

3.4 Características Espacio-Temporales de los Sistemas Groupware

En el capítulo anterior discusión acerca de la clasificación de las aplicaciones groupware que, normalmente, se basan en la matriz espacio-temporal ideada por Johansen [Johansen, 1988] y que se muestra en la 0. Por lo tanto, una de las propiedades que distinguen a los sistemas colaborativos son las relativas al tiempo y al espacio, como se detalló en el apartado 2.4.1.1.

De esta clasificación se destacan nuevos conceptos que se utilizan con frecuencia en el entorno de los sistemas colaborativos. La clasificación categoriza las herramientas según el *lugar* y el *tiempo*.

Cuando se habla de sistemas groupware, se han de tener presentes estos dos términos, pues hablamos de herramientas en las que participan más de un usuario, con lo que la cooperación/comunicación/etc. entre ellos puede variar en esas dos dimensiones. Son características propias de este tipo de sistemas, puesto que en los sistemas monousuario no tiene sentido hacer tal distinción.

Recordando lo mencionado acerca del *lugar* y el *tiempo* en el capítulo anterior, surgen algunos conceptos importantes que merece la pena destacar.

Por un lado, dependiendo de dónde se encuentren físicamente los usuarios del sistema groupware, del *lugar*, la interacción persona-ordenador-persona se puede dar en el *mismo lugar* físico, como podría ser una sala de reuniones, el salón de una conferencia o un lugar de trabajo común; pero también se puede dar en la distancia, es decir, en *lugares diferentes*. Algunos ejemplos serían las salas de video conferencia, el uso de editores colaborativos o las pizarras compartidas.

Por otro lado, la dimensión temporal de estos sistemas facilita una distinción entre aquéllos en los que la interacción se da en tiempo real como la, tan de moda, telefonía IP, o los chats; y aquellos otros en los que el tiempo no juega un papel relevante, o al menos no en exceso, caso del correo electrónico, control de versiones, agendas, etc. Se trata de sistemas *síncronos* y *asíncronos* respectivamente. En [Ellis, 1991] se define la interacción síncrona como aquella en la que los usuarios interactúan en tiempo real, mientras que la interacción asíncrona es aquella en la que los usuarios interactúan sobre un determinado periodo de tiempo.

También la WfMC define un buen número de conceptos ligados a aspectos temporales, especialmente, relacionados con la ejecución de procesos. Algunos de esos términos han sido explicados en el apartado anterior al detallar el concepto de proceso como procedimiento de ejecución de tareas: El *tiempo de finalización* o *deadline*, ejecución en *paralelo* o *secuencial*.

3.5 Tareas y Tareas Cooperativas

El concepto de tarea y todo lo que gira en torno a él es un tema que se ha discutido mucho en diferentes campos de investigación. Por ejemplo, este concepto se ha tenido en cuenta en el diseño de sistemas basados en la interacción del usuario con las máquinas (HCI), en el desarrollo de interfaces de usuario basado en modelos (MB-UIDE), estudio y desarrollo de flujos de trabajo (workflow), etc.

Debido a esta variedad de investigaciones y usos, la definición de tarea difiere ligeramente según el entorno en el que nos movamos.

Las aplicaciones informáticas actuales combinan la colaboración entre los usuarios para realizar determinadas tareas y el uso de la Web como infraestructura. La especificación de los sistemas ha de tener en cuenta, por tanto, características especiales propias de estos sistemas.

Mecanismos como los descritos en [Paternò, 1999; Pinelle, 2003; Van der Veer, 2000] permiten mostrar la organización de las tareas que realizan los usuarios del sistema para proporcionar al diseñador una idea más clara de qué hace quién, de qué manera se ha de realizar algo, etc.

Con el metamodelo que proponemos en este trabajo se describen las tareas que se han de realizar en el sistema en profundidad para informar al diseñador de cómo son.

En este apartado se muestran algunas aproximaciones al concepto de tarea, al modelado de sistemas que se basan en las tareas, etc.

3.5.1 Técnicas tradicionales del análisis de tareas en sistemas centrados en el usuario

Algunos de los trabajos más antiguos acerca del análisis de tareas se centran en el estudio de lo que los usuarios deben hacer, pero de una forma más individualizada, puesto que la idea de grupo en la investigación tecnológica aparece posteriormente.

En [Garrido, 2003] se resumen algunos aspectos comunes a las primeras técnicas ideadas para realizar análisis de tareas:

- Descomposición en tareas del trabajo que se tiene que realizar.
- Especificación del flujo de la tarea, es decir, el orden que ha de seguir cada una de ellas con relación a las demás.
- Modelado de objetos, la descripción de los objetos del dominio que aparecen en el sistema relacionados con cada una de las tareas.
- Paso de información, es decir, la descripción de las modificaciones del sistema cuando se ejecutan las tareas: estados iniciales, finales, etc.

En 1967 surge HTA (Hierarchical Task Analysis) [Annett, 1967]. Es una técnica muy sencilla para la descripción de tareas en base a operaciones y planes. Las **operaciones** son actividades que realizan los usuarios para lograr una meta, mientras que los **planes** son las condiciones necesarias para poder realizar esas actividades. Se trata de una representación gráfica una descomposición jerárquica de las tareas y subtareas en forma de sencillos árboles.

En 1983 aparece GOMS (Goals, Operators, Methods, Selection rules) [Card, 1983]. Es una familia de técnicas de modelado de interfaces de usuario basado en el mecanismo de razonamiento humano para la resolución de problemas. Esta técnica permite describir cómo realizan las tareas los expertos así como la descomposición de las mismas.

Para cada tarea, GOMS describe:

- los **objetivos** (*goals*) y sub-objetivos de los usuarios. Lo que los usuarios quieren conseguir. Por ejemplo "borrar un fichero".
- los **operadores** (*operators*) que son acciones básicas, pequeñas o incluso atómicas, que llevan a cabo los usuarios y que provocarán un cambio en el sistema o en el modelo mental del usuario. Por ejemplo "localizar icono en la pantalla", "mover el cursor hacia un elemento", etc.
- los **métodos** (*methods*) necesarios para lograr esos objetivos. Una descomposición de un objetivo en sub-objetivos/operadores. Por ejemplo "arrastar un fichero a la papelera de reciclaje".
- las **reglas de selección** (*selection rules*) que permiten seleccionar un método u otro en cada momento.

La Figura 24 muestra un ejemplo sobre la descripción en GOMS de la tarea de borrado de un archivo en DOS.

<ul style="list-style-type: none"> • Goal: Delete a File • Method for accomplishing goal of deleting a file <ul style="list-style-type: none"> - retrieve from Long term memory that command verb is "del" - think of directory name & file name and make it the first listed parameter - accomplish goal of entering & executing command - return with goal accomplished
--

Figura 24. Ejemplo sobre la descripción en GOMS de la tarea de borrado de un archivo en DOS

El gran problema de GOMS es que sólo permite describir tareas de tipo secuencial.

Algo más reciente, pero también centrada en el usuario, es la técnica UAN (User Action Notation) [England, 1998] que surge en 1998. Hix y Hartson desarrollan esta técnica que será refinada posteriormente por Hartson y Gray para mostrar el emparejamiento de las tareas con los dispositivos de la computadora. Describen las tareas en base a las acciones que realiza el usuario, la retroalimentación de la interfaz y el estado en el que queda el sistema tras su ejecución. Esta técnica emplea una notación específica. Algunos de los elementos que incluye en la descripción de las tareas son:

- Símbolos, operadores, condiciones y opciones.
- Tablas de acciones de usuario, retroalimentación y estado del sistema.
- Relaciones temporales y restricciones.

El ejemplo de la muestra la tabla que describe la tarea de enviar un archivo a la papelera de reciclaje.

Tabla 2. Ejemplo de tabla que describe la tarea de enviar un archivo a la papelera de reciclaje en UAN

USER ACTION	FEEDBACK	SYSTEM STATE
mouse_down(x,y)		if intersect(icon,x,y) icon = selected
drag_icon(x,y)*	then highlight(icon) show_outline(icon)	
	then highlight(bin)	if intersect(bin,x,y)
mouse_up(x',y')	then hide(icon) show_bin_full()	if intersect(bin,x',y')

En el desarrollo de sistemas centrados en la interacción de los usuarios con la máquina se ha tenido en cuenta las tareas, en concreto los modelos de tareas, como técnica empleada para su diseño. Aparecen estas y otras aproximaciones para el modelado de las tareas del usuario cuya representación suele basarse en [Markopoulos, 2000]:

- Descomposición en sub-tareas representadas en una estructura arbórea.
- Ordenaciones temporales entre las tareas.
- Las tareas están asociadas con representaciones del dominio de tareas en términos de objetos.

En algunos de los siguientes apartados se verán diferentes aproximaciones al modelado de tareas.

3.5.2 ConcurTaskTrees (CTT)

ConcurTaskTrees (CTT) [Paternò, 1999] es una notación para el modelado de tareas que permite diseñar aplicaciones interactivas. Este modelo de tareas proporciona el soporte requerido por aplicaciones cooperativas por medio de la descomposición del modelo en un modelo cooperativo y un modelo por cada rol presente en el sistema.

Esta notación está centrada en la tarea, no en las interacciones que tienen lugar entre los usuarios del sistema, no en el usuario como miembro de un grupo.

Proporciona estructuras jerárquicas que permite a los diseñadores razonar sobre distintos niveles de abstracción y mejora el proceso de diseño refinado; representa de forma sencilla gran variedad de relaciones temporales, muy importante para poder representar interfaces modernos en los que los entornos multimedia están presentes; y permite expresar la información de forma más detallada si fuera necesario, lo que permite modelar sistemas más complejos.

CTT proporciona una representación gráfica sencilla de la estructura jerárquica de las tareas y usa operadores de LOTOS para expresar las relaciones temporales entre tareas de un mismo nivel de abstracción.

Una **tarea** define cómo un usuario puede alcanzar una meta. Esta meta puede ser una modificación del estado del sistema o una consulta al mismo. En CTT las tareas se describen mediante una serie de atributos:

- *name* de la tarea, que se usa para identificarla;
- *type*, es decir, si es *abstracta*, de *usuario*, de *aplicación*, de *interacción*, o de *cooperación* (ver Tabla 4);
- *subtask of*, muestra el nombre de la tarea padre;
- *objects* es un vector con un nombre, un tipo, una lista de acciones objeto de entrada y una lista de acciones objeto de salida;
- *iterative* es un boolean que indica si la tarea es iterativa;
- *first action*: muestra el conjunto de acciones iniciales posibles;
- *last action*: muestra el conjunto de acciones finales posibles;

En cada nivel, la especificación de tareas se construye en dos pasos: (1) se identifican los objetos y (2) se definen las acciones que permiten la comunicación entre ellos.

El modelo de tareas se construye en tres fases:

1. Descomposición lógica jerárquica de las tareas representadas en una estructura arborescente.
2. Identificación de las relaciones temporales entre tareas del mismo nivel.
3. Identificación de los objetos asociados con cada tarea y de las acciones que permiten que se comunique con otros.

Las relaciones temporales entre tareas del mismo nivel se realizan por medio de una extensión de los operadores definidos en LOTOS. De esta manera, CTT permite definir tareas concurrentes. Los operadores que describen las relaciones temporales entre tareas son los que se muestran en la Tabla 3.

Tabla 3. Operadores temporales utilizados en CTT

Simbología	Operador	Descripción
T1 T2	<i>interleaving</i>	Las tareas se pueden realizar en cualquier orden
T1 [] T2	<i>choice</i>	Permite elegir entre dos tareas
T1 [[] T2	<i>synchronization</i>	Dos tareas se tienen que sincronizar para poder intercambiar información
T1 = T2	<i>order independence</i>	Las dos tareas se han de ejecutar, pero cuando una comienza, ésta debe finalizar antes de que empiece la otra
T1 [> T2	<i>deactivation</i>	Cuando una acción de la segunda tarea ocurre, entonces se desactiva la primera
T1 >> T2	<i>enabling</i>	Una tarea comienza cuando ha acabado la anterior
T1 []>> T2	<i>enabling with information passing</i>	Una tarea comienza cuando ha acabado la anterior, pero además le pasa información
T1 > T2	<i>suspend-resume</i>	Suspensión / reanudación de tareas
T1*	<i>iteration</i>	Se trata de una tarea iterativa
T1(n)	<i>finite iteration</i>	Se trata de una tarea iterativa, pero se especifica el número de veces que se ejecuta
[T1]	<i>optional task</i>	No es una tarea obligatoria
T	<i>recursion</i>	Posibilidad de incluir en la especificación de la tarea a la propia tarea

Tabla 4. Tipos de tarea en CTT

	<ul style="list-style-type: none"> • <i>Abstraction</i>. Acciones complejas que no son parte de una única categoría.
	<ul style="list-style-type: none"> • <i>Interaction</i>: Interacción entre el usuario y el sistema.
	<ul style="list-style-type: none"> • <i>Application</i>: Tareas completamente ejecutadas por el sistema.
	<ul style="list-style-type: none"> • <i>User</i>: Tareas ejecutadas únicamente por el usuario, sin interacción con el sistema.
	<ul style="list-style-type: none"> • <i>Cooperation</i>: Tareas compuestas de otras subtareas donde hay varios usuarios para realizar las tareas básicas.

Por ejemplo, si se quisiera modelar una pizarra compartida, es decir, la tarea cooperativa de pintar en una pizarra virtual, la Figura 25 y la Figura 26 representarían los modelos de tareas cooperativo y de role respectivamente haciendo uso de la notación CTT.

Como se puede apreciar en la Figura 25 en un sistema de pizarra virtual compartida, el sistema envía lo que cada usuario dibuja al resto (tarea de aplicación 'Send_picture') y de forma concurrente (operador ||| 'interleaving'), el sistema va mostrando cada usuario lo que dibujan los demás (tarea de aplicación 'Showing'). Estas tareas aparecerán otra vez en el diagrama de role puesto que realmente son tareas propias de los roles, aunque se muestran también en el diagrama cooperativo para representar el modelo cooperativo.

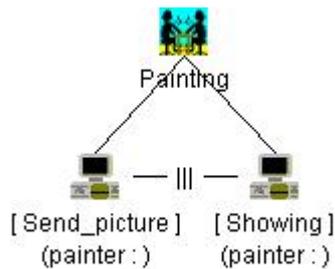


Figura 25. Diagrama cooperativo correspondiente a la tarea cooperativa de pintar en una pizarra virtual compartida en CTT

El diagrama de tareas de la Figura 26 representa el modelo de tareas correspondiente a un usuario que desempeña el rol 'painter'. La representación de los diferentes tipos de tarea y operadores que aparecen en la figura modelan el comportamiento de ese usuario.

Mediante la aplicación se puede escribir en la pizarra virtual (tarea abstracta 'Writing'), o bien es la aplicación la que muestra por pantalla información procedente de otros usuarios (tarea de aplicación 'Showing'), o bien da la posibilidad de salir de la misma (tareas 'Exit_decision' y 'Exit' respectivamente).

Un usuario que se dispone a dibujar en la pizarra virtual en primer lugar toma la decisión de hacerlo (tarea de usuario 'User_decisio), después la aplicación selecciona la herramienta por defecto (tarea de aplicación 'Default_tool'), o bien (operador [] choice), es el propio usuario el que decide qué herramienta de dibujo utilizar (tarea de interacción 'Select_another_tool'). Entonces el usuario dibuja en la pizarra virtual (tarea 'Write_on_screen') y esa información pasa al sistema (operador []>> 'enabling with information passing') que enviará los trazos al resto de usuarios (tarea de aplicación 'Send_picture').

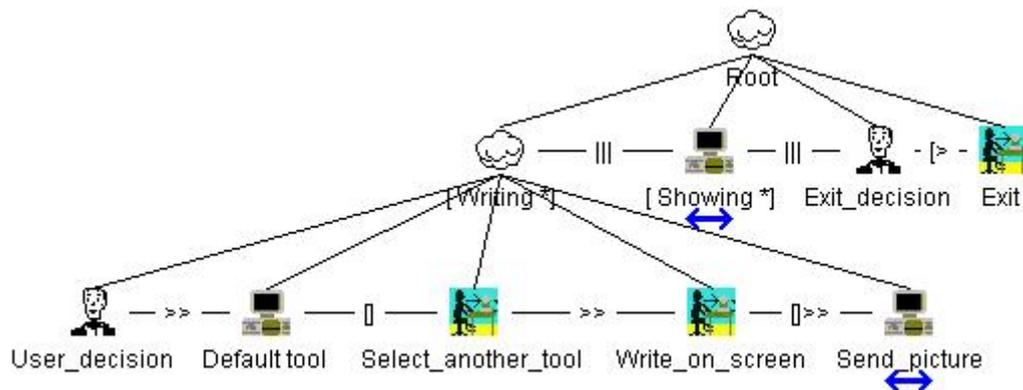


Figura 26. Diagrama que representa las tareas que ha de realizar un usuario que desempeñe el rol 'painter' para pintar en una pizarra virtual compartida

Como se puede apreciar en la Figura 26, las tareas 'Showing' y 'Send_picture' están marcadas con una doble flecha indicando que formarán parte del modelo cooperativo como se comentó con anterioridad.

3.5.3 Collaboration Usability Analysis (CUA)

Collaboration Usability Analysis [Pinelle, 2003], CUA, es una técnica de modelado para el desarrollo y evaluación de sistemas groupware. Los autores apuntan que hay dos requisitos fundamentales que todo método para el análisis de tareas debe cubrir para conseguir modelar correctamente el trabajo en equipo:

- Debe ayudar a los evaluadores a identificar problemas de usabilidad en el equipo de trabajo causadas por la interfaz groupware.
- Debe ayudar a los evaluadores a comprender y explorar el rango de posibilidades que ofrece el sistema groupware en cuestión.

CUA soluciona el primero de los requisitos por medio de los **mecanismos de colaboración** (mechanics of collaboration), que son los operadores básicos de los equipos de trabajo. Es decir, las **acciones** y las **interacciones** (actions and interactions) que un miembro de un grupo debe ejecutar para realizar una tarea en un escenario colaborativo.

El segundo requisito lo solventa capturando la **variabilidad** (variability) de la ejecución de las tareas de grupo. Dado que los miembros de un grupo trabajan de forma concurrente, se intercalan tareas individuales con tareas de equipo, se realizan divisiones del trabajo, se dan distintas composiciones de grupos, etc. Por ello, una técnica de modelado debe ser capaz de representar esa variabilidad.

3.5.3.1 Descripción General de CUA

La técnica de modelado CUA permite realizar evaluaciones de usabilidad sobre sistemas groupware. Los evaluadores pueden analizar estos sistemas por medio de la representación de las tareas compartidas.

Está centrada en los aspectos de trabajo en equipo en situaciones colaborativas, proporcionando representaciones de alto y de bajo nivel. Se pueden representar múltiples **actores** y sus **interacciones**.

Se trata de un modelo de tareas jerárquico que representa las tareas de grupo en un espacio de trabajo compartido.

La jerarquía de tareas incluye escenarios, tareas e instancias de tarea (ver Figura 27). Los **escenarios** describen el contexto de alto nivel de la situación colaborativa. Las **tareas** indican metas específicas en el escenario. A un nivel más bajo, las **instancias de tarea** representan medios individuales y colaborativos de llevar a cabo una tarea.

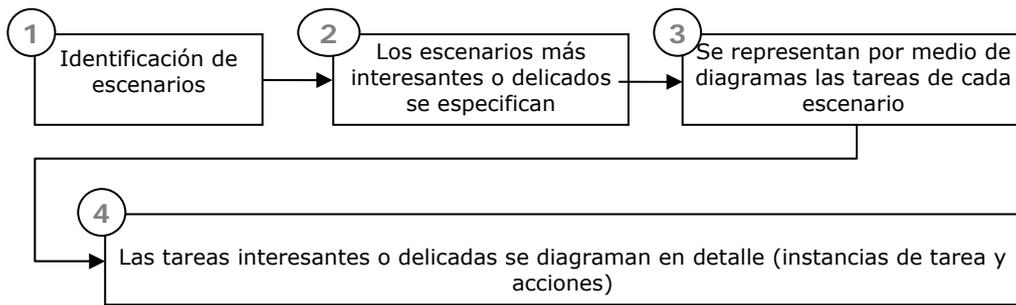


Figura 27. El método CUA

3.5.3.2 La Metodología

CUA comienza con la **identificación de escenarios** del mundo real que describen el trabajo que realiza el usuario. Estos escenarios sólo se identifican, no se analizan o especifican. La Figura 28 muestra un ejemplo de esta identificación de escenarios en la recepción de un centro de salud. Se ha mantenido en inglés para respetar el ejemplo original.

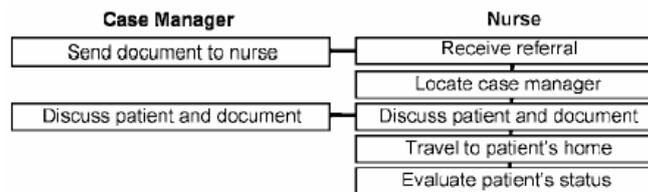


Figura 28. Paso 1: Identificación de escenarios

Posteriormente, aquellos escenarios que sean particularmente relevantes se especifican, tal y como se puede ver en la Figura 29, de forma textual. Un escenario consiste en la descripción de una **actividad de alto nivel**, una **especificación de usuario**, un **resultado esperado** de la ejecución de ese escenario y una serie de **circunstancias** bajo las cuales el escenario se ejecuta.

Scenario: Discuss patient and document

Activity description. A nurse receives a care plan and an assessment document from a referring case manager. The nurse reviews the assessment and wants to gather further information before visiting the patient, so she attempts to locate the case manager in the office. When the nurse finds the case manager, she approaches him and asks him for further information about the patient. The nurse places the assessment document on the desk and points to areas of the assessment where she has questions. The care manager and nurse discuss the patient's situation.

User specification. Case managers are responsible for coordinating community based patient care. They evaluate patients, create care plans, then refer patients to other community-based services. Case managers are fairly experienced with computer systems.

User specification. Nursing is responsible for providing a wide range of nursing services in the community. Nurses do not currently use computers in their jobs.

Intended outcome. Exchange information concerning patient. Nurse, in particular, wants further information about the patient before visiting them for the first time.

Circumstances. Nurses and case managers have office space in same building and usually try to meet face to face during the day.

Figura 29. Paso 2: Los escenarios más interesantes / complejos se especifican textualmente con detalle

De cada uno de esos escenarios importantes se extraen las *tareas* y se realiza un diagrama similar al de la Figura 30.

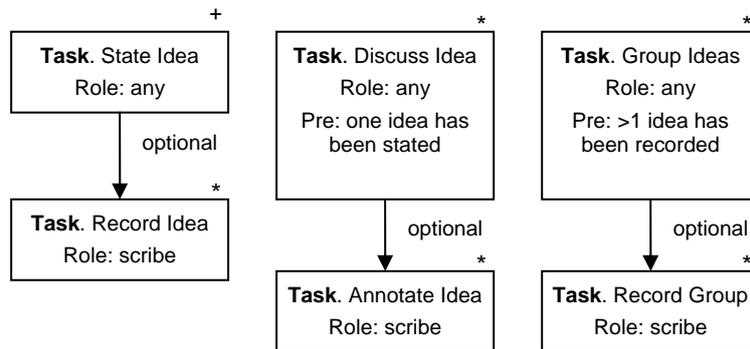


Figura 30. Paso 3: Las tareas de los escenarios principales se representan en forma de diagramas

Por último, las tareas más importantes se representan en detalle por medio de diagramas como el de la Figura 31. Así, las *instancias de tareas individuales y colaborativas* representarían *trabajos individuales o de equipo* respectivamente. Las *instancias de tarea colaborativa* representan los *componentes de trabajo de equipo* de las tareas y se especifican usando los *mecanismos de colaboración*. Cada *tarea colaborativa* se puede realizar a través de un conjunto de posibles *acciones*, es decir, las *acciones* describen cómo se llevan a cabo los *mecanismos*, descritos en la *instancia de tarea colaborativa*, en el mundo real.

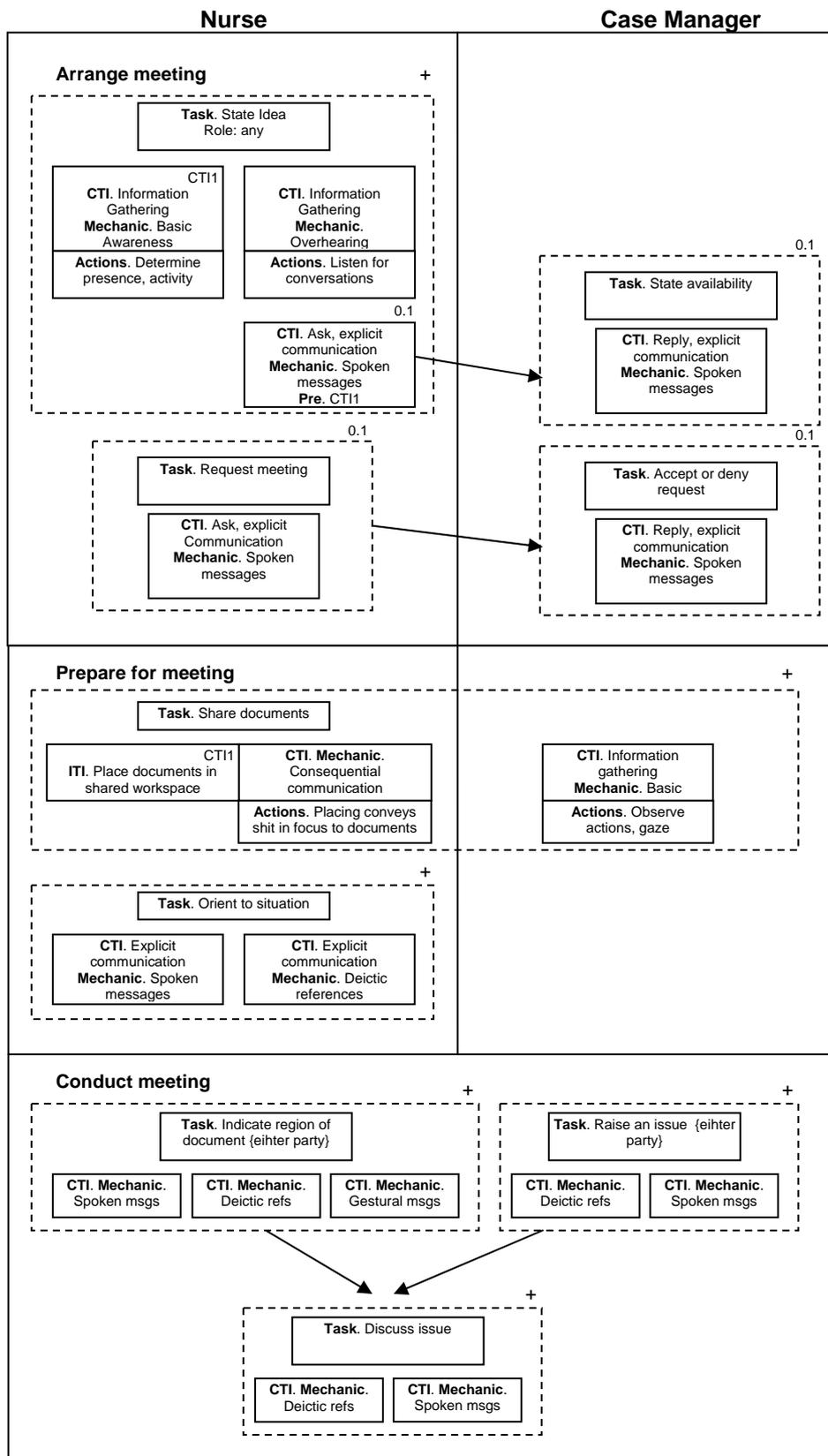


Figura 31. Paso 4: Las tareas más importantes se describen por medio de diagramas (instancias de tareas y acciones)

3.5.3.3 Mecanismos de Colaboración

Los mecanismos de colaboración una vista detallada del equipo de trabajo (fine-grained view of teamwork), especialmente útiles en el análisis para modelos de tareas orientados a la evaluación. Hay dos tipos de mecanismos de colaboración según estén relacionados con la *comunicación* o con la *coordinación*. A su vez, los mecanismos relacionados con comunicación pueden ser de *comunicación explícita* o de *recogida de información*; mientras que los relacionados con coordinación pueden ser de acceso compartido o de transferencia. La Tabla 5 muestra una clasificación de los mecanismos de colaboración en CUA.

Tabla 5. Mecanismos de colaboración en CUA

Category	Mechanic	Typical Action
Communication		
Explicit communication	Spoken messages	Conversational, Verbal shadowing
	Written messages	Conversational, Persistent
	Gestural messages	Indicating, Drawing, Demonstrating
	Deictic references	Pointing+conversation
	Manifesting actions	Stylized actions
Information gathering	Basic awareness	Observing who is in the workspace, what are they doing, and where are they working
	Feedthrough	Changes to objects, Characteristic signs or sounds
	Consequential communication	Characteristic movement, Body position and location, Gaze direction
	Overhearing	Presence of talk, Specific content
	Visual evidence	Normal actions
Coordination		
Shared access (to tools, objects, space, and time)	Obtain resource	Physically take objects or tools, Occupy space
	Reserve resource	Move to closer proximity, Notify others of intention
	Protect work	Monitor others' actions in area, Notify others of protection
Transfer	Handoff object	Physically give/take object, Verbally offer/accept object
	Deposit	Place object and notify

3.5.3.4 Captura de la Variabilidad

Las tareas de grupo se podrían llevar a cabo de múltiples maneras. CUA ayuda a los evaluadores a explorar este rango de posibilidades capturando la variabilidad.

Existen dos tipos de variabilidad: variabilidad en la *secuencia de tareas* y variabilidad en el *grado de acoplamiento entre los miembros del grupo*.

El primero de ellos concierne a la variabilidad que se puede dar por que la ejecución de las tareas ocurra en órdenes diferentes, repetida o incluso por parte de personas distintas. Un modelo debería soportar secuencias de tareas, dependencias, tareas alternativas, tareas opcionales, e iteración.

El segundo se refiere al trabajo llevado a cabo por las personas con otros miembros del grupo antes de realizar la colaboración.

3.5.4 Groupware Task Analysis (GTA)

Designing for Users and Tasks from Concepts to Handles (DUTCH) [Van der Veer, 2000] es un método práctico para el diseño de aplicaciones groupware. Es una aproximación basada en tareas que define un método, los modelos necesarios y las representaciones, así como una herramienta que le da soporte. Este método de diseño se basa en GTA (Groupware Task Analysis). Figura 32 muestra un resumen esquemático del proceso DUTCH (para más detalle, ver la Figura 33).

GTA proporciona un medio para analizar la situación actual en cuanto a tareas que se realizan en el sistema que se quiere modelar, lo que ayuda a realizar una elicitación de requisitos mejor. Entonces se realiza un rediseño de la estructura de tareas que incluye soluciones tecnológicas a los problemas planteados y respuestas tecnológicas a los requisitos. Tras el modelado de tareas se especifica y diseña el sistema. Hay tres subactividades importantes y relacionadas: la especificación de la funcionalidad, la estructura del diálogo entre los usuarios y el sistema y la especificación del modo en el que el sistema se presenta al usuario. En paralelo a todos los procesos se realizan pruebas de usabilidad y evaluación.

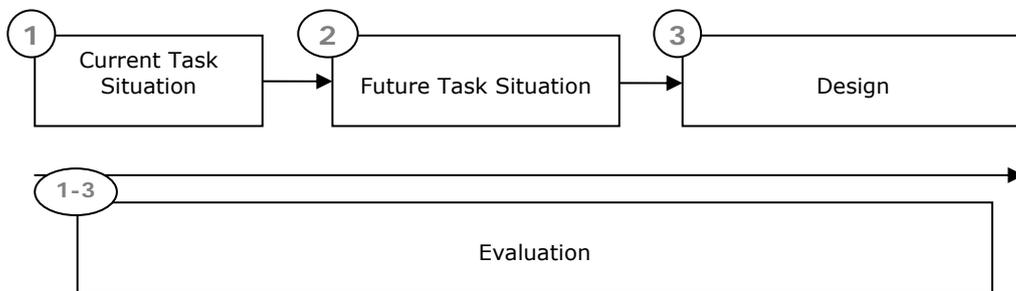


Figura 32. DUTCH method scheme

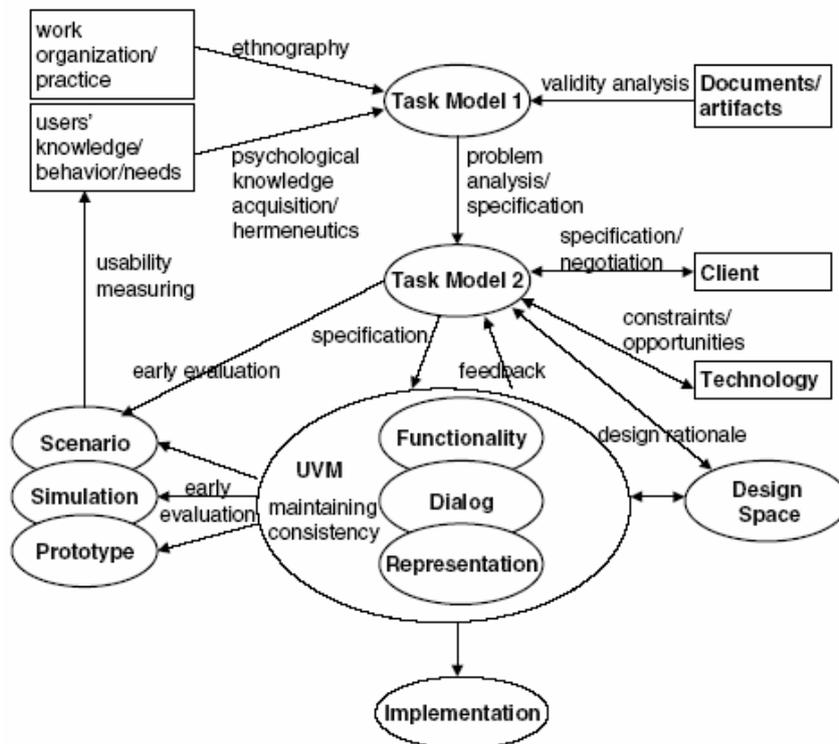


Figura 33. The DUTCH design process

GTA oferta diferentes representaciones del sistema, que están integradas y que permiten mostrarlo de diferentes maneras.

Un *marco conceptual* describe los agentes, el trabajo y la situación del mundo real. Hay una distinción entre los actores y el rol que juegan en el sistema. Una organización representa la relación existente entre los actores y los roles en la asignación de tareas. Los aspectos estructurales y dinámicos del trabajo se modelan por medio de las tareas (que tienen diferentes metas), las acciones y los eventos. La situación se estudia describiendo el entorno y los objetos del mundo real: las cosas físicas, los conceptos no materiales, etc.

La *estructura de trabajo* describe cómo la gente divide su trabajo. Se representa usando árboles de tareas en una descomposición jerárquica del trabajo.

Las *dinámicas de trabajo* se representan por medio de workflows o modelos de actividades: tareas paralelas y opcionales, eventos, colaboración y comunicación entre roles, tiempo, etc.

Los *diagramas de clase* se emplean para representar artefactos y herramientas.

Por último, imágenes, mapas, vídeos capturan partes del entorno de trabajo.

3.5.5 Tareas y Coordinación: un Estrecho Vínculo

La relación entre workflow y tarea siempre ha sido muy estrecha. Tanto es así, que al describir conceptos propios de cada uno de los dos ámbitos se solapan en muchas ocasiones. Esta relación la pone de manifiesto Traetteberg, que hace una discusión [Traetteberg, 1999] sobre la integración de los modelados de workflow y tareas, presentando y comparando algunos conceptos utilizados. Considera que el conocimiento del concepto *trabajo* es fundamental para construir sistemas de información con éxito. Y esto es así porque debe ser entendible por la *organización* y porque los sistemas de información se deben construir de acuerdo con los *objetivos* (objectives) y *metas* (goals) de la misma.

Identifica tres niveles en los que un sistema de información de una organización debería dar soporte al trabajo: son los niveles **organizativo** (organizational), de **grupo** (group) e **individual** (individual). El trabajo que se lleva a cabo en los dos primeros niveles se describe por medio de lenguajes de modelado de **workflow**. Por otro lado, para describir el trabajo individual se utiliza el **análisis de tareas**. Se entiende que para crear aplicaciones software útiles que den soporte al trabajo se han de integrar esos niveles, se han de integrar workflow y análisis de tareas.

Marshak identifica cuatro dimensiones para describir el trabajo [Marshak, 1997] y es en las que se basa Traetteberg: la **estructura de la acción** (action structure), los **actores** (actors) que llevan a cabo el trabajo, las **herramientas** (tools) que utilizan y la **información** (sources) que se necesita.

El concepto fundamental cuando se describe el trabajo es el de **acción** (action), que también se puede ver como **tarea** (task) o **acto** (act). Las acciones son la causa de todo cambio y constituyen los pasos necesarios para conseguir los fines (**goals**). Para que se pueda ejecutar una acción es necesario que se satisfagan algunas **pre-condiciones**.

Recuérdese que Traetteberg definía a un *actor* como a un ser intencionado que realiza *acciones*. También indica que *grupo* sería el conjunto de actores que ejecutan las *subacciones* de la *acción compuesta*, que los *roles* son un conjunto de características típicas que necesitan las *acciones* para se ejecutadas, o que los *objetos* son el dominio en el que se llevan a cabo las *acciones*.

En la Tabla 1 se puede observar un resumen de las definiciones sobre workflow y la interpretación de las mismas en un lenguaje más cercano al análisis de tareas. Con esto Traetteberg pretende ayudar a integrar esos niveles.

3.5.6 Algunos Conceptos Descritos en la WfMC

La WfMC también define muchos conceptos en torno al de tarea. Como se describió en el apartado 3.3, esta organización define la terminología propia del workflow que, como se ha expuesto en el apartado anterior, está sumamente relacionado con el análisis de tareas.

La Figura 34 relaciona de forma muy sencilla algunos de los conceptos básicos que define la Workflow Management Coalition [WfMC, 1999].

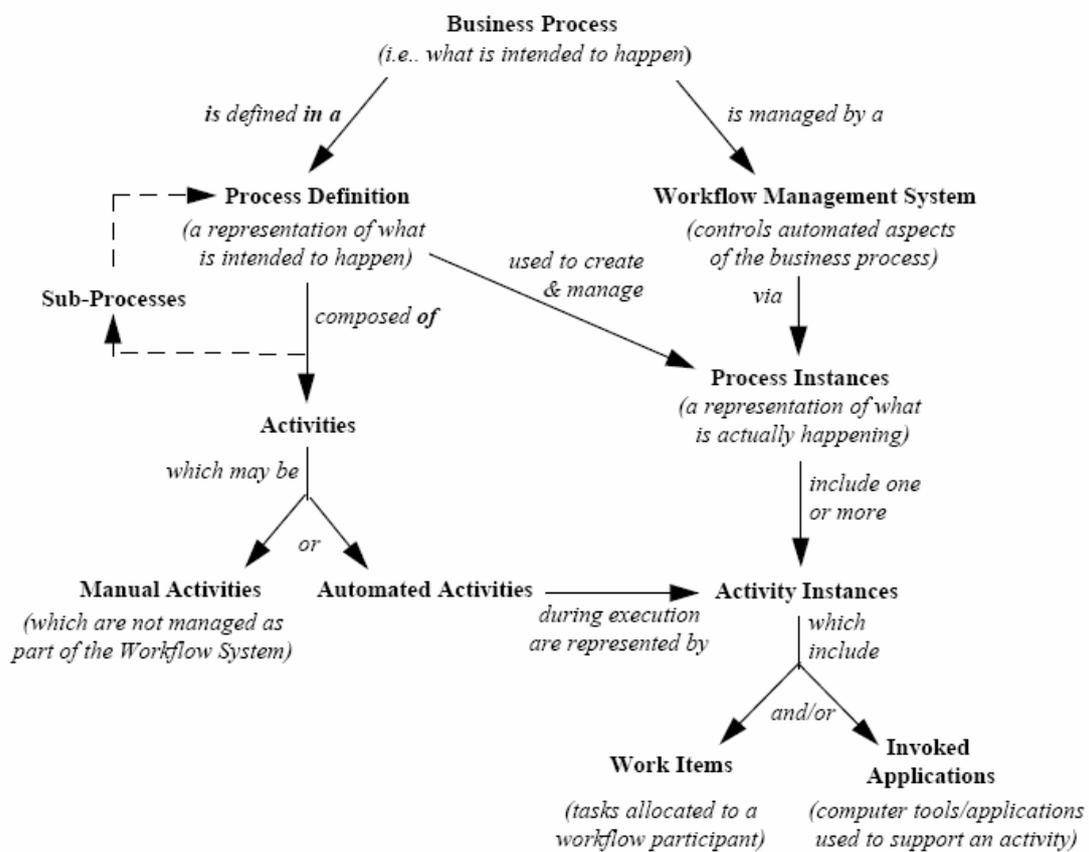


Figura 34. Relaciones entre la terminología básica

Los conceptos de *workflow* y *sistema de gestión de workflow* se comentaron en el apartado anterior. Para la WfMC, un **proceso de negocio** o *business process* es un conjunto de procedimientos o *actividades* que resuelven un objetivo, normalmente en el contexto de una estructura organizacional definiendo roles funcionales y relaciones. Un *proceso de negocio* define condiciones de iniciación y produce salidas. Puede conllevar interacciones formales o informales entre *participantes*. La duración puede ser variable. Un proceso de negocio puede consistir en *actividades automatizadas* capaces de gestionar workflow y/o *actividades manuales* que quedan fuera del alcance de la gestión del workflow.

La **definición de proceso** o *process definition* consiste en una red de *actividades* y sus *relaciones*, el criterio que indica la iniciación y la finalización del *proceso* e información sobre las *actividades* individuales como *participants*, IT applications asociadas, datos... La *definición de proceso* puede incluir tanto *actividades manuales*, como *actividades automatizadas*. Puede contener referencias a *sub-procesos* que además se definen de forma separada, pero que forman parte de la definición global. La *definición de proceso* puede hacer referencia a una *organización* o *modelo de recursos* separados para permitir que los *participantes* sean indirectamente definidos. Es posible encontrar estas otras expresiones como sinónimos de definición de proceso: Model Definition, Routing Definition, Flow Diagram, State Transition Diagram, Flow Schematic, Workflow Script, Instruction Sheet Definition, Case Type.

Una **actividad** (*activity*) es una descripción de una pieza de trabajo que forma un paso lógico en un *proceso*. Las *actividades* pueden ser (1) *manuales*, no se automatizan; o (2) *automatizadas*. Una *actividad de flujo de trabajo* requiere *recursos* humanos y/o de máquinas para ejecutar *procesos*. Cuando se requiere un *recurso* humano, la *actividad* se asigna a un *participante*. Una *definición de proceso* consiste en varias *actividades* lógicamente relacionadas que contribuyen en su conjunto a la realización del *proceso*. Es la unidad más pequeña de trabajo y es programada por el motor de workflow durante el *proceso*. Una *actividad manual* puede formar parte del *proceso de negocio* y estar incluida en su *definición de proceso* correspondiente, pero no forma parte del workflow automatizado resultado de la ejecución del *proceso* soportada por computador. Normalmente cuando se hable sólo de *actividad*, se hará referencia por defecto a *actividad automatizada*, no *manual*. Algunos sinónimos son Step, Node, **Task**, Work Element, Process Element, Operation, Instruction.

Una **actividad automatizada** o *automated activity* es aquella susceptible de ser computacionalmente automatizada usando un *sistema de gestión de workflow* que gestione la *actividad* durante la ejecución del *proceso* del que forma parte. También se conocen como Workflow Activity o Activity (coloquial).

Una **actividad manual** o *manual activity* es una *actividad* del *proceso de negocio* que no es susceptible de ser automatizada y por lo tanto no forma parte del WMS. Estas actividades pueden estar incluidas en la *definición de proceso* pero no forman parte del workflow resultante. También se conocen como Non-automated Activity, Manual Step, **Human Task**, Manual Work.

Una **instancia** o *instance* es una representación de la ocurrencia de un *proceso* o de una *actividad* de un *proceso*, incluyendo sus datos asociados. Cada *instancia* representa un hilo de ejecución separado del proceso o de la *actividad*, que se puede controlar independientemente y que tendrá sus propios estado interno e identidad visible externa.

La **instancia de proceso** o *process instance* es la ocurrencia de un *proceso*. Representa una ocurrencia individual del *proceso*, con sus propios datos y normalmente capaz de controlar y auditar si progresa hacia el final.

La **instancia de actividad** o *activity instance* es la representación de una *actividad* en una ocurrencia de un *proceso*. Algunos sinónimos son: Step Instance, Node Instance, Task Instance, Work Element Instance.

Un **elemento de trabajo** o *work item* es la representación del trabajo que será procesado por un *participante* en el contexto de una *actividad* de una *instancia de proceso*. Una *actividad* genera uno o varios *elementos de trabajo*, que juntos constituyen la **tarea** que será abordada por el *usuario*. En algunos casos la *actividad* se puede llevar a cabo completamente por una *aplicación invocada*, que puede operar sin un *participante*. Entonces no hay asignación de *elementos de*

trabajo. Los *elementos de trabajo* se suelen asignar al *usuario* vía *lista de trabajo* que mantiene detalles de los *elementos de trabajo* asignados a *usuarios* y un *manejador de la lista de trabajo* que interacciona con la *lista de trabajo* en lugar del *usuario*. El control y el progreso de los *elementos de trabajo* dependen del *manejador de la lista de trabajo* y de los *usuarios*, no tanto del motor de workflow, al que se notifica el estado de los *elementos de trabajo* por medio del interfaz del *manejador de lista de trabajo*. El interfaz WfMC WAPI (WfMC WAPI interface) incluye un API para este propósito. El procesamiento de un *elemento de trabajo* se procesa mediante (1) la *invocación de aplicaciones* o (2) de forma independiente por un *participante*; manteniendo siempre notificado al WMS de la conclusión de un *elemento de trabajo* particular. Algunos sinónimos de *elemento de trabajo* son: Work, Work Object, Work Queue Item, Element, Work Pool Item, **Task**.

Una **lista de trabajo** (*worklist*) es una lista de *elementos de trabajo* asociados con un *participante*. En algunos casos asociados con un grupo de *participantes* que comparten una *lista de trabajo* común. La *lista de trabajo* forma parte del interfaz entre el *manejador de lista de trabajo* y el *motor de workflow*. Normalmente un *manejador de lista de trabajo* pedirá *elementos de trabajo* de un *motor de workflow* para crear esta *lista*. Algunos sinónimos de *lista de trabajo* son: Work Queue, In-Tray, To-Do List.

Un **manejador de lista de trabajo** o *worklist handler* es un software que gestiona la interacción entre el o los *usuarios* y la *lista de trabajo* gestionada por un *motor de workflow*. Esto permite que los *elementos de trabajo* se pasen desde el WMS a los *usuarios* y que se envíen las notificaciones de conclusión (u otras condiciones de estado) entre el *usuario* y el WMS. Algunos sinónimos de *manejador de lista de trabajo* son: WFM Front End, WFM Application, Workflow To-Do List , Application, Task Manager, Active Work Performer.

Acerca de los procesos y su ejecución, la WfMC define una serie de conceptos como:

- El **tiempo de finalización** o *deadline* es una restricción por la que una *actividad* o un *elemento de trabajo* deben ser concluido antes de un cierto tiempo. Puede ser un dato relevante del workflow o un atributo del *proceso de definición*. También se conoce como Completion Time.
- El **encaminamiento paralelo** o *parallel routing* es un segmento de una *instancia de proceso* gobernada por un WMS donde dos o más *instancias de actividad* se ejecutan en paralelo lanzando, en consecuencia, varios hilos de control. Un *enrutamiento paralelo* normalmente empieza con un *AND-Split* y termina con un *AND-Join*. También se conoce como Parallel workflow processing, Concurrent Processing.
- Un **encaminamiento secuencial** o *sequential routing* es un segmento de una *instancia de proceso* gobernada por un WMS donde varias *instancias de actividad* se ejecutan de modo secuencial en un único hilo de control. Durante un *encaminamiento secuencial* no hay *AND-Split* ni *AND-Join*. También se conoce como Serial Routing.
- Una **bifurcación** o *AND-Split* es un punto del workflow en el que el hilo de control se divide en dos o más hilos que se ejecutan en *paralelo*, permitiendo que varias *actividades* se ejecuten simultáneamente. Los hilos de control creados se ejecutan en *paralelo* y de forma independiente hasta que encuentran una *unión*. Sinónimo: Split.
- Una **unión** o *AND-Join* es un punto del workflow en el que varios hilos de control que se ejecutan en paralelo convergen a un único hilo de control. Requiere sincronización. Una *unión* se podría modelar como una *Pre-condición*

de la actividad de *unión*. La unión se puede encontrar con nombres como Join, Rendezvous, Synchronisation join.

- Una **bifurcación condicional** u *OR-Split* es un punto del workflow en el que se ha de tomar una decisión de por dónde seguir el hilo de control ante múltiples alternativas. La transición a la siguiente actividad se hace de acuerdo a la salida de la *condición*. Algunos sinónimos son Conditional Branching, Conditional Routing, Switch, Branch.
- Una **unión asíncrona** u *OR-Join* asíncrona es un punto del workflow en el que varios hilos de control convergen a un único hilo de control. Como no viene de ejecución en *paralelo*, no requiere sincronización. Un hilo de control puede llegar a una *actividad* desde varias alternativas de *actividades* precedentes. La unión asíncrona también se conoce como Join, Asynchronous join.
- Una **iteración** o *iteration* es la ejecución repetida de una o más *actividades* del flujo de trabajo hasta que se cumple una condición. Sinónimos: Workflow Loop, While Loop, Activity Block.
- También contempla los conceptos de **pre-condición** y **post-condición**. Una pre-condición o pre-condition es una expresión lógica que un *motor de workflow* puede evaluar para decidir si la *instancia de un proceso*, o de una *actividad* en una instancia de un *proceso*, puede comenzar. Se pueden definir una o más *pre-condiciones* como criterio de entrada a una *actividad* o *proceso de instancia*. Se pueden testear variables del sistema tales como la fecha o la hora o incluso datos externos. Las *pre-condiciones* se definen en el *proceso de definición*. Una *unión* se puede especificar en forma de *pre-condición*, donde la condición requiere que cada uno de los hilos convergentes haya alcanzado un estado específico. Algunos sinónimos de pre-condición son: Entry criteria, Activity start rules.
- Una **post-condición** o *post-condition* es una expresión lógica que un *motor de workflow* puede evaluar para decidir si la *instancia de un proceso*, o de una *actividad* en una *instancia de un proceso*, se ha completado. Se pueden definir una o más *post-condiciones* como criterio de finalización para una *actividad* o *instancia de proceso* particulares. Las condiciones pueden formar parte de una *iteración*, en la que se ejecutan una o más *actividades* repetidamente hasta que se cumple la *post-condición* definida. Se pueden referir a datos relevantes del workflow o se pueden testear variables del sistema tales como la fecha o la hora o incluso datos externos. Las *post-condiciones* se definen en el *proceso de definición*. Sinónimos: Exit criteria, Activity completion rules.

3.5.7 El Concepto de Tarea en un Estudio Típico de Workflow

Wil van der Aalst define en [Aalst, 2004] una serie de términos en torno al concepto de tarea. En el mayor nivel de abstracción identifica **casos** o *cases*. El objetivo principal del workflow es tratar los **casos**, que podrían equivaler a las *instancias de proceso* según la WfMC. Un *caso* podría ser una reclamación de indemnización, la aplicación de un préstamo hipotecario, la devolución de un impuesto, una orden, un paciente en un hospital, etc. Son casos del mundo real, flujos de trabajo que se pueden modelar. *Casos* similares forman un **tipo de caso** o *case type* (wfmc::definición de proceso). Cada *caso* tiene una **identidad** o *identity* propia que lo distingue de los demás y una vida limitada.

Entre la aparición y desaparición de un *caso* hay un **estado** o *state* particular, *estado* que viene determinado por tres elementos: **atributos** o *case attributes*, **condiciones** o *conditions* y **contenido** o *content*. Normalmente un *sistema de*

workflow no contiene información sobre el contenido del caso. Éste suele estar en forma de documentos, archivos, bases de datos, etc. que no son gestionados por el *WMS*.

Mediante la identificación de las **tareas** o *task* (aproximadamente se correspondería con la *actividad* en la WfMC) se puede estructurar el workflow. Se trata de una unidad lógica indivisible, de manera que si algo va mal durante la ejecución de la *task*, se vuelve al principio de la *task*: **rollback**. En cualquier caso esa atomicidad implícita depende del contexto, ya que para unos puede tratarse de algo indivisible pero para otros no. De todos modos, el concepto de *task* lleva de forma implícita el que sea indivisible. Ejemplos de *tasks* son escribir una carta, rellenar una queja, sellar un documento, chequear datos personales, etc. Existen tareas **manuales**, **automáticas** y **semi-automáticas** (*manual tasks*, *automatic tasks* y *semi-automatic tasks*), según sean realizadas por una persona completamente, o sin su intervención en ningún momento o con la colaboración de ambos.

Es importante destacar que una *tarea* es un trabajo genérico, no la ejecución de una *actividad* para un *caso* específico.

Es decir:

- Una **tarea** (aprox. *wfmc::actividad*) es una unidad lógica indivisible genérica.
- Un **caso** (*wfmc::instancia de proceso*) es un *proceso* concreto del mundo real.
- Un **elemento de trabajo** (*wfmc::elemento de trabajo*) es una combinación de un *caso* y una *tarea* que se va a llevar a cabo.
- Una **actividad** (*wfmc::instancia de actividad*) es la ejecución actual de un *elemento de trabajo*.

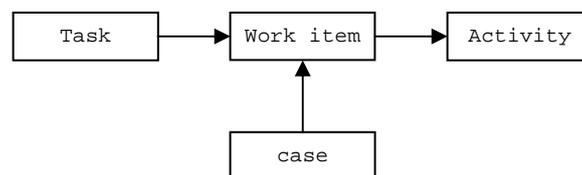


Figura 35. Relación entre *task*, *work item*, *case* y *activity*

El **proceso** o *process* indica las *tareas* que se deben realizar y en qué orden. Es el procedimiento a seguir en un *tipo de caso*. Para decidir este orden se utilizan las **condiciones**. Se puede definir un *proceso* en base a otros *procesos*. Un *proceso* tiene un principio y un final, que marcan lógicamente el principio y el final del *caso*.

También Aalst define la manera en la que se *encaminan* (routing) las *tareas* en los *procesos*, de una manera muy similar a cómo se hacía en la WfMC. Ya vimos anteriormente que desde el principio del *proceso* hasta el final se van eligiendo caminos en las bifurcaciones según unas *condiciones*. Hay cuatro construcciones básicas:

- **Secuencial** o *sequential*, se realizan las *tareas* una detrás de otra. Hay una clara dependencia entre *tareas*: la salida de una de ellas es la entrada de la siguiente.
- **Paralela** o *parallel*, se pueden ejecutar varias *tareas* al mismo tiempo, o en cualquier orden. Se inicializa mediante un *AND-split* (*wfmc::AND-Split*) y se resincroniza con un *AND-join* (*wfmc::AND-Join*).

- **Encaminamiento selectivo** o *selective routing*, cuando hay una selección entre dos o más *tareas*, pero sólo se ejecuta una según qué condición. Se inicializa mediante un *OR-split* y se unen (no sincronizan) con un *OR-join*. Se suelen emplear también los términos *alternative* o *conditional routing*.
- **Iteration**, ejecución de una o varias *tasks* de forma repetitiva.

Por último, comentar que también se dan los disparos o triggers: por iniciativa del *recurso*, por un *evento* externo o por una señal temporal. Si un elemento de trabajo se puede llevar a cabo sin la estimulación externa, entonces no necesita ningún tipo de trigger.

3.5.8 Otras Definiciones en torno a Tareas

En [Molina , 2006] se tienen en cuenta numerosos conceptos relacionados con las tareas a la hora de especificar la interfaz de usuario de los sistemas groupware.

En primer lugar se aclara el uso los conceptos de interacción, inter-acción y acción en la metodología:

- *Interacción* es toda *acción* que se produce entre los usuarios y el sistema durante el uso de una aplicación interactiva para obtener algún tipo de objetivo.
- *Inter-acción* se refiere a las acciones que los usuarios realizan en el contexto de un grupo de trabajo y que son visibles para todos los miembros de dicho grupo. Estas acciones generalmente se realizarán apoyándose en el uso de herramientas, aplicaciones o sistemas.
- *Acción* indica las operaciones de naturaleza atómica en las que se pueden dividir las *tareas* a realizar.

La metodología ofrece una serie de vistas para centrar la especificación del sistema en algunos asuntos concretos. En la *Vista de Inter-Acción* se utilizan los términos *Tarea*, *Tarea de Trabajo en Grupo*, *Tarea Individual*, *Tarea Colaborativa* y *Tarea Cooperativa*:

- Una *tarea* es una actividad realizada por los actores de una organización para alcanzar un cierto objetivo.
- *Tarea de Trabajo en Grupo* es una tarea en cuya realización está envuelto un equipo de trabajo.
- *Tarea Individual* es una tarea desempeñada por un único rol.
- *Tarea Colaborativa* es una tarea de trabajo en grupo en la que intervienen varios roles trabajando de forma coordinada en una tarea común que da un resultado común.
- *Tarea Cooperativa* es una tarea de trabajo en grupo en la que existe una meta común, pero en la que los miembros del equipo trabajan de forma independiente o sobre porciones distintas del espacio de información común.

Por último, en la *Vista de Interacción* aparecen los conceptos de *Tarea Interactiva*, *Tareas de Interacción*, *Tareas de Aplicación* y *Relación entre Tareas*:

- Se define *Tarea Interactiva* como la actividad lógica necesaria para alcanzar un determinado objetivo, y a las que la IU debería dar soporte. Dependiendo de quien realice esta tarea en diálogo persona-ordenador que tiene lugar a través de la IU, se clasifica en tareas de interacción (iniciadas por el usuario) y tareas de aplicación (realizadas por la aplicación).

- Las *Tareas de Interacción* son aquellas tareas llevadas a cabo por el usuario, e iniciadas por él, al interactuar con el sistema.
- Las *Tareas de Aplicación* son aquellas tareas ejecutadas por la aplicación en el contexto del diálogo que se da entre el usuario y la aplicación interactiva.
- *Relación entre Tareas Interactivas*: se trata de relaciones de precedencia y paso de información que se dan entre tareas interactivas.

De todos modos, a pesar de que como en otras muchas aproximaciones se define el concepto de *acción* como las operaciones de naturaleza atómica en las que se pueden dividir las *tareas* a realizar, para no complicar el modelo se prescinde del concepto de acción. En la siguiente figura se observan las dos aproximaciones. La utilizada es la (a):

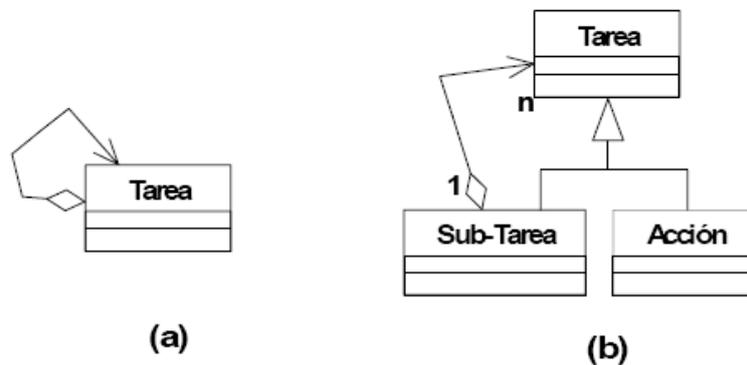


Figura 36. Representación de la relación de descomposición de tareas sin incluir el concepto de acción (a) e incluyéndolo (b) en [Molina, 2006]

3.6 Awareness y Contexto Compartido

La percepción que el usuario tiene de lo que el resto de usuarios hacen en el sistema se conoce con el término anglosajón *awareness*: “*knowing what is going on*” [Endsley, 1995]. Se hará uso indistinto de *awareness* y *percepción*. Gutwin y Greenberg [Gutwin, 2004] resumen las cuatro características básicas del *awareness* recogidas de otros trabajos anteriores:

- *Awareness* es conocimiento sobre el estado de un entorno particular.
- Los entornos cambian con el tiempo, con lo que el *awareness* debe actualizarse constantemente.
- La gente mantiene su *awareness* interactuando con el sistema.
- El *awareness* es un objetivo secundario, esto es, la meta es llevar a cabo ciertas tareas en el sistema, no que el sistema mantenga el *awareness* simplemente.

Según la última característica, el *awareness* se puede ver como un requisito no funcional del sistema.

Se trata de un concepto muy ligado al *contexto compartido*, es decir, el espacio de trabajo común en una tarea de grupo. Formalmente un contexto compartido (*shared context*) es un conjunto de objetos y de acciones que se pueden hacer sobre ellos que es visible por un conjunto de usuarios [Ellis, 1991]. Surgen otros conceptos relacionados como el de *ventana de grupo* (group window), una

colección de ventanas cuyas instancias aparecen en distintos dispositivos para diferentes usuarios, donde el cambio en una de ellas se refleja en el resto de instancias.

En una interacción cara a cara la conciencia de grupo, la percepción de lo que los demás hacen, el conocimiento del contexto compartido es más fácil y natural que en una interacción en la que hay máquinas y espacio de por medio.

En [Gutwin, 2002] se define *workspace awareness* como la noción que se tiene hasta el momento actual sobre la interacción de otra persona con el espacio de trabajo compartido. Sin un buen awareness la facilidad y la naturalidad de las colaboraciones se pierden y las herramientas groupware dejan de ser útiles para llegar a convertirse incluso en un obstáculo.

Se destacan tres como las razones fundamentales por las que se podría perjudicar el workspace awareness en entornos CSCW [Gutwin, 2004]:

- En estos espacios de trabajo sólo se muestra una fracción de la información que se podría tener en una interacción cara a cara.
- La información que genera un usuario en un sistema computacional tras su interacción con él es más pobre que la información que se genera en torno a esa interacción en un espacio de trabajo físico.
- Los sistemas groupware, a veces, ni siquiera muestran esa poca información de que disponen.

Gutwin y Greenberg categorizan los elementos que consideran fundamentales en el workspace awareness, es decir, qué información se genera en él: *quién, qué y dónde*. La Tabla 6 muestra esta categorización y el conjunto de elementos.

Tabla 6. Elementos del workspace awareness

Category	Element	Specific questions
Who	Presence Identity Authorship	Is anyone in the workspace? Who is participating? Who is that? Who is doing that?
What	Action Intention Artifact	What are they doing? What goal is that action part of? What object are they working on?
Where	Location Gaze View Reach	Where are they working? Where are they looking? How much can they see? How far can they reach?

Puesto que los entornos van cambiando con el tiempo, quizás sea necesario mantener un cierto awareness sobre el pasado. En esa línea se proponen los elementos que se muestran en la Tabla 7 [Gutwin, 1997]. En [Tam, 2006] se propone un framework que tiene en cuenta estos elementos para cambios asíncronos en documentos colaborativos y workspaces awareness en general.

Tan importante como qué información se genera en el workspace awareness es el cómo se recoge esa información: (1) por medio de la conversación, gestos y otras *comunicaciones intencionadas*; (2) percepción que obtienen los usuarios al ver los movimientos de los cuerpos del resto de usuarios así como la *comunicación resultante* que surge como consecuencia de la observación; y (3) otros *artefactos del entorno* son también una fuente de información: posición, orientación, movimientos, etc. El mecanismo por el que se asocian ciertas interacciones a una

persona por medio presencia y el sonido de los artefactos se conoce como *feedthrough*.

Tabla 7. Elementos del workspace awareness relacionados con el pasado

Category	Element	Specific questions
How	Action history Artifact history	How did that operation happen? How did this Artifact come to be in this state?
When	Event history	When did that event happen?
Who (past)	Presence history	Who was here, and when?
Where (past)	Location history	Where has a person been?
What (past)	Action history	What has a person been doing?

Para dar soporte al awareness en sistemas groupware distribuidos, en [Gutwin, 2004] se proponen una serie de técnicas computacionales que han demostrado ser de utilidad:

- *Personificación* (embodiments): representan de alguna manera la presencia de una persona en un entorno virtual. Facilitan la comunicación por medio de gestos y la comunicación resultante.
 - *Telepunteros* (telepointers). Indican la presencia de otras personas y lo que están haciendo en ese momento. Suelen incorporar el nombre del usuario. El concepto de telepointer ha aparecido en numerosos trabajos por su utilidad [Ellis, 1991].
 - *Avatares* (avatars). Pequeños iconos que proporcionan algún tipo de información gestual fácilmente interpretable.
 - *Video embodiments*. Mezcla de imágenes de los usuarios con representaciones del espacio de trabajo.
- *Artefactos expresivos*. A menudo, el problema que debilita el awareness de un sistema radica en el modo en el que se realizan las acciones. (1) Puede que las acciones simbólicas (acciones realizadas por medio de botones, barras de herramientas, comandos, combinaciones de teclas) que se realizan no tengan una representación simbólica en el espacio de trabajo común, (2) se pueden realizar de diferentes maneras dificultando así conocer qué o cómo se ha hecho, o incluso (3) pueden ocurrir tan rápidamente que no da tiempo a que el resto de usuarios las interpreten.

Algunos mecanismos como el feedthrough hacen esas acciones más visibles y aumentan el awareness del sistema. Las dos técnicas propuestas son las siguientes:

- *Process feedthrough*. Algunas acciones realizadas a través de widgets como botones o menú (manipulación simbólica) se realizan sin dejar ningún tipo de feedback al resto de usuarios. El conocimiento de algunas de esas acciones podrían ser útiles para el resto de usuarios que podrían tener una idea de las acciones que el resto de usuarios está componiendo. Un ejemplo de cómo proporcionar awareness sería iluminar un botón que pulsa un usuario para el resto de usuarios. Así ellos también sabrían que se ha pulsado ese botón. El feedthrough trata de compensar ese feedback perdido en la manipulación simbólica, proporcionando un feedback explícito para acciones específicas de los usuarios.

- *Action indicators y animations.* Una acción tan rápida como eliminar un objeto de una pantalla puede pasar desapercibida para el resto de usuarios. Una solución en este sentido, sería dejar una pequeña marca temporal en el lugar donde se ha eliminado el objeto en las pantallas remotas. Se prolonga así una acción que en principio es instantánea.
- *Visibilidad.* Las técnicas de personificación y artefactos expresivos vistas con anterioridad facilitan el awareness cuando los usuarios trabajan sobre el mismo espacio de trabajo. Pero cuando una persona está viendo una parte diferente del espacio de trabajo, puede perder información. Proporcionar diferentes vistas soluciona el problema: una vista de radar, vistas reducidas de los otros usuarios, etc. El concepto de *vista* (view) se define en [Ellis, 1991] como una representación visual o multimedia de una porción de un contexto compartido. Vistas diferentes pueden contener la misma información presentada de forma distinta.

Por último se comentan tres aspectos fundamentales a tener en cuenta apuntados en [Ellis, 1991]:

- *Protocolos de comunicación.* De alguna manera ya se ha comentado con anterioridad la necesidad de una comunicación efectiva entre los usuarios de una aplicación groupware. Este punto viene únicamente a constatar ese nivel de importancia.
- *Control de Acceso.* El control de acceso determina quién puede acceder a qué objeto y de qué manera. Es algo necesario que los usuarios del sistema sean conscientes en determinados momentos del control de acceso sobre los objetos. Por ejemplo, si un usuario está manipulando un objeto en una pizarra compartida para desplazarlo hacia la derecha, otro usuario no podría desplazar al mismo tiempo el mismo objeto hacia la izquierda. El sistema debe reflejar que el objeto está bloqueado por el primer usuario, no prohibir sin más el uso del objeto al segundo puesto que daría sensación de mal funcionamiento.
- *Notificaciones.* Se trata de un mecanismo muy efectivo de mantener informados de forma explícita al resto de usuarios del sistema.

3.7 Diseño de Interfaces de Usuario Basado en Modelos

A lo largo de los últimos diecisiete años, el diseño de los sistemas informáticos ha ido evolucionando, no sólo en su aspecto, sino también en el modo en el que se realiza.

Existen numerosos trabajos en torno a la generación más o menos automática, más o menos adaptativa, más o menos multi-plataforma de las aplicaciones.

En una aplicación groupware, como en cualquier otro sistema, el diseño de la interfaz es fundamental, puesto que en ella radica buena parte del éxito del desarrollo. En HCI ha sido un área importante y han surgido técnicas y métodos para facilitar su diseño. En una aplicación groupware el usuario tendrá su propia interfaz y estos métodos son aplicables para su diseño. Sin embargo, los sistemas CSCW son sistemas con algunas particularidades, como se ha mostrado en capítulos anteriores, que merecerían la pena cuidar especialmente para conseguir un diseño de interfaz de mayor calidad y más cercana a las necesidades del usuario.

En este apartado se enumeran algunos de esos trabajos que han contribuido en el avance del diseño de interfaces de usuario. Los más significativos y que más han influenciado el desarrollo de esta tesis se describen con mayor detalle.

3.7.1 Enumeración de Trabajos para el Diseño de Interfaces de Usuario

Algunos de los trabajos más relevantes en el diseño de sistemas son los que se describen brevemente a continuación. La gran mayoría de ellos se centran en el diseño de sistemas en los que la interacción con el usuario toma un papel fundamental. Sólo AMENITIES y CIAM tienen en cuenta las colaboraciones explícitas que se puedan dar a través del sistema.

La mayoría de las aproximaciones se basan en conceptos como CIO (Concrete Interaction Object), AIO (Abstract Interaction Object), Window o PU (Presentation Unit). Estos conceptos se comentan con mayor nivel de detalle en el siguiente apartado (3.7.2), donde se describen aquellos trabajos que más han influenciado la etapa de diseño (ver apartado 5.4) de la metodología que se presenta en esta tesis.

Las aproximaciones enumeradas son las siguientes:

- TRIDENT (Tools foR an Interactive Development ENvironment) [Bodart, 1990]
Es un entorno metodológico para el desarrollo de aplicaciones interactivas que utiliza modelos declarativos: Modelo de Tareas y Modelo de Aplicación, representados por un Grafo de Cadenas de Actividad (ACG) y por un diagrama entidad –relación respectivamente.
Por medio de esta aproximación se resuelven la selección de los elementos de presentación, la elección de objetos de interacción, su distribución por la pantalla, la secuenciación de diálogos y conversación básica.
- UIDE (User Interface Design Environment) [Foley, 1991]
UIDE es un sistema que permite la generación automática de la interfaz de usuario a partir de un Modelo de Aplicación que describe cómo se ha de comportar la interfaz durante su ejecución. Una extensión del sistema [Sukaviriya, 1995] incluye un Modelo de Interfaz que especifica sus componentes, tareas y restricciones.
- JANUS [Balzert, 1996]
JANUS permite la generación automática de la interfaz de usuario en base a un Modelo de Dominio extendido
- MASTERMIND [Szekely, 1996]
Tres son los modelos en los que se basa MASTERMIND para diseñar las interfaces de usuario: Modelo de Presentación, de Diálogo y de Interacción. La interfaz se puede generar de forma automática, como en muchos de las aproximaciones presentadas.
- MOBI-D (Model-Based Interface Designer) [Puerta, 1996] [Puerta, 1997]
Se trata de un sistema que soporta el diseño de la interfaz de usuario por medio de modelos de Tareas, Dominio, Usuario, Diálogo y Presentación. El lenguaje de modelado asociado, MIMIC, proporciona además la posibilidad de realizar un Modelo de Diseño.
Una de las principales ventajas de MOBI-D se produce en su Modelo de Diseño, donde se describen las relaciones dinámicas entre las entidades del resto de modelos.

- OVID (Object, View and Interaction Design) [Roberts, 1998]

OVID proporciona una especificación completa de la interfaz de usuario de una aplicación. Se basa en las metodologías orientadas a objetos, tratando de compensar una carencia común a todas ellas, precisamente la carencia de técnicas específicamente orientadas a la generación de la interfaz.

Se basa en varios modelos: Modelo Conceptual de Usuarios (según la experiencia del usuario y su modelo mental), Modelo del diseñador (que describe las componentes de la interfaz y sus relaciones a partir de la información obtenida en el modelo anterior), Modelo del programador (donde se describen los detalles de implementación necesarios).

- TADEUS [Stary, 1999; Stoiber, 2000]

TADEUS proporciona (1) un framework basado en modelos para la representación de interfaces de usuario, (2) una metodología y (3) el entorno correspondiente para el desarrollo de dichos interfaces.

TADEUS utiliza diferentes modelos para comprender en profundidad a los usuarios finales y la organización del trabajo durante el desarrollo: modelo de tareas, modelo de usuario, modelo de la información, modelo de interacción. Cada modelo tiene una especificación estructural y una dinámica.

- El *modelo de tareas* se deriva de modelos de inteligencia empresarial. Las tareas relevantes para los procesos de negocio interactivos pueden dividirse en otras subtareas (acciones y operaciones), o bien pueden provenir de otras super-tareas. La notación utilizada para el modelado es ORD y la especificación dinámica se realiza por medio de OBDs.
- El *modelo de usuario* captura los roles que se asignan a tareas y/u operaciones que se deben realizar por medio del sistema interactivo. También captura características individuales de usuarios relacionadas con interacción.
- El *modelo de la información* representa el dominio del problema.
- El *modelo de interacción* se basa en una arquitectura de interfaz de usuario genérica como las plataformas MFC o JFC. Se especifican todos los controles (GUI) necesarios para representar las partes estructurales y dinámicas de las tareas.

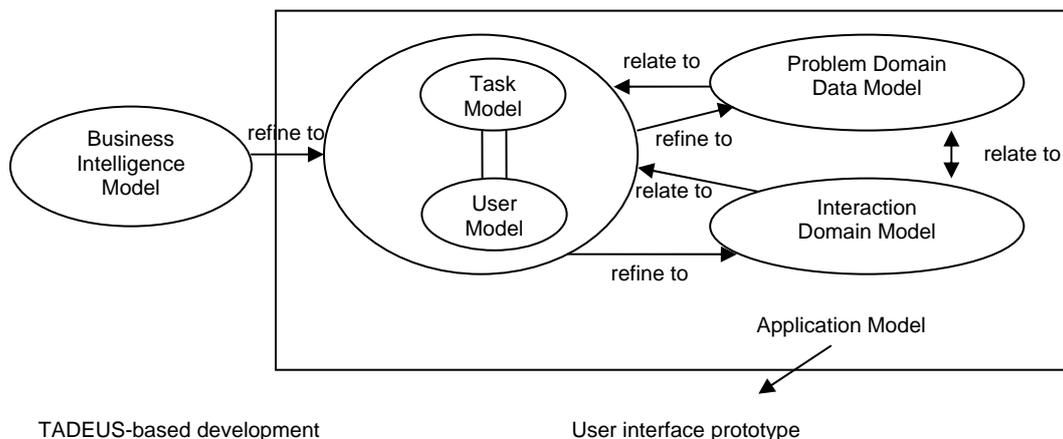


Figura 37. Marco de referencia de TADEUS

- Wisdom (Whitewater Interactive System Development with Object Models) [Nunes, 2001]

Esta aproximación basada en UML proporciona modelos de prototipos evolutivos. WISDOM da soporte al desarrollo centrado en el usuario y al diseño de la interfaz de usuario.

Proporciona mecanismos para abordar el modelado de los roles de usuario, modelado de la interacción, diálogo y presentación. Introducen nuevos elementos para la interfaz de usuario desde el análisis.

Así mismo, propone hacer uso de un subconjunto determinado de la notación de UML para reducir el número de conceptos que se requieren para el desarrollo de sistemas interactivos.

- IDEAS (Interface Development Environment within OASIS) [Lozano, 2001]

IDEAS es una metodología completa para el desarrollo de interfaces de usuario interactivas que en su fase de diseño propone un Modelo de Diálogo que describe la interfaz gráfica de la GUI. Este modelo se aborda a partir de cuatro diagramas: Diagrama de Interacción de Diálogos, Diagrama de Estados Internos, Diagrama de Especificación de Componentes y Tabla de Definición de Componentes.

Propone el uso de unos contratos representados por diagramas con clases y agregaciones de UML. El contrato pone en relación los AIOs con los objetos del dominio.

- UMLi (The Unified Modeling Language for Interactive Applications) [Silva, 2000] [Silva, 2002]

En UMLi se propone el uso del tradicional UML para el modelado de sistemas con unas extensiones que permitan especificar con mayor completitud las interacciones de los usuarios.

En este sentido utiliza un diagrama para la interfaz de usuario con seis constructores a modo de AIO que especifican el rol de cada objeto de interacción en la interfaz de usuario.

UMLi se describirá posteriormente con mayor nivel de detalle por tratarse de un trabajo reciente que ha tenido repercusión y que ha influenciado esta tesis en el diseño de sistemas CSCW.

- Just-UI [Molina, 2003]

Posibilita la generación automática de interfaces de usuario de aplicaciones de gestión para diferentes plataformas a partir de un modelado conceptual: patrones conceptuales para la captura de requisitos a partir de una extensión de OO-METHOD [Pastor, 1997].

- OO-H (Object Oriented Hypermedia Method) [Cachero, 2003]

OO-H es un método para la generación de interfaces de usuario Web personalizadas a partir del Modelo de Usuario que define.

- TERESA (Transformation Environment for inteRactivE Systems representAtions) [Mori, 2004]

TERESA es una herramienta basada en la notación CTT [Paternò, 1999] para el modelado de tareas que permite, a partir de las mismas, generar la interfaz de usuario en diferentes plataformas (portátiles, ordenadores de sobremesa, pda, etc.). La notación CTT ha sido explicada con detalle por el uso relevante que se hace de ella en esta tesis.

- UsiXML (USer Interface eXtensible Markup Language) [Limbourg, 2004]

UsiXML es un lenguaje de marcado basado en XML que describe la interfaz de usuario desde varias perspectivas: Character User Interfaces (CUIs), Graphical User Interfaces (GUIs), Auditory User Interfaces y Multimodal User Interfaces.

Proporciona un medio para la descripción de aplicaciones interactivas que preserve un diseño independiente de características como la plataforma final o el entorno.

Describe mediante un nivel de abstracción alto los elementos constitutivos de la interfaz de usuario de una aplicación: widgets, controles, etc.

Así mismo, está basado en el proyecto europeo Cameleon [Calvary, 2003] que propone cuatro niveles de abstracción como marco de desarrollo de interfaces de usuario: Tareas & Conceptos, UI abstracta (AUI), UI Concreta (CUI) y UI Final (FUI).

- AB-UIDE (Agent-Based User Interface Development Environment) [López, 2004] [López, 2005]

AB-UIDE es un método que permite la generación de interfaces de usuario capaces de adaptarse a las distintas situaciones que potencialmente pueden surgir durante el proceso de interacción. Los agentes en los que se basa colaboran para proporcionar inteligentemente al usuario las adaptaciones más adecuadas en cada situación que se presenta durante la interacción con la interfaz de usuario diseñada.

- CIAM (Collaborative Interactive Applications Methodology) [Molina, 2006]

Junto con la anteriormente comentada AMENITIES es una de las pocas aproximaciones relacionadas con entornos CSCW para el desarrollo de la UI. Se trata de un marco metodológico para el diseño de la Interfaz de Usuario en aplicaciones colaborativas.

No se trata de una metodología completa sino que se propone como extensión o inclusión en otras metodologías existentes para completar las carencias que tienen en el desarrollo de aplicaciones CSCW. Por lo tanto, no aborda el diseño de manera directa, sino que facilita que en el desarrollo de las aplicaciones groupware se tengan en cuenta aspectos groupware desde las tareas.

Puesto que es un trabajo relacionado con sistemas CSCW, ha influenciado el desarrollo de esta tesis. Por este motivo, en el apartado 3.9.2 se describe con mayor detalle y aparecen varias referencias durante este trabajo.

- FlowiXML [Guerrero, 2007]

FlowiXML es una metodología que está siendo desarrollada en la actualidad por Josefina Guerrero García bajo la supervisión del Dr. Jean Vanderdonckt en la Universidad de Louvain, Bélgica.

Esta metodología está orientada al desarrollo de interfaces de sistemas workflow que estén basados explícitamente los requisitos y procesos de una organización de un modo co-evolutivo. Está basada en la definición de un modelo de proceso y un modelo de tareas que consideran los componentes de la organización que están involucrados.

3.7.2 De Objetos de Interacción Abstractos a Facetas

Las últimas tendencias en el desarrollo de sistemas abordan el diseño de los sistemas desde una perspectiva genérica. Esto se debe al gran número de posibilidades que ofrece el mercado y las necesidades de los usuarios.

Es decir, la idea subyacente para el desarrollo de una aplicación es la que es, sin embargo, según el contexto de ejecución, la plataforma objetivo (pda, portátil, teléfono móvil, etc), etc. el modo en el que la interfaz de usuario se presente puede variar considerablemente.

Por lo tanto, el objeto del diseño en la actualidad pasa por conseguir que los modelos generados en el análisis y en el propio diseño sean genéricos y se especifiquen detalles de implementación posteriormente según las necesidades de plataforma, o características de personalización, etc.

3.7.2.1 Primera Aproximación al Diseño Genérico de Interfaces de Usuario

Este es el fundamento de los Objetos de Interacción Abstractos, del inglés Abstract Interaction Objects (AIOs). La primera aproximación, en 1990, al diseño de interfaces de usuario (UI) teniendo en cuenta esta peculiaridad fue TRIDENT [Bodart, 1990; Bodart, 1995]. TRIDENT es un conjunto de herramientas interactivas que genera automáticamente la UI en aplicaciones altamente interactivas.

Para ello propone el uso de los AIOs en el diseño de la UI, que proporciona una selección de objetos independientes de detalles de implementación. Por medio de reglas automáticas se seleccionan los AIOs a partir de los modelos de datos de aplicación y diálogo. Finalmente, una técnica basada en árboles de decisión hace este razonamiento obvio para el usuario.

En general, cuando se selecciona la plataforma deseada, las características particulares de implementación, los AIOs se traducen a Objetos de Interacción Concretos (Concrete Interaction Objects, CIOs) en la fase de implementación. Estos CIOs específicos también se conocen como *widgets* (windows gadgets) o *controls or physical interactors* (según la terminología IFIP).

Los AIOs (logical interactors según la terminología IFIP) abstraen características de conjuntos de CIOs independientes del entorno. Vanderdonckt y Bodart propusieron una clasificación de hasta seis tipos de AIO. Cada elemento de cada tipo se identifica con un nombre genérico (p.e. check box), tiene atributos generales y particulares (p.e. height, width, color, states), eventos abstractos (p.e. value selection, mouse click) y primitivas.

Tabla 8. Tipos de Objeto de Interacción Abstracto y ejemplos de elementos

Tipo AIO	Elementos AIO (algunos)
Action Objects	Menu, menu item, menu bar, drop-down menu, cascade menu, submenu
Scrolling Objects	scroll arrow, scroll cursor, scroll bar, frame
Static Objects	label, separator, group box, prompt, icon
Control Objects	edit box, scale, dial, check box, switch, radio box, spin button, push button, list box, dropdown list box, combination box, table
Dialog Objects	window, help window, dialog box, expandable dialog box, radio dialog box, panel
Feedback Objects	message, progression indicator, contextual cursor

Cada AIO se puede corresponder con 0, 1 o varios CIOs en cada entorno y además pueden ser simples y compuestos. Los tipos de AIO simples se pueden ver en la Tabla 8, mientras que los compuestos lo están por otros AIOs simples o compuestos.

3.7.2.2 Extensión de la Idea de Objeto de Interacción Abstracto en el Diseño de Sistemas Interactivos

La idea del AIO se extiende a otros métodos, técnicas y metodologías para el diseño de interfaces de usuario para herramientas interactivas. La mayoría de las aproximaciones comentadas en el apartado anterior adoptan este enfoque.

En 2001, la metodología IDEAS [Lozano, 2001] propone un Modelo de Diálogo para la descripción gráfica de la GUI que lleve a la implementación final de la UI. Este Modelo de Diálogo es una descripción abstracta de las acciones, y sus posibles relaciones temporales, que los usuarios y los sistemas pueden llevar a cabo a nivel de UI durante una sesión interactiva.

Concretamente establece cuándo el usuario puede invocar comandos, seleccionar o especificar datos de entrada y cuándo el sistema puede requerir datos del usuario o mostrar los datos de salida. Para todo ello se hace uso de los AIOs.

El Modelo de Diálogo se aborda desde cuatro diagramas: Diagrama de Interacción de Diálogos, Diagrama de Estados Internos, Diagrama de Especificación de Componentes y Tabla de Definición de Componentes.

El primero de ellos, el *Diagrama de Interacción de Diálogos*, modela el comportamiento de la GUI en base a las ventanas y diálogos (se emplea el término diálogo con carácter general normalmente) que el usuario necesita para llevar a cabo sus tareas y, lógicamente, las selecciones que el usuario hace para pasar de un diálogo a otro. La Figura 38 muestra un ejemplo de Diagrama de Interacción de Diálogos en IDEAS, donde se aprecian cada uno de los diálogos (en forma de cuadros) así como las acciones que tienen que desempeñar los usuarios para llegar de un sitio a otro. La notación se basa en los diagramas de estados –conocidos como statechart– de [Harel, 1987].

IDEAS propone además modelar el comportamiento intra-estados, es decir, lo que sucede en cada diálogo en particular. Para ello se utilizan los diagramas de estados de UML, dando lugar a lo que se conoce como *Diagrama de Estados Internos*, ejemplo del cual se puede apreciar en la Figura 39.

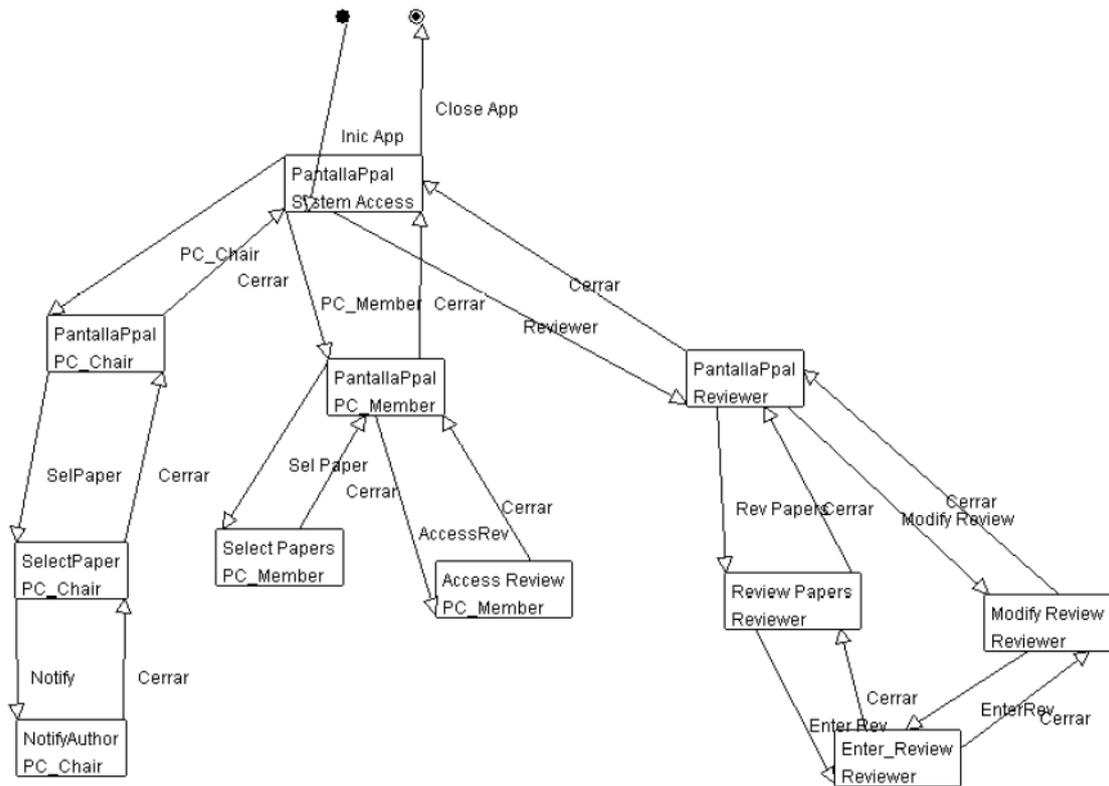


Figura 38. Ejemplo de Diagrama de Interacción de Diálogos en IDEAS

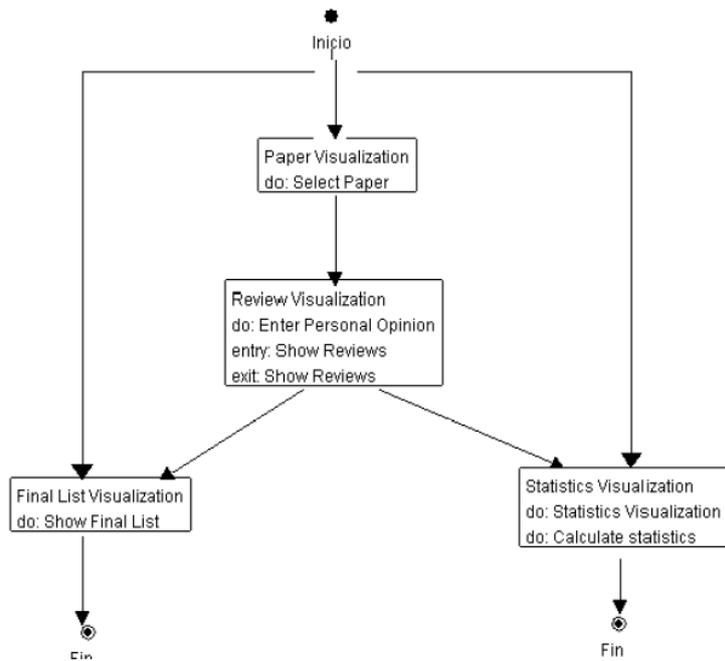


Figura 39. Ejemplo de Diagrama de Interacción de Diálogos en IDEAS

Es por medio del *Diagrama de Especificación de Componentes* como se representa la estructura de la GUI en base a los AIOs. Es decir, modela el conjunto de AIOs que el usuario necesita en cada diálogo. IDEAS simplifica el uso de estos objetos abstractos identificando únicamente tres tipos:

- Componentes

Este AIO tiene el objetivo de agrupar semánticamente otros AIOs formando unidades cognitivas para el usuario que posteriormente se concretizarán en forma de bordes, colores, agrupaciones, etc.

- Herramientas de Visualización

Son objetos abstractos para que el usuario pueda mostrar o introducir información de o al sistema.

- Herramientas de Control

Estos AIOs permiten que el usuario pueda controlar el funcionamiento de la aplicación: aceptar, cancelar, navegar, etc.

La Figura 40 muestra un ejemplo de cómo se representaría en IDEAS un diálogo en base a los AIOs que contenga para que el usuario pueda realizar las tareas que precise.

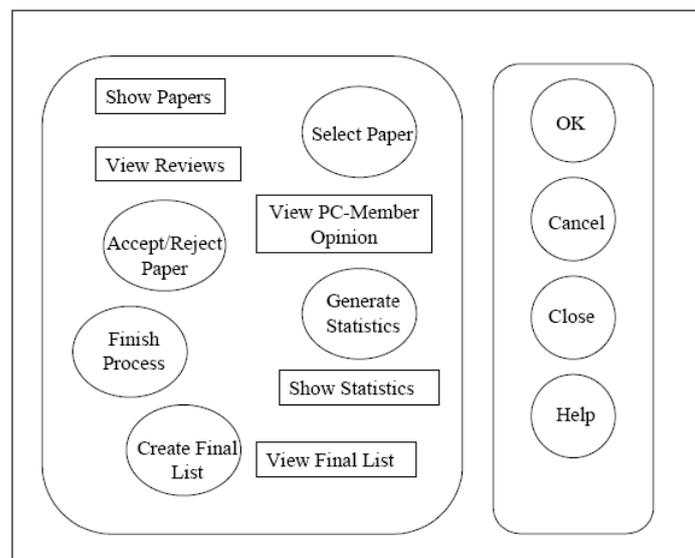


Figura 40. Ejemplo de Diagrama de Especificación de Componentes en IDEAS

Otra novedosa aportación de la metodología para el diseño de las UI de sistemas interactivos es el uso de *contratos* que ponen en relación los objetos del dominio con los AIOs que los manejan. Esta relación se efectúa por medio de agregaciones de clases y AIOs en un mismo diagrama mostrando así esa asociación. La Figura 41 muestra un ejemplo de ello.

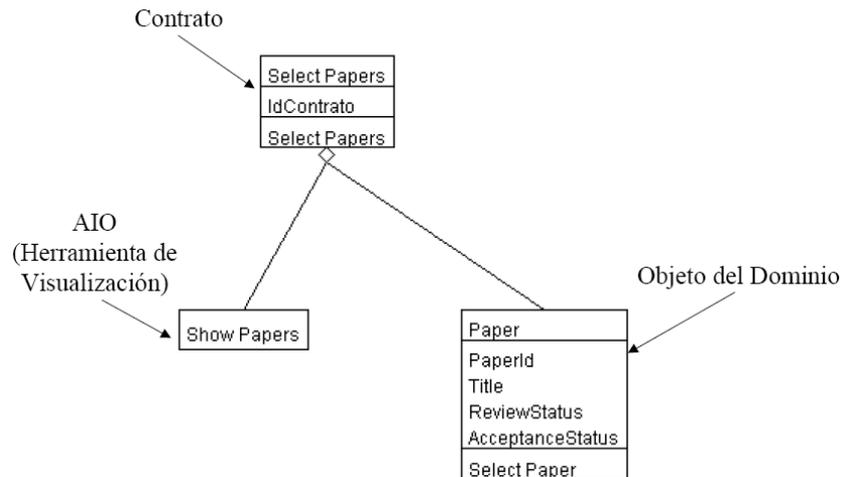


Figura 41. Ejemplo de *Contrato* en IDEAS para poner en relación AIOs con los objetos del dominio

3.7.2.3 Agrupaciones de Objetos de Interacción Abstractos

Las diferentes aproximaciones incorporan conceptos similares en torno a los AIOs. Uno de ellos, la idea de agrupar AIOs dentro de un mismo diálogo (según el modelo conceptual de IDEAS) se mantiene por la naturalidad que supone su uso para presentar la información relacionada de forma más sencilla para el usuario. Es el caso del tipo *Componente* descrito con anterioridad en IDEAS.

Entre 2000 y 2002 surge UMLi [Silva, 2000] como una nueva notación que extiende mínimamente UML para poder llevar a cabo el diseño integrado de aplicaciones y sus interfaces de usuario. Basada en modelos de interfaces de usuario declarativos, es un nexo entre desarrolladores de UI y desarrolladores de aplicaciones, es decir, se describe al mismo tiempo la interfaz y la aplicación.

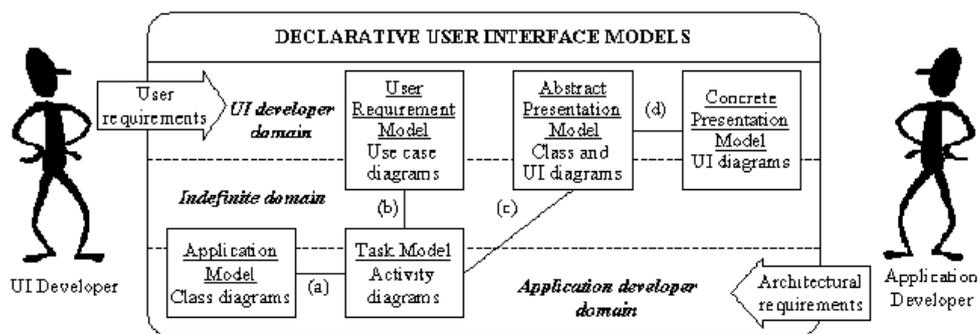


Figura 42. Modelos de interfaz de usuario declarativos en UMLi

La Figura 42 muestra el uso de los distintos modelos y diagramas desde ambos dominios (desarrollador de aplicaciones y de la UI). Destacan el Modelo de Presentación Abstracta y el Modelo de Presentación Concreta donde se utilizan los AIOs y los CIOs correspondientes para el diseño de la UI.

El Diagrama de Interfaz de Usuario que proponen hace uso seis AIOs o constructores que especifican el rol de cada objeto de interacción en una presentación de UI. Los dos primeros (*FreeContainers* y *Containers*) son AIOs de agrupación, lo que confirma la tendencia por este tipo de AIOs:

- *FreeContainers* . Son objetos de interacción de alto nivel que no pueden estar contenidos en otros objetos de interacción (p.e. ventanas). También se les conoce como *unidades de presentación* puesto que los objetos de interacción que contenga se presentan todos al mismo tiempo.
- *Containers* . Estos AIOs agrupan otros AIOs que no sean FreeContainers.
- *Inputters* . AIOs responsables de recibir información de los usuarios.
- *Displayers* . AIOs responsables de enviar información visual a los usuarios.
- *Editors* . Son AIOs Inputters y Displayers al mismo tiempo.
- *ActionInvokers* . AIOs responsables de recibir información de los usuarios en forma de eventos.

El modelo de presentación para un diálogo determinado usando UMLi se puede refinar [Silva, 2002]. Es decir, algunos de los objetos de interacción se pueden descomponer en otros que muestren una descripción más precisa de la información. Así, un diagrama de UI que se haya modelado como el de la Figura 43 (a) podría refinarse a un diagrama de UI como en del la Figura 43 (b).

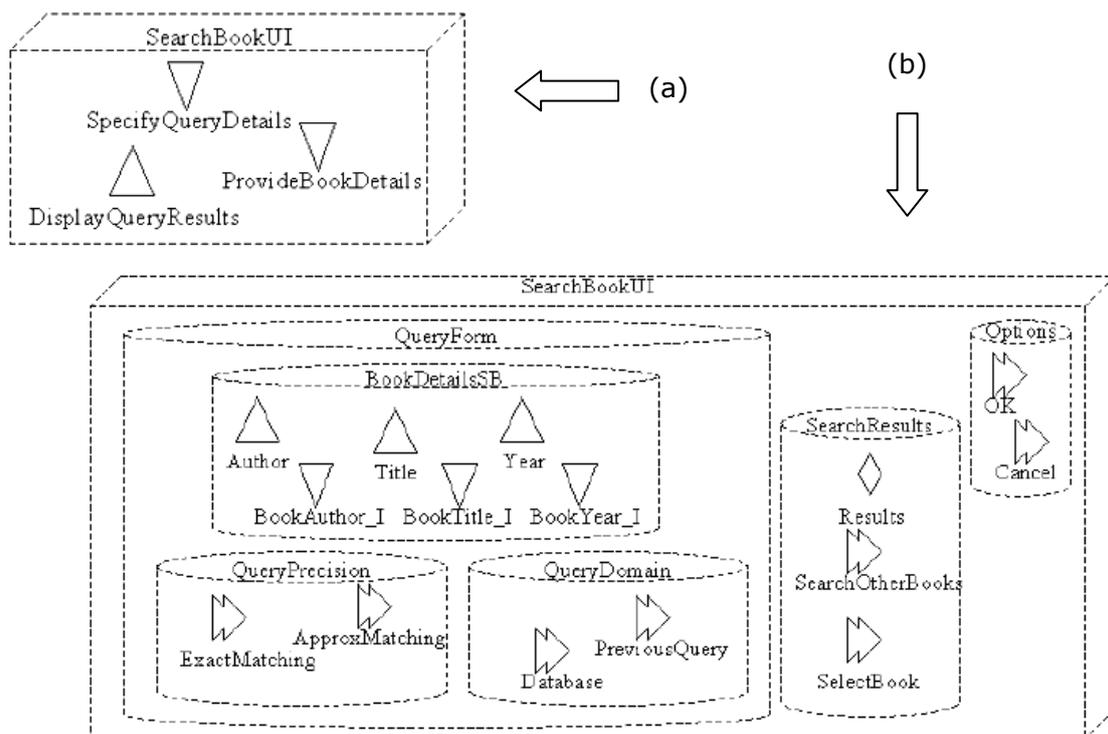


Figura 43. (a) Ejemplo de diagrama de interfaz de usuario en UMLi en su versión preliminar. (b) Ejemplo del mismo diagrama refinado

3.7.2.4 Facetas para Describir los Objetos de Interacción

En 2004, una última evolución de los AIOs propone el uso de facetas para su descripción. En el marco de la tesis de Quentin Limbourg [Limbourg, 2005] se desarrolla un lenguaje para el desarrollo de UI independientes de la plataforma final y de otras características de implementación: UsiXML [Limbourg, 2004].

El Modelo de Interfaz de Usuario Abstracto (AUI) representa esta realidad en base a los AIOs y las *relaciones de interfaz de usuario abstractas*. Estos conceptos constituyen un vocabulario independiente de la implementación. Concretamente, los AIOs son independientes de la *modalidad* y de la *plataforma*.

La *modalidad* indica el modo de interacción del usuario: interacción gráfica, vocal, reconocimiento y síntesis del habla, interacción basada en video, virtual, realidad mixta o aumentada. Desde [Nigay, 1995], modalidad se puede definir de una manera más precisa como la asociación de un dispositivo físico d con un lenguaje de interacción L : $\langle d, L \rangle$. UsiXML soporta dos modalidades: habla (entrada y salida) y gráfica (entrada y salida):

- *Speech input*: $\langle \text{microphone}, \text{pseudo-natural language NL} \rangle$, donde NL se define por medio de una gramática específica.
- *Speech output*: $\langle \text{speech synthesizer}, \text{pseudo natural language NL} \rangle$, donde NL se define por medio de una gramática específica.
- *Graphic input*: $\langle \text{pointing device PD}, \text{direct manipulation} \rangle$, donde PD normalmente es un ratón.
- *Graphic output*: $\langle \text{screen}, \text{drawing language} \rangle$, donde un *drawing language* puede ser, por ejemplo, procedural o declarativo, pixel based o vector based.

La independencia de la *plataforma* es tanto software como hardware, es decir, entendida como sistema operativo y dispositivo.

Una novedad presentada en [Limbourg, 2005] en relación a los AIOs es su división en sólo dos tipos: aquellos que no contienen ningún otro, *Abstract Individual Components (AIC)*; y los que agrupan otros objetos de interacción, los *Abstract Containers (AC)*.

Un AIC es una abstracción que permite la descripción de objetos de interacción independientemente de la modalidad en la que se implementará.

La siguiente gran novedad es el uso de *facet*s (*facets*) para describir cada AIC. Cada AIC puede estar compuesto por múltiples *facet*s. Cada una de ellas describe una función particular que el AIC correspondiente puede llevar a cabo en el mundo físico. En concreto se identifican cuatro tipos de *facet*a:

- *Input* describe la acción de entrada soportada por un AIC.
- *Output* describe qué datos se pueden presentar al usuario por medio del AIC.
- *Navigation* describe una posible transición que un AIC puede habilitar.
- *Control* describe enlaces entre un AIC y funciones del sistema, como por ejemplo, métodos del modelo de dominio.

Un AC es una entidad que permite la agrupación lógica de otros ACs u otros AICs. Los ACs dan soporte a la ejecución de conjuntos de tareas conectadas lógicamente o semánticamente. Este concepto se conoce en otros trabajos como *unidades de presentación (presentation units, PE)* o *espacios de trabajo (work spaces, WE)*.

En el proceso de reificación al nivel concreto un AIC podría asumir diferentes funcionalidades puesto que puede asumir diferentes *facet*s. Un AC en ese proceso de reificación se correspondería con contenedores gráficos como ventanas, cuadros de diálogo, layouts, etc.

La Figura 44 muestra un detalle de la relación de los conceptos empleados en esta ontología.

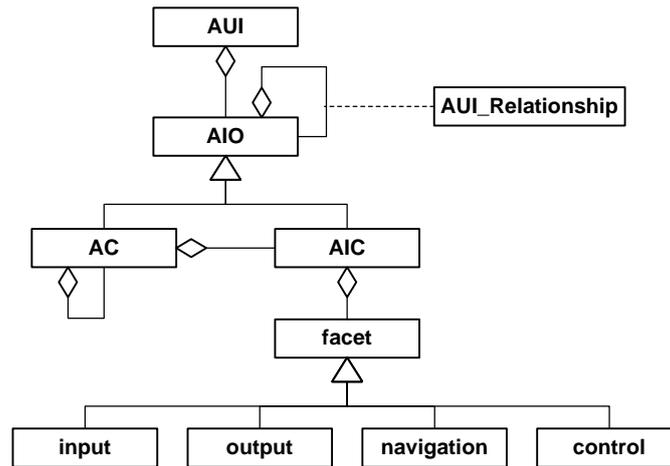


Figura 44. Detalle de la relación de los conceptos empleados en UsiXML en el modelado de la UI abstracta

Así mismo, es importante destacar que la metodología está basada en el Cameleon Reference Framework [Calvary, 2003] que define los pasos para el desarrollo de UI para aplicaciones interactivas multi-contexto. Los cuatro pasos que identifica son los siguientes (también se pueden apreciar en la Figura 45):

- *Final UI (FUI)*: es la UI operacional, la implementación en un contexto determinado con detalles de plataforma, lenguaje, etc.
- *Concrete UI (CUI)*: concretiza un AUI para un contexto de uso determinado a través de los correspondientes CIOs. Es independiente de la plataforma y se considera una reificación de un AUI a nivel superior y una abstracción del FUI con respecto a la plataforma.
- *Abstract UI (AUI)*: define los espacios de interacción (o unidades de presentación) agrupando tareas de acuerdo a varios criterios: AIOs.
- *Task & Concepts (T&C)*: describe las tareas que se han de llevar a cabo los conceptos del dominio requeridos por esas tareas para ser llevadas a cabo.

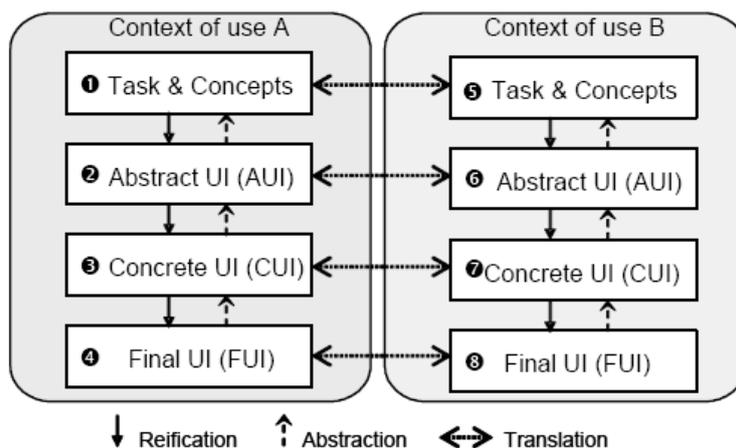


Figura 45. Marco de referencia Cameleon para el desarrollo de aplicaciones interactivas multi-contexto. Imagen obtenida de [Limbourg, 2004]

Como se puede apreciar, este marco también proporciona tres tipos de transformación entre los diferentes niveles. El proceso de *abstracción/reificación* (*abstraction/reification*) que consiste en abstraer o concretizar, respectivamente,

verticalmente entre los niveles definidos, mientras que el proceso de *traducción* (*translation*) supone un cambio en los artefactos utilizados según el contexto de uso.

El lenguaje de especificación de modelos de presentación propuesto en UsiXML no tiene una representación gráfica, sin embargo, en la tesis de Montero [Montero, 2005] sobre calidad y experiencia en el desarrollo de interfaces de usuario se asocia una notación gráfica para cada elemento. Esta notación se ha seguido empleado en trabajos posteriores como se puede ver en [Vanderdonckt, 2005]. La Figura 46 muestra los símbolos empleados para representar cada objeto de UsiXML.

Icono	Descripción
	Icono asociado a un objeto Container. Objeto de interacción abstracto que permite aglutinar componentes en su interior.
	Icono asociado a un Componente. Objeto de interacción abstracto que mediante la incorporación de distintas facetas definidas en usiXML permite dar soporte a la interacción mantenida entre usuario y sistema.
	Iconos asociados a las diferentes facetas definidas en usiXML. A través de ellas es posible dotar a los componentes de interacción de facilidades para dar soporte a diferentes acciones de interacción, respectivamente, entrada de datos, salida de datos, operaciones de control y de navegación.

Figura 46. Notación gráfica de IdealXML sobre objetos de UsiXML

3.8 Implementación de Interfaces de Usuario

Las etapas de implementación de UIs en HCI de los últimos tiempos están muy orientadas a concretizar modelos abstractos de etapas anteriores en base a detalles de implementación, plataforma, etc. Son un paso consecuente del diseño de sistemas y su historia reciente viene muy ligada a la de esta etapa de desarrollo de AUIs (Abstract User Interfaces). La mayoría de los trabajos comentados en el apartado dedicado al diseño de UIs (apartado 3.7) culminan esa etapa con una de implementación que concretiza los elementos que formarán parte de la interfaz de usuario final (proceso de reificación).

Otro ejemplo es Multimodal Teresa [Mori, 2004; Paternò, 2006], un entorno para el diseño de interfaces multimodal. Maneja diferentes niveles de abstracción partiendo del modelo de tareas de CTT para crear las UIs mediante una serie de transformaciones semi-automáticas. Interfaces que se generarán para la o las plataformas específicas a partir del mismo modelo inicial.

La Figura 47 muestra un esquema del funcionamiento de TERESA. En primer lugar se construye un modelo de tareas de alto nivel para aplicaciones independientemente de la plataforma. El primer paso del método consiste en la elaboración de un *modelo de tareas del sistema* para las diferentes plataformas consideradas (telefonía, escritorio, voz, etc.). Entonces se genera una interfaz de usuario abstracta, *AbstractUI*, según la plataforma. En el siguiente paso, según propiedades específicas de cada plataforma, se generan las interfaces de usuario concretas: *ConcretUI*. El cuarto y último paso consiste en la generación del código a partir de la descripción de la interfaz concreta.

Por los cuatro pasos del método seguido puede apreciarse que Multimodal Teresa también se basa en el Cameleon Reference Framework [Calvary, 2003] que define los pasos para el desarrollo de UI para aplicaciones interactivas multi-contexto.

Muchos...

- ... lenguajes empleados en el diseño de interfaces (UIDLs, User Interface Description Languages)...
- XIML [Eisenstein, 2001], SunML [Dery-Pinna, 2003], UIML [Ali, 2004], UsiXML [Limbourg, 2004])
- ... entornos y sistemas basados en modelos ...
- TRIDENT [Bodart, 1990], UIDE [Foley, 1991], JANUS [Balzert, 1996], MASTERMIND [Szekely, 1996], MOBI-D [Puerta, 1996] [Puerta, 1997], OVID [Roberts, 1998], TADEUS [Stary, 1999; Stoiber, 2000], Wisdom [Nunes, 2001], UMLi [Silva, 2000] [Silva, 2002], Just-UI [Molina, 2003], TERESA [Mori, 2004], etc.
- ... y métodos y metodologías...
- IDEAS [Lozano, 2001], OO-H [Cachero, 2003], AB-UIDE [López, 2004] [López, 2005], [Montero, 2005], etc.

... parten de esta idea general para construir UIs para plataformas específicas; y los trabajos más recientes en esta línea continúan haciendo uso del marco de referencia Cameleon o aproximaciones similares para generar las interfaces [Paternò, 2006; Lepreux, 2006; Coyette, 2005; Stanculescu, 2005; Montero, 2005; López, 2005; etc.].

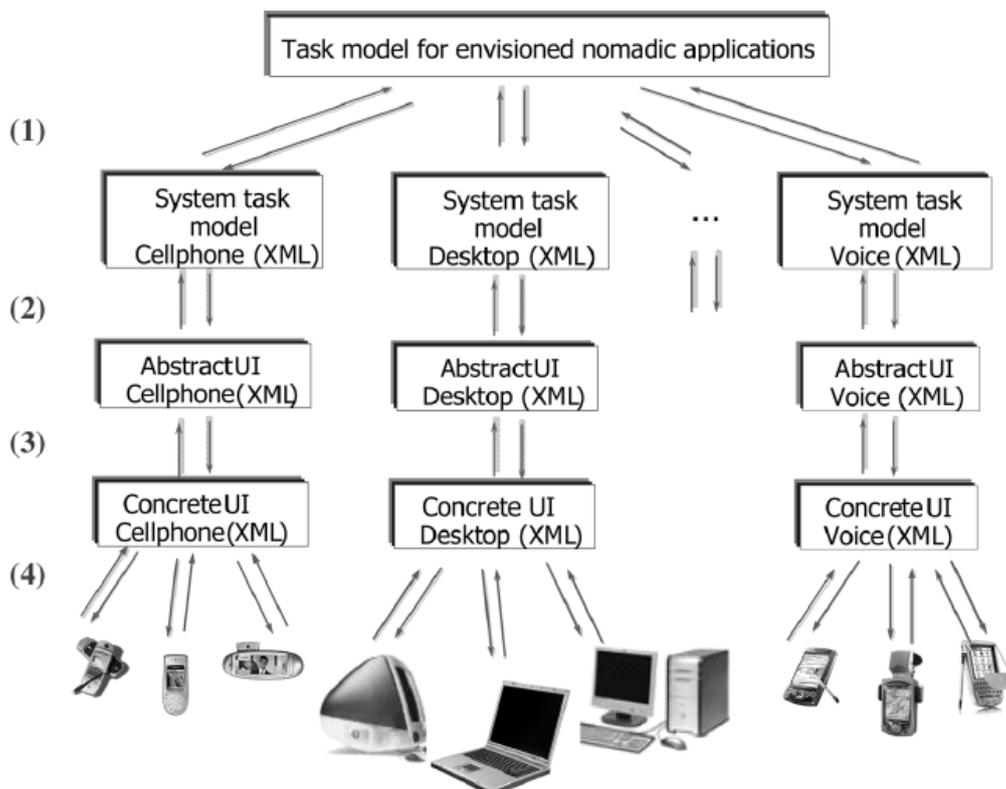


Figura 47. En Multimodal Teresa se pueden diseñar diferentes interfaces de usuario, para diferentes plataformas, a partir de un mismo modelo de tareas

La Figura 48 muestra un ejemplo en UsiXML a partir del cual se genera una sencilla interfaz.

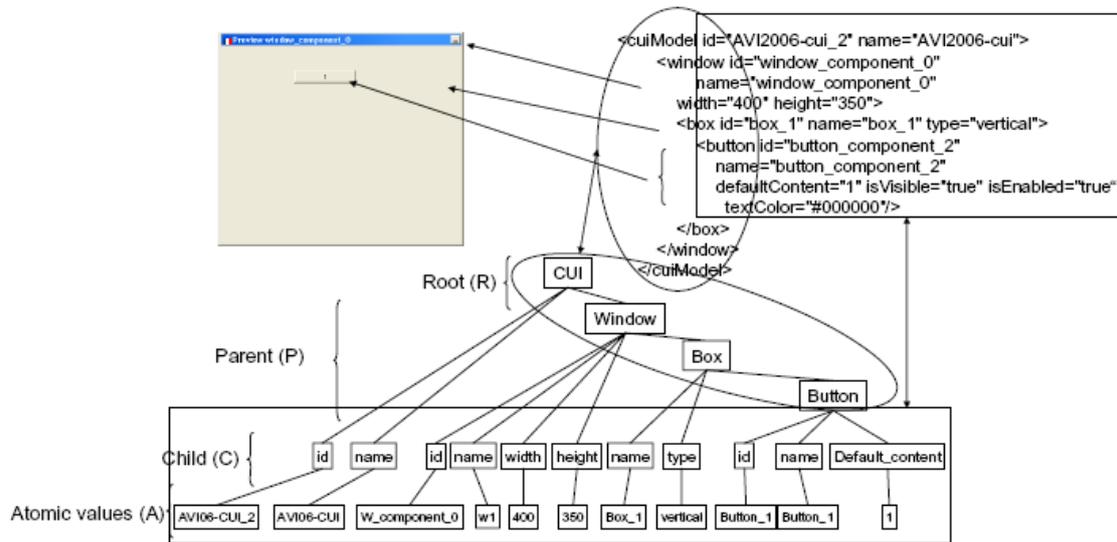


Figura 48. Una interfaz de usuario a partir de un esquema en UsiXML. Extraída de [Lepreux, 2006]

3.9 Metodologías para el Desarrollo de Entornos Colaborativos

La Ingeniería del Software ha propuesto numerosas metodologías para el desarrollo de software desde la concepción de la idea hasta el mantenimiento del sistema desarrollado.

Ya en el campo de la interacción persona-ordenador, en HCI, se han realizado esfuerzos para desarrollar metodologías y procesos para diseñar interfaces. Normalmente HCI se refiere a la interacción establecida entre un usuario y la computadora, pero estas técnicas e ideas se pueden extender para mejorar la comunicación y la cooperación entre grupos de usuarios y entre los usuarios de esos grupos: interacción persona-ordenador-persona.

Las metodologías HCI tratan de combinar características propias de la ingeniería del software con las de los sistemas de interacción persona-ordenador. Una característica de las metodologías HCI más recientes es su tendencia a centrarse en el continuo feedback y las conversaciones mantenidas entre usuarios, diseñadores, e ingenieros desde el inicio hasta el final del proceso de desarrollo. El proceso de desarrollo está totalmente centrado en el usuario. A menudo, los proyectos centrados en el usuario se apoyan en estudios etnográficos del entorno en el que interactúan con el sistema dichos usuarios. Esto es especialmente importante en entornos CSCW pues el usuario no sólo interactúa con el sistema, sino que además se tiene en cuenta su interacción con el resto de usuarios. Esto añade una complejidad que requiere nuevos métodos y técnicas, y lógicamente nuevas metodologías que guíen el desarrollo de estos sistemas.

Algunos autores piensan que estos sistemas CSCW se deben tratar de un modo diferente al resto de sistemas, de manera que se consideren aspectos como el modelado del grupo o de las tareas. El uso de metodologías específicas produciría unos resultados mejores puesto que se recogen y modelan estos aspectos colaborativos de acuerdo con el mundo real.

En este tipo de entornos se deben tener en cuenta diferentes características especiales puesto que el enfoque es diferente. Se trata de sistemas en los que participan muchos usuarios de formas muy diferentes: concurrentemente o no, distribuidos en el espacio o no, etc.

Sin embargo, las metodologías para el desarrollo específico de entornos colaborativos no son tantas. En [Dumont, 2001] se describe un método para la especificación de interfaces a través del uso de escenarios, en [Zhao, 2001] se muestra el modelado del proceso de un diseño cooperativo, en [Kirsh, 2004; Morris, 2004] se habla acerca de metodologías para la evaluación de sistemas colaborativos, etc.

En los siguientes apartados se muestran algunas metodologías de desarrollo de software centras en algún aspecto de los entornos colaborativos.

3.9.1 AMENITIES

AMENITIES (A METHodology for aNalysis and desIgn of cooperaTive systEmS) [Garrido, 2003] es una metodología que surge con el objeto de abordar la complejidad de los entornos colaborativos. Se centra en el grupo y cubre aspectos relevantes de su comportamiento y estructura.

Las fases generales de la metodología son:

1. Análisis del sistema y obtención de requisitos.
2. Modelado del sistema cooperativo.
3. Análisis del modelo cooperativo.
4. Diseño del sistema, introduciendo nuevos elementos o cambios en el modelo cooperativo, probablemente como resultado del análisis anterior.
5. Desarrollo del sistema software.

La metodología sigue un proceso iterativo que permite el refinamiento del modelo como consecuencia de su análisis, la revisión de los requisitos, etc. En el esquema general de la metodología se identifican cuatro modelos:

- El *Modelo de Requisitos*, donde se lleva a cabo la obtención y especificación de requisitos por medio de técnicas como la etnografía aplicada y los casos de uso de UML. En primer lugar se elicitan los requisitos por medio de la etnografía, para especificarlos y estructurarlos posteriormente por medio de los casos de uso.
- El *Modelo Formal* permite llevar a cabo un análisis automatizado del sistema por medio de Redes de Petri Coloreadas (CPN, Color Petri Nets), que proporcionan la semántica necesaria.
- El *Modelo Cooperativo (COMO)* es un modelo conceptual para describir y representar un sistema cooperativo desde su estructura y funcionamiento. Se emplea una notación gráfica, COMO-UML, para construir dicho modelo. La notación combina partes declarativas y operacionales.
- El *Modelo de Desarrollo Software* es el punto de conexión de la metodología con el desarrollo del software propiamente dicho. Hace uso de las notaciones UML: diagramas de paquetes y de componentes, diagramas de clases y de interfaces, diagramas de colaboración; y diagramas de componentes y despliegue.



Figura 49. Vistas conceptuales del modelo cooperativo de AMENITIES

3.9.1.1 Vistas Conceptuales del Modelo Cooperativo de AMENITIES

El modelo cooperativo el núcleo de AMENITIES, y el estudio de la faceta cooperativa de estos sistemas se lleva a cabo mediante cuatro vistas que muestran los aspectos más relevantes de los mismos:

- Vista de Grupo
- Vista Cognitiva
- Vista de Interacción
- Vista de Información

En la Figura 49 se muestra un esquema de las vistas conceptuales empleadas para representar un sistema cooperativo.

Vista de Grupo

La *Vista de Grupo* hace ver el sistema desde el entorno colaborativo donde se implantará, desde el escenario concreto, desde fuera.

Se describe la organización y las restricciones que se imponen dentro de esa asociación. Se ve la organización como un grupo de usuarios interrelacionados que realizan tareas.

Los *usuarios* son diferentes porque realizan *tareas* diferentes, es decir, tienen *roles* distintos.

Aparecen entonces tres conceptos importantes para esta vista. El *usuario*, que es el actor del escenario colaborativo. Las *tareas*, que serán las actividades y acciones a realizar para conseguir cierto objetivo. Y el *rol*, que determina la relación entre los usuarios y las tareas que deben llevar a cabo.

Las relaciones entre los usuarios se ven condicionadas por restricciones impuestas al sistema colaborativo:

- *Capacidades*: conocimientos que debe adquirir un usuario para desempeñar un rol determinado.
- *Leyes*: reglas del grupo deducidas por la propia estructura social.

Se trata de una vista dinámica, en el sentido de que con el paso del tiempo los usuarios pueden adquirir nuevas capacidades y las leyes pueden cambiar. Lógicamente, las organizaciones no son estáticas y esos cambios se deben reflejar en el sistema.

Vista Cognitiva

La *Vista Cognitiva* representa el conocimiento que posee o adquiere cada miembro del grupo en el escenario colaborativo. Según ese conocimiento, surgen una serie de tareas que se describen en esta vista, primero a grandes rasgos, y luego con un nivel de granularidad más fino.

El análisis de las tareas conlleva un estudio de las actividades que se han de realizar en el grupo; la división del trabajo, pues es sabido que descomponer un problema grande en varios pequeños ("divide y vencerás") suele resultar más productivo, eficiente y clarificador; y la interrelación de las diferentes tareas, ya que, aunque se trate de objetivos distintos con actividades distintas, las tareas casi siempre están relacionadas de alguna manera y no se puede pasar por alto este asunto.

El análisis de tareas se realiza en dos fases:

- En primer lugar se define el interfaz de rol.
- Se identifican todas las tareas y se describen las características principales de cada una de ellas
- Se detecta la relación existente con otros participantes (tareas) y entorno (eventos).
- Se estudia si puede ser interrumpida por otra tarea, cuál es su naturaleza cooperativa, cuál es el mecanismo de activación y el modo de sincronización.
- A continuación se describen en detalle y se pormenorizan cada tarea recogiendo también información de otras vistas. Secuencialidad, concurrencia, optatividad, decisiones, etc.

Vista de Interacción

En la *Vista de Interacción* se realiza un estudio de los procesos que implican un *diálogo* entre los participantes. Se establece cuál es el modo de diálogo o los requisitos que se imponen sobre los medios a utilizar.

Los modos de diálogo se establecen mediante *protocolos*. Por ejemplo, protocolos democráticos en los que la toma de las decisiones se hace por mayoría, protocolos de consenso en los que la decisión se toma por aprobación unánime, o protocolos jerárquicos.

Vista de Información

Por último, en la *Vista de Información* se ha de recoger toda la *información* que se comparte sobre el escenario colaborativo. Se puede tratar de *documentos*, que es la información que se gestiona, *eventos* o *recursos*. Además puede ser *implícita* en actividades y acciones o *explícita* como flujo entre actividades.

Las vistas se modelan mediante diagramas UML [Rumbaugh, 1999] modificados y extendidos para reflejar la semántica de grupo.

En el proceso de desarrollo se identifican tres fases:

- En la primera se obtiene el modelo de requisitos mediante casos de uso, técnicas de etnografía aplicada o modelos teóricos.
- La segunda en la que se realiza el modelo del sistema cooperativo con sus características especiales.
- La última que permite pasar a una primera aproximación del diseño inicial del sistema.

Actualmente están trabajando sobre esta última fase teniendo en cuenta dos aspectos:

- Se realiza una representación del sistema colaborativo usando estructuras de naturaleza hipertexto que permiten representar aspectos estáticos y dinámicos
- Se identifican patrones que facilitan la construcción de estas estructuras y el posterior diseño de las aplicaciones encargadas de gestionarlas.

Además están trabajando en el paso del modelo a métodos formales para que permita describir la semántica del sistema. En concreto se trata de redes de petri coloreadas.

3.9.1.2 Marco Conceptual del Modelo Cooperativo

AMENITIES define el marco conceptual en el que se basa para modelar la parte cooperativa de los sistemas. En dicho marco se describen algunos términos tal y como se puede apreciar en la Tabla 9. Así mismo, se define un conjunto de relaciones entre los conceptos mencionados con anterioridad:

- Asociación general. Representa una relación entre instancias de los conceptos relacionados.
- Agregación. Representa que un concepto es un agregado de otros.
- Especialización/generalización. Indica una relación entre un concepto más general y varios más específicos.

La notación empleada para expresar el modelo cooperativo, COMO-UML, está basada en UML. Concretamente parte de los diagramas de estados y actividades, sobre los que realiza una serie de extensiones para su aplicación a sistemas cooperativos.

Tabla 9. Conceptos definidos en el marco conceptual de AMENITIES

Concepto	Definición
Evento	Ocurrencia de algún hecho que tiene una localización tanto en el espacio como en el tiempo.
Acción	Unidad básica de trabajo ejecutable atómicamente.
Artefacto	Dispositivo (hardware y/o aplicación software) utilizado para llevar a cabo ciertas acciones.
Objeto de información	Entidad que contiene la información requerida para llevar a cabo acciones, o que se genera como resultado de éstas.
Subactividad	Unidad de trabajo formada por un conjunto de acciones y otras subactividades que permite estructurar tareas.
Tarea	Conjunto de subactividades/acciones cuya realización permite alcanzar objetivos.
Actor	Usuario, programa o entidad que puede desempeñar roles.
Rol	Comportamiento esperado de un actor en base a un conjunto identificable de tareas a realizar.
Capacidad	Habilidad o responsabilidad asociada a un actor que le permite desempeñar roles y llevar a cabo tareas, subactividades o acciones.
Tarea cooperativa	Tarea en la cual para su realización interviene más de un actor, bien desempeñando el mismo rol o distintos.
Organización	Conjunto de roles, y relaciones entre ellos, que se dan en un lugar de trabajo.
Ley	Norma impuesta por una organización que restringe su funcionamiento en base a reglas sociales, culturales, capacidades de los actores, etc.
Grupo	Conjunto de actores desempeñando roles que pertenecen a una misma organización o que participan en la realización de tareas cooperativas.
Protocolo de interacción	Conjunto de reglas de comportamiento que utilizan los miembros de un grupo para llevar a cabo subactividades.

3.9.2 CIAM: una Propuesta Metodológica para el Desarrollo de la Interfaz de Usuario en Sistemas Groupware

[Molina, 2006] propone un marco metodológico, CIAM (Collaborative Interactive Applications Methodology), basado en un conjunto de modelos que permiten a los ingenieros guiar el proceso de diseño y desarrollo de la Interfaz de Usuario en aplicaciones interactivas para el trabajo en grupo.

Esta metodología propone una serie de vistas para centrar la atención del desarrollo en diferentes aspectos. Las vistas no son independientes unas de otras, sino que son complementarias y están relacionadas.

3.9.2.1 Vistas para el Modelado en CIAM

En la metodología se plantean diferentes vistas para el diseño de la interfaz de un sistema groupware: *Vista de Organización*, *Vista de Inter-Acción*, *Vista de Datos* y

Vista de Interacción. De esta manera se modelan los aspectos que consideran más relevantes en este tipo de sistemas.

Vista de Organización

La organización está formada por el conjunto de roles a los que debe dar soporte la herramienta groupware que se pretende diseñar.

En la Vista de Organización se tienen en cuenta los diferentes aspectos relativos a la estructura de la organización en la que se desarrolla el trabajo en grupo: *organización, rol, grupo, agente software, actor y equipo de trabajo.* Ver Tabla 10.

Tabla 10. Definición de conceptos de la *Vista Organizacional* en [Molina, 2006]

Concepto	Definición
Vista Organizacional	
Organización	Estructura social en un entorno de trabajo formada por un conjunto de roles.
Rol	Se define un rol como un conjunto de actividades o responsabilidades que pueden ser desempeñadas por uno o más usuarios de un sistema en un momento determinado.
Grupo	Conjunto de entidades (actores que desempeñan roles) que pertenecen a una misma organización o que participan en la realización de tareas cooperativas o colaborativas.
Agente Software	Entidades software que realizan un conjunto de operaciones en nombre de un usuario o de otro programa con un cierto grado de independencia o de autonomía. Además de ser autónomo debe ser reactivo, social y con iniciativa propia.
Actor	Un actor es una entidad activa (humana o no, individual o grupal), que puede desempeñar uno o más roles en un determinado momento dentro del sistema.
Equipo de Trabajo	Conjunto de roles que trabajan de forma conjunta en una tarea de trabajo en grupo

Adicionalmente se identifican *relaciones jerárquicas* entre los roles así como *relaciones de especialización/generalización* de roles, donde un rol puede ser generalizado o especializado en otro.

Cabe destacar la definición de equipo de trabajo como conjunto de roles que trabajan de forma conjunta en la realización de cierta tarea de trabajo en grupo. Existe pues una diferencia entre grupo y equipo. Un equipo es un tipo específico de grupo. Concretamente se emplea la definición de equipo de trabajo propuesta por [Campos, 2005]: *“Un número reducido de personas con habilidades complementarias, que tienen un compromiso con un propósito común, una serie de metas de desempeño y un enfoque, de todo lo cual son mutuamente responsables”.*

Para abordar el modelado de la organización se emplea una notación similar a los diagramas de clases de UML, utilizando un icono representativo para cada uno de los elementos (rol, actor, grupo, etc.).

Vista de Inter-Acción

Esta vista permite modelar la estructura y el flujo de trabajo, donde el concepto principal es la tarea. Una tarea es una actividad desarrollada por actores para alcanzar un cierto objetivo.

En la Vista de Inter-Acción se tienen en cuenta conceptos como *proceso CSCW*, *tarea evento*, *interdependencia*, *tarea de trabajo en grupo*, *tarea individual*, *tarea colaborativa* y *tarea cooperativa*. Ver Tabla 11.

Para la representación del flujo de trabajo se emplean los operadores temporales incluidos en CTT.

Tabla 11. Definición de conceptos de la *Vista de Inter-Acción* en [Molina, 2006]

Concepto	Definición
	Vista de Inter-Acción
Proceso CSCW	Conjunto de actividades necesarias para alcanzar un objetivo en el seno de una organización. El proceso incluye una organización específica de las actividades de trabajo en grupo. El proceso incluye información acerca de la comunicación y la coordinación necesaria para llevar a cabo el trabajo. Muestra como el flujo de control pasa de una actividad a otra.
Tarea	Actividad realizada por los actores de una organización para alcanzar un cierto objetivo.
Evento	Ocurrencia de algún hecho que tiene lugar en un instante concreto de tiempo y que produce un cambio en el estado del sistema.
Interdependencia	Relación que se puede dar entre las tareas que definen el flujo de trabajo en la organización. Esta podrá ser de distintos tipos: temporal, de datos y de notificación.
Tarea de Trabajo en Grupo	Tarea en cuya realización está envuelto un equipo de trabajo
Tarea Individual	Tarea desempeñada por un único rol
Tarea Colaborativa	Tarea de trabajo en grupo en la que intervienen varios roles trabajando de forma coordinada en una tarea común que da un resultado común.
Tarea Cooperativa	Tarea de trabajo en grupo en la que existe una meta común, pero en la que los miembros del equipo trabajan de forma independiente o sobre porciones distintas del espacio de información común.

Para el modelado de la inter-acción se emplean diagramas de estados que representan la tarea a realizar, los roles envueltos y los objetos manejados, quedando en los arcos de comunicación la información relativa a la información transferida, las condiciones de ejecución y los operadores temporales.

Previamente se utiliza una representación llamada *tabla de participación*, unos diagramas inspirados en los diagramas de actividad de UML, que muestra de forma sencilla cuáles son los roles envueltos en una tarea, sin especificar relaciones temporales.

Vista de Datos

En la Vista de Datos se modelan los datos manipulados durante el trabajo en grupo, por lo que se tienen en cuenta conceptos como *contexto compartido*, *objeto*, *atributos*, *operaciones*, *objeto compuesto* y *objeto simple*. Ver Tabla 12.

Para el modelado de esta vista se emplea un diagrama de clases de UML.

Tabla 12. Definición de conceptos de la *Vista de Datos* en [Molina, 2006]

Concepto	Definición
Vista de Datos	
Contexto Compartido	Se denomina al conjunto de objetos (documentos, etc), y acciones que pueden realizarse sobre ellos, que son accedidos y manipulados en el seno de una actividad de trabajo en grupo.
Objeto	Entidad que contiene la información necesaria para la realización de las tareas, o que se genera como resultado de éstas.
Atributos	Pares (Identificador, Valor) que definen propiedades de los objetos del modelo de datos. Definen la estructura del objeto.
Operaciones	Operaciones a realizar sobre los objetos de la vista de datos. Definen el comportamiento del objeto.
Objeto Compuesto	Objeto de información compuesto de objetos más simples.
Objeto Simple	Unidad mínima de información en el modelo de datos.

Tabla 13. Definición de conceptos de la *Vista de Interacción* en [Molina, 2006]

Concepto	Definición
Vista de Interacción	
Herramienta	Software empleado por los usuarios para realizar las actividades y alcanzar sus objetivos.
Herramienta Dependiente de la Tarea	Herramientas que dan soporte a las tareas directamente relacionadas con el dominio de la aplicación.
Herramienta Independiente de la Tarea	Herramientas de soporte a la comunicación y coordinación.
Interfaz de Usuario	Superficie de contacto entre los usuarios y la aplicación software.
Tarea Interactiva	Se define tarea interactiva como la actividad lógica necesaria para alcanzar un determinado objetivo, y a las que la IU debería dar soporte. Dependiendo de quien realice esta tarea en diálogo persona-ordenador que tiene lugar a través de la IU, se clasifica en tareas de interacción (iniciadas por el usuario) y tareas de aplicación (realizadas por la aplicación).
Tareas de Interacción	Tareas llevadas a cabo por el usuario, e iniciadas por él, al interactuar con el sistema.
Tareas de Aplicación	Tareas ejecutadas por la aplicación en el contexto del diálogo que se da entre el usuario y la aplicación interactiva.
Relación entre Tareas Interactivas	Relaciones de precedencia y paso de información que se dan entre tareas interactivas.
Interfaz de Usuario Colaborativa	También llamadas Interfaces Multiusuario, hace referencia a las IU de las aplicaciones groupware.
Áreas de Visualización	Secciones en las que se puede dividir el contexto compartido en una tarea de trabajo en grupo y que expresa la capacidad de ver o tener referencia a un objeto o conjunto de objetos de dicho contexto.

Vista de Interacción

La Vista de Interacción describe las herramientas y los aspectos interactivos de la aplicación. En esta vista son importantes los conceptos: *herramienta*, *herramienta dependiente de la tarea*, *herramienta independiente de la tarea*, *interfaz de usuario*, *tarea interactiva*, *tareas de interacción*, *tareas de aplicación*, *relación entre tareas*, *interfaz de usuario colaborativa* y *áreas de visualización*. Ver 0.

Para el modelado de la interacción se emplea CIAM (*Collaborative Interactive Applications Notation*), una extensión de CTT que incluye tres áreas de visualización:

- (a) visualización colaborativa, que representa el contexto compartido;
- (b) visualización individual, que representa la interacción individual de cada miembro del grupo y
- (c) el subárbol que especifica el diálogo con el área del contexto compartido que solo puede ser accedido de forma exclusiva por uno de los miembros del grupo a la vez.

3.9.2.2 Etapas de CIAM

Las etapas de las que consta la metodología CIAM (Figura 50) se centran en el modelado del grupo en una primera instancia, después en el modelado del proceso (cooperativo, colaborativo y de coordinación) y finalmente en el modelado de las tareas interactivas que realiza el usuario.

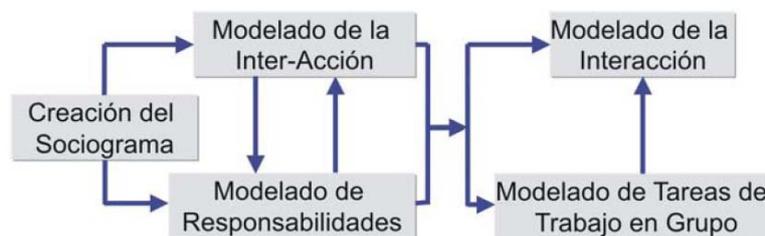


Figura 50. Etapas metodológicas de CIAM

La Tabla 14 describe las técnicas empleadas en cada una de las etapas que se describen posteriormente.

Creación del Sociograma

La primera etapa es la creación del *Sociograma* que representa la estructura de la organización y las relaciones que existen entre los diferentes miembros de la misma. Así, además de categorías como las identificadas en la Vista de Organización (*rol*, *actor*, *agente software*, *grupo*, *equipo de trabajo*), se identifican relaciones básicas entre ellos:

- Relaciones de *herencia*, donde se permiten heredar responsabilidades siempre que se dé una determinada procondición.
- Relaciones de *desempeño* entre actores y roles.
- Y relaciones de *asociación* entre roles, que muestran si existen situaciones donde dichos roles cooperan o colaboran entre sí.

Tabla 14. Etapas de CIAM y las técnicas empleadas en cada una

Etapa de CIAM	Técnica de especificación	Tipo de representación	Resultado obtenido
Creación del Sociograma	Sociograma	Gráfica	Representación de la estructura de la organización
Modelado de Responsabilidades	Modelo de Responsabilidades	Textual	Se obtiene la especificación detallada de las responsabilidades de cada rol
Modelado de la Inter-acción	Tabla de participación	Textual	Relación entre las principales tareas y los roles del sistema
	Modelo de Inter-acción	Gráfica	Representación de la estructura y flujo del trabajo a realizar por la organización
Modelado de tareas de Trabajo en Grupo	Matriz de Control de Acceso	Textual	Relaciona objetos de información (o atributos) y roles a nivel de tarea de trabajo en grupo
	Tabla de Permisos de Operación sobre el contexto compartido	Textual	Relaciona operaciones y roles a nivel de tarea de trabajo en grupo
Modelado de las Tareas Cooperativas	Grafo de Descomposición de Responsabilidades	Gráfica	Representación del reparto de responsabilidades en una tarea cooperativa
Modelado de Tareas Colaborativas	Especificación del Contexto Compartido (diagrama de clases en notación estándar UML)	Gráfica	Representación del contexto compartido de una tarea colaborativa, la división en áreas de visualización y la política de finalización
Modelado de la Interacción	Árbol de Descomposición de Tareas Interactivas (notación CTT ampliada con iconos para representar áreas de visualización en tareas colaborativas)	Gráfica	Modelado de la interacción entre cada rol y la aplicación; así como la interacción con el contexto compartido en tareas colaborativas

Modelado de Inter-acción

En segundo lugar, la etapa de *Modelado de la Inter-acción* especifican las tareas que definen el trabajo en grupo. Dichas tareas pueden ser clasificadas como *cooperativas, colaborativas o individuales*.

Existen una serie de relaciones entre procesos: *dependencia temporal, dependencia de datos, notificación*. También una serie de condiciones: de *terminación, de plazo y de ejecución*.

Se emplean las *tablas de participación* y el *modelo de inter-acción*.

Modelado de Responsabilidades

Tras especificar el trabajo en grupo se amplíane el conjunto de responsabilidades asignado a cada rol, añadiendo responsabilidades individuales.

Modelado de Tareas de Trabajo en Grupo

En esta etapa se definen con mayor nivel de detalle las tareas cooperativas y colaborativas identificadas en la etapa anteriores.

Por un lado se modelan las *Tareas Cooperativas* por medio de los *grafos de descomposición* de responsabilidades. Se detallan las sub-tareas en las que se descomponen las tareas cooperativas

Por otro lado se modelan las *Tareas Colaborativas*, incluyendo la especificación de los roles envueltos en su ejecución y de los objetos del modelo de datos manejados por el equipo de trabajo. Se utiliza la especificación del *contexto compartido*.

Modelado de Interacción

En esta etapa se modelan únicamente aspectos de interacción persona-ordenador. Se emplean los *árboles de descomposición de tareas interactivas* en notación CTT.

CAPÍTULO 4. MODELO CONCEPTUAL PARA ENTORNOS CSCW

Los sistemas CSCW se caracterizan por la diversidad. Como se puede apreciar del estudio realizado en los capítulos anteriores, esta línea de investigación proviene del esfuerzo realizado por numerosos investigadores de diferentes áreas como la psicología, la filosofía, la sociología, la informática, etc. Incluso investigadores de diferentes áreas dentro de la propia informática han contribuido al desarrollo del CSCW.

Si bien es cierto que esta diversidad ha propiciado la aparición de este campo de investigación enriqueciéndolo con puntos de vista tan diferentes, también es cierto que ha existido desde siempre un problema relacionado con el uso del lenguaje, convirtiendo el desarrollo de proyectos en una auténtica "Torre de Babel".

Este es el primer problema al que nos enfrentamos a la hora de realizar nuestra primera incursión en el área y en este capítulo se muestra la solución que adoptamos para solucionarlo: un modelo conceptual que recoge los términos de uso frecuente definidos y relacionados de acuerdo con lo que los investigadores de más prestigio han propuesto en sus trabajos.

4.1 Una Ontología como Solución a la Diversidad de Vocabulario Empleado en el Desarrollo de Aplicaciones Groupware

Para solucionar el problema planteado existen diferentes alternativas. En este apartado se estudian algunas, cuyo resumen de ventajas e inconvenientes de uso se han detectado aparecen en la siguiente tabla, y se justifica la solución adoptada.

Tabla 15. Ventajas e inconvenientes en técnicas para la categorización y/o definición de conceptos

	Ventaja	Inconveniente
Taxonomía	<ul style="list-style-type: none"> • Sencillez • Claridad 	<ul style="list-style-type: none"> • Rigidez en la clasificación • Los elementos pueden encajar en diferentes clasificaciones: ambigüedad • Los elementos pueden no encajar fácilmente en ninguna clasificación • Relación entre los elementos demasiado sencilla • Más que para definir, están pensadas para clasificar
Ontología	<ul style="list-style-type: none"> • Novedad: Web 2.0 • Relaciones explícitas entre los elementos • Posibilidad de Publicación • Posibilidad de búsquedas avanzadas • Posibilidad de compartir el conocimiento • Amplia especificación de conceptos • Extensible • Alto nivel de formalismo 	<ul style="list-style-type: none"> • Dificultad si no se entiende la escala del problema • Diferentes enfoques sobre ontologías • Búsquedas más rígidas y menos eficaces que mediante el uso de otras técnicas como las folcsonomías • No pensadas para la categorización
Folcsonomía	<ul style="list-style-type: none"> • Sencillez • Novedad: Web 2.0 • Flexibilidad • Disminuye las barreras de la cooperación • Búsquedas más precisas • Interrelación de conceptos por etiquetas • Feedback por las opiniones sociales • Introduce "inteligencia humana" 	<ul style="list-style-type: none"> • No están pensadas para definir conceptos • Más bien pensados para categorizar • Las relaciones no son explícitas • Ambigüedad terminológica • Basadas en opiniones sociales • Escasa especificación de conceptos • Bajo nivel de formalismo
Microformato	<ul style="list-style-type: none"> • Sencillez • Novedad: Web 2.0 • Operan sobre estándares existentes • Centrados en el pensamiento humano • Extensibles 	<ul style="list-style-type: none"> • No para definir conceptos • Más bien pensados para clasificar • Basados en opiniones sociales • Escasa especificación de conceptos • Bajo nivel de formalismo

4.1.1 Planteamiento del Problema

En el capítulo anterior se ha planteado el problema que supone trabajar en entornos multidisciplinares como CSCW. Los mismos términos pueden tener significados diferentes según el campo de investigación correspondiente, e incluso en áreas diferentes del mismo campo de investigación se pueden encontrar divergencias terminológicas.

Los entornos colaborativos son un campo relativamente nuevo que carece de estándares. Trabajar con un vocabulario con términos comunes con significados notablemente dispares es un problema que necesita una solución inmediata, antes de poder continuar con cualquier tipo de discusión.

4.1.2 Selección de una Solución

En la discusión sobre el estado del arte del capítulo anterior (ver apartado 3.1) se describen algunas técnicas empleadas en la descripción, caracterización, clasificación y uso de conceptos.

La *taxonomía* es una técnica para la clasificación de términos, conceptos, elementos en general, normalmente en forma de árbol con relaciones padre-hijo. El resultado es una clasificación rígida donde los elementos que se quieren clasificar no siempre encajan en alguna de las clasificaciones identificadas, e incluso se puede dar el caso de que un elemento encaje en más de una, creando ambigüedad.

Los términos se pueden definir de todos modos puesto que sería independiente de la taxonomía. Sin embargo, las relaciones entre ellos se limitarían a las de padre-hijo. Las taxonomías están pensadas para clasificar conceptos, más que para definirlos. Por todo ello no son una técnica que pueda resolver el problema planteado.

Las *ontologías* son un concepto relativamente novedoso que permite definir conceptos con claridad, expresar sus relaciones de una manera completa, así como publicar y compartir esa información. Se pueden realizar búsquedas avanzadas sobre las mismas, son extensibles y están ideadas para realizar especificaciones amplias de los conceptos. En cualquier caso las búsquedas serían más eficientes con otras técnicas como las *folcsonomías*.

Lo que se plantea en esta tesis es la necesidad de un vocabulario común para el desarrollo de entornos colaborativos por medio de la metodología presentada. En principio no se pretende elaborar un estándar con el vocabulario necesario para los entornos colaborativos, algo que, por otro lado, sería demasiado ambicioso.

Puesto que la escala del problema es menor que la de proponer una solución universal, en principio las ventajas de publicación, búsqueda y compartición de conceptos que introducen las ontologías complejas de Informática no son necesarias. De hecho, no entender la escala de este problema y hacer uso de una ontología con toda su potencialidad sólo introduciría una complejidad innecesaria.

Las *folcsonomías* son una técnica novedosa dentro de la Web 2.0 que permiten categorizar elementos por medio de etiquetas de una manera muy sencilla y flexible.

Los usuarios introducen esas etiquetas sobre los conceptos disminuyendo las barreras de la cooperación y permitiendo un feedback basado en opiniones sociales. De esta manera las búsquedas son más eficientes, puesto que se agilizan por medio de las etiquetas que caracterizan los elementos. Así mismo, los elementos podrían formar una red de interrelación, pudiendo navegar por conceptos relacionados que

podrían ser del interés de la comunidad. Se dice que el uso de folcsonomías dota a los sistemas de "inteligencia humana".

Sin embargo, el hecho de que esté basado en opiniones sociales reduce el nivel de formalidad de la caracterización, pudiéndose incluso encontrar términos sinónimos caracterizando elementos. El empleo de estos sinónimos también alejaría a los usuarios de aquello que pretenden encontrar.

El uso de folcsonomías para la elaboración de un modelo conceptual no sería adecuado porque esta técnica está pensada para categorizar conceptos, no para definirlos con precisión, por lo que la especificación no es suficiente.

Otro concepto nuevo que aparece en el entorno de la Web 2.0 es el de *microformato*. También se trata de una técnica sencilla que, en este caso, opera sobre estándares existentes y está centrada más en el pensamiento humano que en la propia máquina. Extiende estándares con nuevos atributos y valores sobre dicho atributos (es extensible), lo que permite categorizar la información. Sin embargo, tal y como ocurre con los microformatos, no están pensados para definir conceptos y el nivel de formalismo es bajo.

En definitiva, después de estudiar cada una de las técnicas se concluye elaborar un modelo conceptual basado en el uso de ontologías de un modo "relajado".

Las ontologías permiten la definición completa y formal de los conceptos que se emplearán en el desarrollo de los entornos colaborativos, pero se dice que se hará de un modo "relajado" porque, debido a la escala del problema planteado, no será necesario el uso de lenguajes como OWL para publicar y compartir la ontología.

Las características necesarias para llevar a cabo esta definición conceptual las cubren las ontologías:

- Se necesita una técnica que permita la definición conceptual, haciendo uso de un conjunto de metadatos para realizar una definición amplia y completa.
- Un grado de formalidad alto, sin necesidad de que sea absoluto.
- Interrelación explícita de los diferentes conceptos, que se muestra por medio del modelo conceptual.
- La posibilidad de que este modelo conceptual pueda ser extendido. La inclusión de nuevos elementos es previsible, más aún si se extiende el modelo de colaboración a otros campos como el e-learning o la realidad virtual.

Otras características propias de las ontologías más completas del campo de la informática no son necesarias para este caso. Adecuando la posible solución mediante el uso de ontologías a la escala del problema, se prescinde de algunas características de las mismas. Sin embargo, estas características se podrían incluir en cualquier momento si fuera necesario, extendiendo de nuevo su funcionalidad y sus ventajas:

- Se prescinde de la posibilidad de publicación y compartición del conocimiento de forma digital mediante OWL y Web Semántica, puesto que se pretende definir el vocabulario necesario para el desarrollo de entornos colaborativos mediante la metodología propuesta. No se pretende consolidar este vocabulario como un estándar.
- No se necesita realizar búsquedas avanzadas sobre los conceptos puesto que no se trata de un conjunto amplio y lo que se pretende es definirlos y relacionarlos bien para poder formar un vocabulario de manera que se pueda "hablar con propiedad". De hecho este es uno de los puntos que nos lleva a no utilizar técnicas de categorización como las folcsonomías o los microformatos.

4.1.3 Descripción de la Solución Propuesta

El desarrollo de ontologías en general y la definición de términos en torno a los trabajos colaborativos en particular es un esfuerzo que se ha realizado desde hace muchos años para concretar un vocabulario común a un grupo de personas que trabajan en un mismo campo, de manera que todas hablen con propiedad un "mismo lenguaje". De esta manera se evitan incoherencias, imprecisiones, ambigüedades y se avanza adecuadamente en el estudio de una materia determinada. La sección de trabajos relacionados del Capítulo 1 discute algunas de las aproximaciones llevadas a cabo en ese sentido.

Los conceptos que aparecen en esta ontología se concretan por medio de una definición, una definición contextual, unos posibles sinónimos en castellano, los términos anglosajones empleados, la notación empleada y el metamodelo parcial asociado:

- La *definición* es una proposición que expone con claridad y exactitud los caracteres genéricos y diferenciales del concepto en el marco de las estructuras organizativas de los usuarios de un sistema y las colaboraciones que se puedan dar en ella.
- La *definición contextual* expone con una o varias proposiciones la relación del término con otros términos del modelo conceptual. Especifica el concepto según el entorno terminológico que lo enriquece y/o modifica sutilmente.
- Los *sinónimos* no son deseables en una definición ontológica, pero en caso de darse se muestran en este metadato.
- Debido a la influencia del inglés en el campo de la investigación informática, se ha considerado fundamental realizar una correspondencia de los términos empleados en castellano con los *términos anglosajones* utilizados.
- El metadato *nota* muestra algún comentario adicional de relevancia.
- La *notación* muestra la imagen gráfica utilizada para representar el concepto. No todos los conceptos tendrán una notación, porque algunos de ellos se utilizan de un modo más abstracto. Esta representación gráfica se utiliza en los diagramas que representan los modelos de la realidad que se especifica.
- El *metamodelo parcial* asociado describe la situación del concepto definido en el marco ontológico mediante notación UML 2.0 [OMG, 2005]. Muestra las relaciones con el resto de conceptos de un modo semi-formal. Se dice que es un metamodelo parcial porque es un detalle del metamodelo que representa las estructuras organizativas colaborativas de los usuarios de un sistema. Este metamodelo parcial sólo muestra las relaciones del concepto que se define con sus conceptos relacionados, no las relaciones de esos conceptos con otros conceptos.

Definir una ontología para especificar conceptos relacionados con el dominio de conocimiento "entornos colaborativos" proporciona un vocabulario común bien definido, una lista de términos y significados que describen los objetos del dominio y las relaciones entre ellos de un modo formal.

Existen multitud de trabajos en los que se definen términos relacionados con los entornos colaborativos: grupo, rol, actor, tarea cooperativa, etc. Las definiciones de todos esos conceptos pueden variar significativamente de un trabajo a otro. Con el objetivo de tener una base firme para especificar la estructura organizativa de los usuarios de un sistema colaborativo, así como las relaciones que tienen lugar entre ellos, se presenta esta ontología: un modelo conceptual que nos permite hablar un lenguaje común y evitar ambigüedades en el uso de algunos de estos términos.

El modelo conceptual que se propone en este trabajo es la base de una serie de diagramas que hemos definido para representar entornos colaborativos. Diagramas que dan soporte a una técnica de modelado centrada en el usuario como miembro de una organización que interactúa (ver Capítulo 5). Es decir, no se trata de un elemento aislado, sino de una parte de esa organización que tiene una serie de características, que pertenece a grupos, que desempeña roles, que realiza tareas cooperativas para lograr objetivos que no serían alcanzables de forma individual. El vocabulario empleado para especificar esta estructura y estas relaciones entre usuarios es lo que define esta ontología.

Los términos definidos en la ontología aparecerán en el Modelo Conceptual relacionados unos con otros. La definición de cada uno de los términos se realizará junto con la descripción del modelo correspondiente de manera que la lectura resulte más sencilla.

4.2 Un Modelo Conceptual para la Especificación de Entornos Colaborativos: Descripción de los Modelos

El apartado anterior ilustra la necesidad de un glosario de términos básico para el diseño de un sistema colaborativo. Se han agrupado y relacionado los términos de manera que la especificación de un entorno colaborativo se pueda realizar a partir de distintos modelos que se describen en los siguientes subapartados.

Los modelos (Figura 51) reflejan la realidad colaborativa del entorno y se integran como parte de la metodología para especificar las necesidades del sistema desde diferentes puntos de vista complementarios que llevan a una especificación completa y coherente del sistema colaborativo.

No se trata de modelos inconexos, sino que están relacionados unos con otros y existe una trazabilidad entre ellos, de manera que un cambio en uno podría suponer un cambio en algún otro, lo que está dentro de la lógica de la especificación del sistema desde diferentes puntos de vista. Esa trazabilidad asegura la coherencia de dicha especificación y quedará patente en la descripción de la metodología empleada en el Capítulo 5.

En primer lugar, la estructura organizativa de los usuarios de un entorno colaborativo se representa mediante el *Modelo de Organización*. Mediante este modelo se representa la estructura de los actores del sistema así como las relaciones de colaboración que tienen lugar entre ellos.

El *Modelo de Tareas* especifica las acciones individuales o colectivas que deben realizar los usuarios del sistema para conseguir objetivos determinados. También se especifican otro tipo de acciones que no realicen personas (tareas de aplicación o abstractas).

El *Modelo de Objetivos* representa las metas de los grupos y los fines para los que se llevan a cabo las tareas a cualquier nivel de granularidad.

El *Modelo de Sesión* proporciona una vista del sistema en un momento determinado. Es la representación de un escenario concreto.

La percepción de los usuarios sobre lo que los demás están haciendo en el sistema o awareness descrito en el apartado 3.6 no requiere un modelo aparte sino que es algo que, como se verá en la metodología descrita en el apartado 5.4, es algo que se debe tener en cuenta a lo largo del proceso, y especialmente en la etapa de diseño.

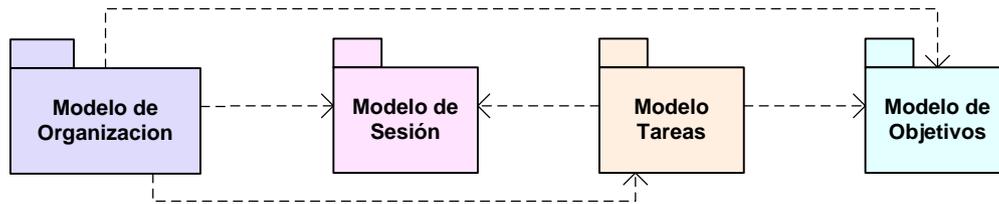


Figura 51. Relación entre modelos

En este apartado se describen los diferentes modelos identificados como necesarios para la especificación de un sistema colaborativo centrado en el usuario y dirigido por tareas, generando así una metodología de desarrollo de sistemas que permite crear interfaces de usuario que consideran la interacción entre los diferentes usuarios del un entorno CSCW a través de las máquinas y de las infraestructuras de red.

Como se comentó en el apartado anterior, por una lectura más sencilla, en cada modelo se definirán los términos asociados al mismo, es decir, esa parte de la *ontología*. Así mismo se describe la parte del *metamodelo* que sirve como base para la generación del modelo correspondiente. Por último, se describe el *diagrama* asociado para la realización de dicho modelado. Por lo tanto, el Modelo Conceptual u Ontología para los entornos colaborativos que se emplea en este trabajo queda definido por medio de la ontología propiamente dicha, como conjunto de términos bien definidos, el metamodelo que relaciona los conceptos y los diagramas y notaciones empleados en la representación del sistema por medio de los modelos.

4.2.1 Modelo de Organización

El Modelo de Organización es un modelo de un nivel de granularidad alto en el que se muestra gráficamente, por medio del Diagrama de Estructura Organizativa (OSD, Organizational Structure Diagram), la organización de los usuarios del sistema. Es decir, cómo se agrupan, qué relaciones tienen esos grupos, y qué roles se desempeñan en el sistema.

El Modelo de Organización surge de la necesidad de situar en el entorno colaborativo al usuario. El número de usuarios de las aplicaciones ha crecido exponencialmente en los últimos años, pero ya no sólo entendido como número de personas que hacen uso de la misma, sino también en cuanto a las posibilidades de uso de la herramienta por parte de los usuarios según desempeñan un rol u otro, según están en uno u otro grupo, etc.

El OSD permite reflejar gráficamente la organización de los usuarios del sistema. "De un vistazo" es posible tener una percepción de la situación de los roles en el sistema, de la agrupación de usuarios, de quién pertenece a qué grupo, etc. La estructura arborescente del diagrama permite ver a la organización, entendida como conjunto de todos los usuarios, roles, etc., como el mayor de todos los grupos. Es el nodo raíz del árbol, que se va ramificando para dividir el "problema" de grupo en problemas más pequeños.

Los conceptos identificados para poder realizar dicha representación de forma completa, coherente y correcta son: *estructura organizativa, elemento organizativo, relación organizativa, role, actor, grupo, individuo, usuario, agente, relación organizativa estructural, relación organizativa de grupo, relación de desempeño, relación de agregación, relación de jerarquía e co-interacción.*

4.2.1.1 Definición de Conceptos

Estructura Organizativa

Definición

La Estructura Organizativa de los participantes de un sistema colaborativo es la expresión de la distribución de sus elementos, incluyendo sus relaciones de pertenencia a grupos y sus relaciones de desempeño de roles.

Definición Contextual

Elemento Organizativo, Relación Organizativa

La *estructura organizativa* se representa mediante el *Diagrama de Estructura Organizativa (OSD)* que refleja el modelo organizativo. Se define de acuerdo a algunos conceptos que se conocen como *elementos organizativos* (*grupo, rol, actor, individuo, usuario y agente*) así como en base a algunas relaciones organizativas entre dichos elementos (*relación de desempeño, relación de agregación, relación de jerarquía y relación de co-interacción*). Por lo tanto, mediante la estructura organizativa se relacionan los elementos *rol, actor, grupo, individuo, usuario y agente*.

Rol Los *elementos organizativos* que constituyen la *estructura organizativa* desempeñan *roles* en el sistema que dan sentido a su existencia.

Restricciones Los *actores* del sistema pueden desempeñar *roles* o pertenecer a *grupos* según una serie de *restricciones* sobre los mismos.

Sinónimos

Organización

Término Anglosajón

Organizational structure

Nota

- *Estructura organizativa* es un concepto empleado dentro del lenguaje usual para el diseño de entornos colaborativos, pero no existe una representación física en los diagramas que modelan la realidad colaborativa.

De hecho, el Modelo de Organización presentado en el trabajo permite representar esa *estructura organizativa* y sus características por medio del OSD, diagrama que se explicará tras la explicación del Metamodelo de Organización.

Notación

No.

Término Anglosajón

Organizational item

Nota

No.

Notación

No. Ver notación particular para cada uno de ellos.

Metamodelo Parcial

No. Ver metamodelos parciales de cada uno de los *Elementos Organizativos*.

Rol

Definición

Un *rol* es el conjunto de tareas que puede desempeñar un *actor*.

Definición Contextual

Actor, Tarea Un *actor* del sistema no tiene sentido por sí mismo. Un *actor* es lo que es debido al *rol* que desempeña en cada momento en el sistema. Por lo tanto, un actor no tiene sentido en el sistema si no desempeña al menos un *rol*. Es por este motivo por lo que el concepto de *tarea* está ligado al de *rol* en el metamodelo. Un *actor* no tiene asociada directamente ninguna *tarea*, sino que las tareas están asociadas a los roles. Cada *actor*, por tanto, lleva a cabo tareas dependiendo del *rol* que desempeña.

Restricción, Actor Un *actor* podría desempeñar un *rol* solo si cumple con unas determinadas *restricciones*.

Sinónimos

Papel.

Término Anglosajón

Nota

Ver notas del concepto de *actor*.

Notación

Bajo la forma gráfica utilizada para representar el concepto de *rol*, y formando parte de la misma, se utiliza una etiqueta con el nombre de dicho *rol* en letras capitales.



ROLE_1

Figura 53. Notación empleada para representar el concepto de *rol*

Metamodelo Parcial

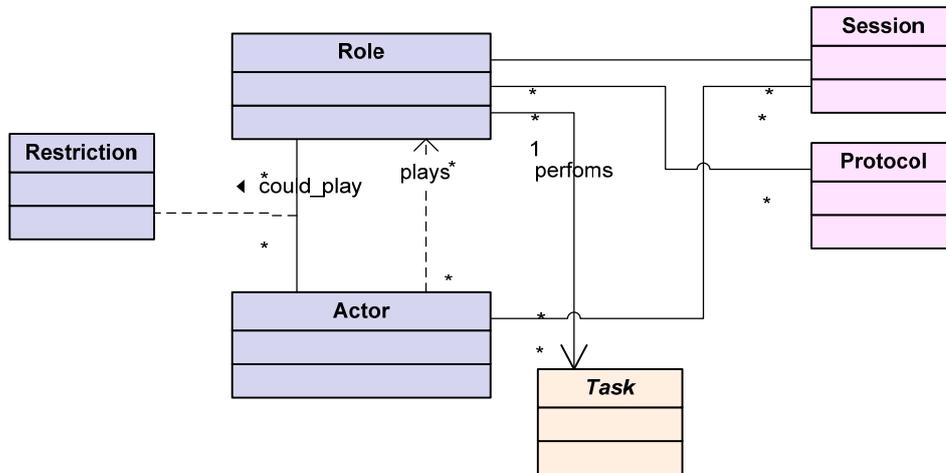


Figura 54. Metamodelo parcial del rol

Actor

Definición

Un *actor* es una o varias personas, u otro u otros sistemas externos, que interactúa con el sistema.

Definición Contextual

Rol, Tarea El *actor* puede interactuar directamente con el sistema colaborativo llevando a cabo tareas individuales, o bien, puede interactuar con otros actores a través del sistema realizando de este modo tareas cooperativas. Esta es la clave de la diferencia entre interacción persona-ordenador y colaboración (interacción persona-ordenador-persona). En la primera el *actor* interactúa con el sistema, mientras que en la segunda el *actor* interactúa con otros actores en el sistema. En cualquier caso un *actor* realiza una *tarea* en función del *rol* que desempeña en ese momento, es decir, no hay una relación directa *actor-tarea*, sino que el *actor* realiza las tareas de acuerdo al *rol* desempeñado en un momento dado. La relación es, pues, transitiva a través del *rol*.

Grupo Un *actor* puede ser un elemento colectivo.

Individuo Un *actor* puede ser un elemento individuo.

Capacidad, Rol Un *actor* posee determinadas *capacidades* que le perdesempeñar ciertos *roles*.

Grupo, Individuo, Usuario, Agente Merece la pena destacar que se trabaja con el término *actor* como algo más genérico. Los términos *grupo*, *individuo*, *usuario* o *agente* caracterizan al término *actor*.

Sinónimos

No.

Término Anglosajón

Actor.

Nota

- El concepto de actor introducido por la ingeniería del software orientada a objetos [Jacobson, 1992] introduce una pequeña confusión, puesto que se emplea para hacer referencia al rol que juega en un sistema [Constantine, 1999]. Coloquialmente un actor sería una persona (por ejemplo) desempeñando algo, mientras que el rol es ese algo que está siendo desempeñado. En [Constantine, 1999], se hace esta discusión y terminan por utilizar el concepto *rol de usuario (user role)* para concretar el rol que juega un actor que es humano.
- Algo similar ocurre con el actor de los Casos de Uso en UML 2.0 [OMG, 2005]. En UML 2.0 (así como en su versión anterior), un actor especifica un rol desempeñado por un usuario u otro sistema que interactúa con el sujeto. El término rol para esta definición se emplea de un modo informal. Por tanto, en UML 2.0 rol y actor se podrían considerar sinónimos. Un actor modela un tipo de rol desempeñado por una entidad que interactúa con el sujeto, pero que es externo al sujeto (sujeto empleado aquí como el sistema).

Un actor en UML 2.0 puede representar roles desempeñados por humanos, hardware externo u otros sujetos (sistemas). Un actor no es necesariamente una entidad física.

El *actor* que se define en este trabajo para modelar las estructuras organizativas de los usuarios de un sistema colaborativo no representa los roles desempeñados por humanos, hardware externo u otros sujetos (sistemas). El *actor* definido aquí representa precisamente esos humanos, hardware externo u otros sistemas. El *actor* puede ser interno del sistema o externo al sistema, pero que interactúa con él.

En este sentido, la definición de actor y de rol, en este trabajo, se asemeja más a las utilizadas por [Constantine, 1999].

- Por otro lado, mencionar que se ha considerado que un *actor* sólo puede ser o bien un grupo, o bien un elemento *individuo* (ya sea un usuario o un *agente*) y que además un mismo *actor* no puede ser a la vez *grupo* y elemento *individuo*.

Notación

Bajo la forma gráfica utilizada para representar el concepto de *actor*, y formando parte de la misma, se utiliza una etiqueta con el nombre de dicho *actor* en letras capitales. Puesto que es un concepto genérico, podría omitirse el nombre del *actor* en su representación.



Figura 55. Notación empleada para representar el concepto de actor

Metamodelo Parcial

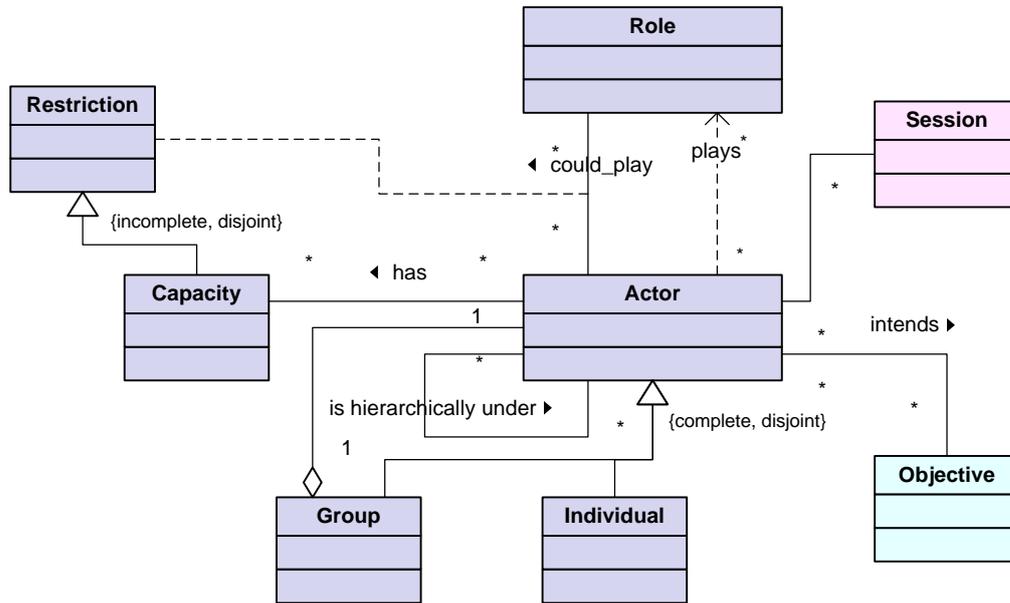


Figura 56. Metamodelo parcial del actor

Grupo

Definición

Un *grupo* es un conjunto de *actores*, individuos o colectivos, que desempeñan *roles* para lograr un *objetivo* común.

Definición Contextual

- Objetivo, Actor Un *grupo* es un conjunto de *actores* que necesitan interactuar y colaborar para alcanzar un *objetivo* común. Dichos *objetivos* comunes no serían realizables sin tal colaboración.
- Grupo Los *grupos* podrían formar parte de otros *grupos*. De esta manera, la *organización* entera podría verse como el mayor de los *grupos*.
- Ley Un *grupo* tiene unas *leyes* y un *actor* puede formar parte de un *grupo* si cumple con sus *leyes*.

Ver Actor.

Sinónimos

No.

Término Anglosajón

Group.

Nota

No.

Notación

Bajo la forma gráfica utilizada para representar el concepto de *grupo*, y formando parte de la misma, se utiliza una etiqueta con el nombre de dicho *grupo* en letras capitales.



Figura 57. Notación empleada para representar el concepto de *grupo*

Metamodelo Parcial

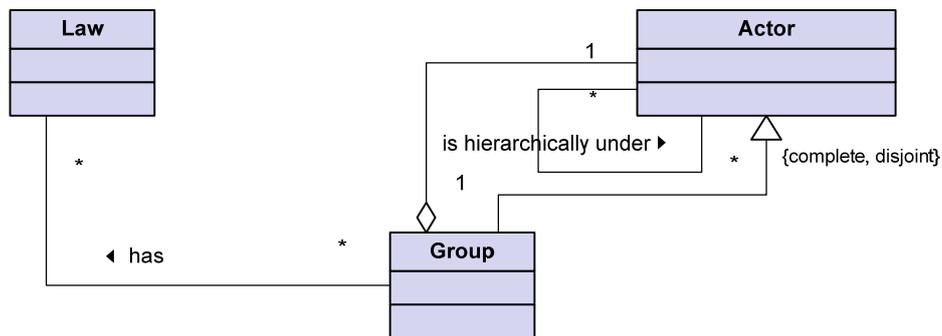


Figura 58. Metamodelo parcial del *grupo*

Individuo

Definición

Un elemento *individuo* es un, y sólo un, *actor* que desempeña *roles*.

Definición Contextual

Usuario, Agente El concepto *individuo* únicamente expresa unidad. Representa, por tanto, cualquier elemento no colectivo que interactúa con o en el sistema, ya sea una persona o un elemento impersonal, como podría ser una pieza de software que también interactúa con el sistema. En el primer caso se habla de un *usuario*. En el segundo de un *agente*.

Ver *Actor*.

Sinónimos

No.

Término Anglosajón

Individual.

Nota

- El término *individuo* referencia al adjetivo, a algo individual, no colectivo. Nótese cómo se hace uso del término *individuo* en la definición al tratarse de un adjetivo: "elemento *individuo*". Sin embargo, por comodidad, normalmente

se relajará el uso del lenguaje y se empleará como un sustantivo, siendo también válido "Un *individuo* es...".

- A pesar de que se contemplan estas dos únicas posibilidades como elemento *individuo* (*usuario* y *agente*), no se descarta que en una extensión del modelo conceptual aparezcan nuevos tipos de elemento *individuo*. Así mismo, se considera que un elemento *individuo* siempre será o bien un *usuario*, o bien un *agente*, es decir, ningún elemento *individuo* puede ser *usuario* y *agente* a la vez.

Notación

Bajo la forma gráfica utilizada para representar el concepto de *individuo*, y formando parte de la misma, se utiliza una etiqueta con el nombre de dicho *individuo*. Sólo la primera de las letras en capital. Puesto que es un concepto genérico, podría omitirse el nombre del *individuo* en su representación.



Figura 59. Notación empleada para representar el concepto de *individuo*

Metamodelo Parcial

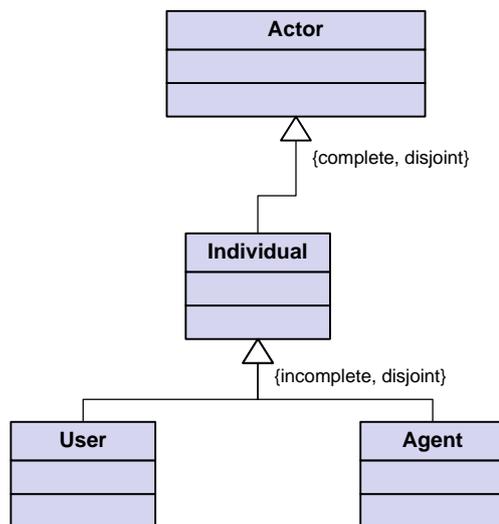


Figura 60. Metamodelo parcial del *individuo*

Usuario

Definición

Un *usuario* es un elemento *individuo* humano.

Definición Contextual

Ver *individuo* y *actor*.

Sinónimos

No. Podríamos hablar de *persona* o humano como sinónimos, pero sería un error puesto que son conceptos mucho más genéricos y no tiene porqué tratarse de elementos que interactúan con el sistema necesariamente.

Término Anglosajón

User.

Nota

No.

Notación

Bajo la forma gráfica utilizada para representar el concepto de *usuario*, y formando parte de la misma, se utiliza una etiqueta con el nombre de dicho *usuario*. Sólo la primera de las letras en capital.



Figura 61. Notación empleada para representar el concepto de *usuario*

Metamodelo Parcial

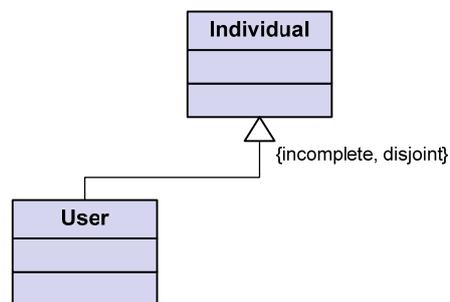


Figura 62. Metamodelo parcial de *usuario*

Agente

Definición

Un *agente* es un elemento *individuo* no humano.

Definición Contextual

Ver *Individuo* y *Actor*.

Sinónimos

Sistema externo. En cualquier caso se debe tener cuidado puesto que no es un sinónimo bidireccional. Un *sistema externo* es una agente siempre, sin embargo, existe la posibilidad de que un *agente* sea un sistema interno.

Término Anglosajón

Agent.

Nota

- Todo elemento *individuo* que no sea una persona se considera un *agente*. Así, un sistema software externo que interactuara con el sistema que se especifica sería un *agente*, así como un clásico agente inteligente, que mecanizara algunas tareas dentro del sistema, sería considerado también un *agente*.

Notación

Bajo la forma gráfica utilizada para representar el concepto de *agente*, y formando parte de la misma, se utiliza una etiqueta con el nombre de dicho *agente*. Sólo la primera de las letras en capital. La 'S' del icono representativo viene de sistema, aludiendo al tipo de *actor* que representa.



Figura 63. Notación empleada para representar el concepto de *agente*

Metamodelo Parcial

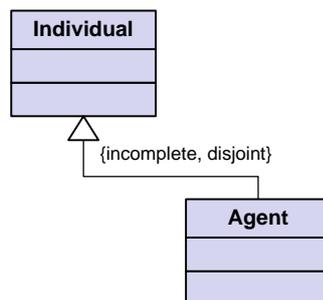


Figura 64. Metamodelo parcial de *agente*

Restricción

Definición

Una *restricción* es una condición sobre un *actor*.

Definición Contextual

Ley, Capacidad

Las *leyes* y *capacidades* tienen que ver con las políticas que gobiernan el comportamiento del *grupo*, y por tanto, con el comportamiento social de sus miembros: obligaciones, permisos, prohibiciones, etc. Así, el sistema impone las restricciones (leyes) que rigen su funcionamiento obligando a los elementos que lo forman a comunicarse, coordinarse y colaborar. Los *actores* adquieren/abandonan habilidades o responsabilidades (capacidades) que en ciertos casos las leyes demandan para llevar a cabo el trabajo.

Rol, Actor Las *restricciones* determinan si un *actor* puede, es capaz de desempeñar un *rol*.

Sinónimos

No.

Término Anglosajón

Restriction.

Nota

La definición contextual de *restricción* pertenece a [Garrido, 2003].

Notación

No.

Metamodelo Parcial

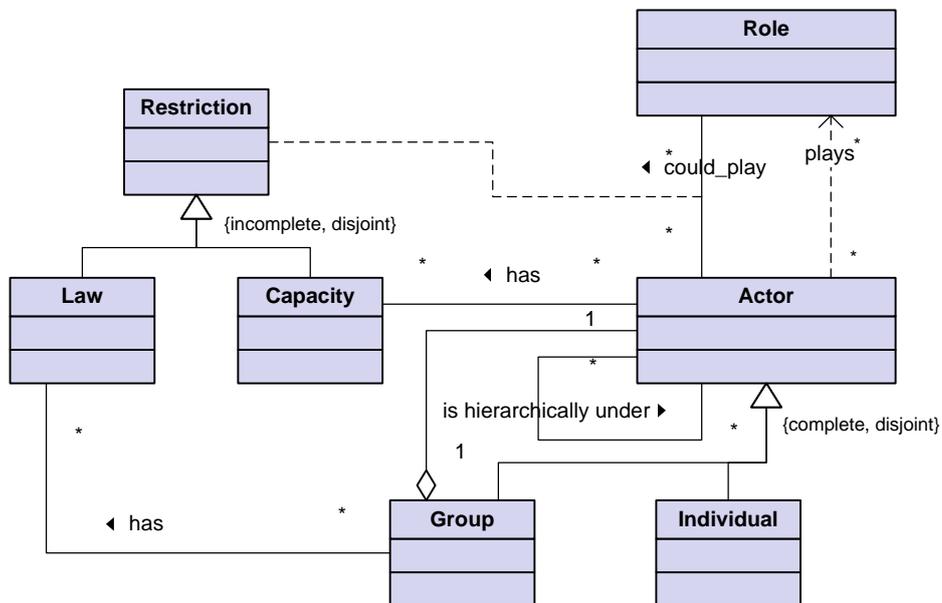


Figura 65. Metamodelo parcial de la *restricción*

Ley

Definición

Una *ley* es una norma impuesta por un *grupo* que restringe su funcionamiento en base a reglas sociales, culturales, capacidades de los *actores*, etc.

Definición Contextual

No.

Sinónimos

No.

Término Anglosajón

Law.

Nota

La definición de *ley* es una adaptación de la de [Garrido, 2003].

Notación

No.

Metamodelo Parcial

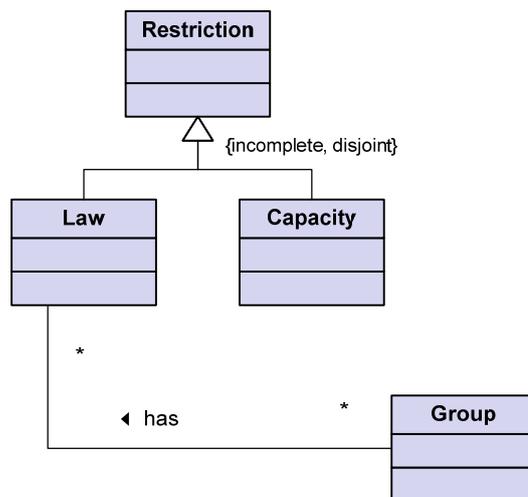


Figura 66. Metamodelo parcial de la *ley*

Capacidad

Definición

Una *capacidad* es una habilidad o responsabilidad asociada a un *actor* que le permite desempeñar *roles* y llevar a cabo *tareas*.

Definición Contextual

No.

Sinónimos

No.

Término Anglosajón

Capacity.

Nota

La definición de *capacidad* es una adaptación de la de [Garrido, 2003].

Notación

No.

Metamodelo Parcial

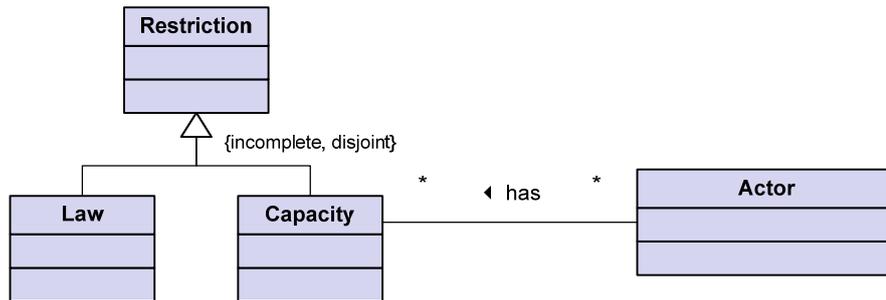


Figura 67. Metamodelo parcial de la *capacidad*

Relación Organizativa

Definición

Una *relación organizativa* es toda asociación existente entre dos *elementos organizativos* por la cual se establece la estructura lógica de los *actores* del sistema o se identifica una interacción entre ellos.

Definición Contextual

Rol, Actor (Grupo, Individuo, Usuario, Agente)

La *estructura organizativa* de los usuarios de un sistema se define de acuerdo a los *elementos organizativos* y a las *relaciones organizativas* que se dan entre ellos. Se distinguen dos tipos de relaciones entre *elementos organizativos*: *relaciones estructurales* y *relaciones colaborativas*.

Sinónimos

No.

Término Anglosajón

Organizational Relationship.

Nota

La siguiente figura clasifica las diferentes *relaciones organizativas*.

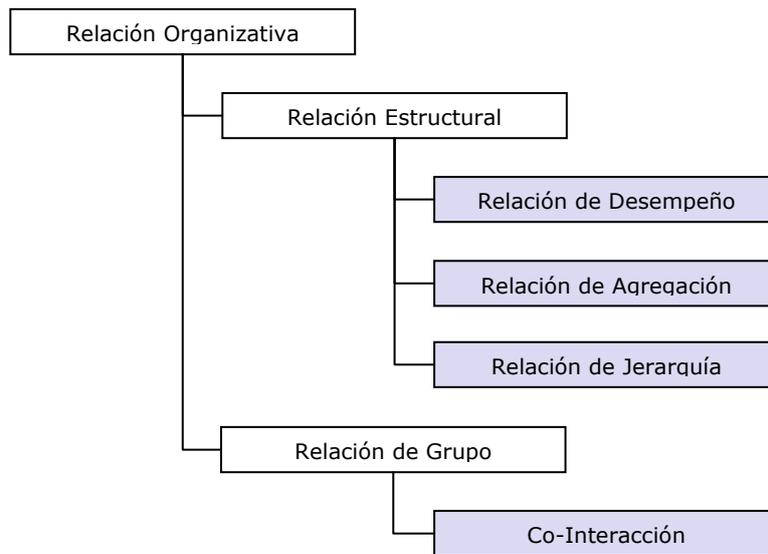


Figura 68. Esquema clasificatorio de las *relaciones organizativas* de un sistema.

Notación

No.

Metamodelo Parcial

No. Ver metamodelos parciales de cada una de las *relaciones organizativas*.

Relación Organizativa Estructural

Definición

Una *relación organizativa estructural* es una asociación entre *elementos organizativos* que describe las relaciones de dependencia entre ellos, definiendo de esta manera la estructura organizativa lógica de los *actores* del sistema.

Definición Contextual

Rol, Actor (Grupo, Individuo, Usuario, Agente)

Las *relaciones organizativas estructurales* describen las relaciones de pertenencia a *grupos* y los *roles* que desempeñan los *actores* del sistema. La *estructura organizativa* del los *actores* queda especificada por medio de todas estas relaciones de forma explícita. Existen tres tipos de *relación estructural* que permiten especificar lo anteriormente comentado: *relación de desempeño*, *relación de agregación* y *relación de jerarquía*.

Sinónimos

Relación estructural.

Término Anglosajón

Structural relationship.

Nota

No.

Notación

No. Ver notaciones de cada una de las *relaciones organizativas estructurales*.

Metamodelo Parcial

No. Ver metamodelos parciales de cada una de las *relaciones organizativas estructurales*.

Relación de Desempeño

Definición

Una *relación de desempeño* es una *relación organizativa estructural* que identifica el *role* que juega un *actor*.

Definición Contextual

Rol, Actor Por medio de una relación de desempeño es posible saber cómo es el *actor* (sus características) y lo que puede hacer (tareas) según el *rol* que desempeña.

Pero va más allá de esta relación. Es una relación de dependencia similar a la definida en UML 2.0 [OMG, 2005] que enriquece la semántica que existe en torno al concepto *role-actor*. Concretamente, por medio de esta relación se explicita que un *actor* no tiene sentido en el sistema sino por el/los *roles* que desempeña. Un *actor* no puede realizar nada en el sistema si no tiene asociado al menos un *role*.

Ver *Rol* y *Actor*.

Sinónimos

Desempeño.

Término Anglosajón

Play relationship.

Nota

No.

Notación



Figura 69. Notación empleada para representar el concepto de *relación de desempeño*

Metamodelo Parcial

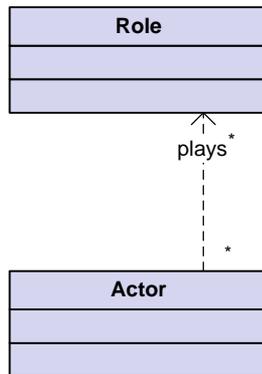


Figura 70. Metamodelo parcial de la *relación de desempeño*

Relación de Agregación

Definición

Una *relación de agregación* es una *relación organizativa estructural* que identifica una asociación entre el todo y las partes.

Definición Contextual

Actor, Grupo, Individuo

La *relación de agregación* define los elementos (*actores*) que pertenecen a cada grupo. La relación de agregación tiene sentido entre un *grupo* y los *actores* que lo forman. Esto significa que un *grupo* podría estar formado por otros *grupos* o bien por otros elementos organizativos individuales como *usuario* o *agentes*.

en la definición del sistema, se identifican los *roles* a priori y se agrupan según sea necesario, generando la *estructura organizativa* del sistema. Puesto que los *actores* no tienen sentido sin esa definición de *roles* por la relación de desempeños entre ellos, la afirmación anterior sigue teniendo sentido.

Sinónimos

Agregación

Término Anglosajón

Aggregation relationship.

Nota

Se trata de la definición tradicional de agregación en UML [OMG, 2005].

Notación



Figura 71. Notación empleada para representar el concepto de *relación de agregación*

Metamodelo Parcial

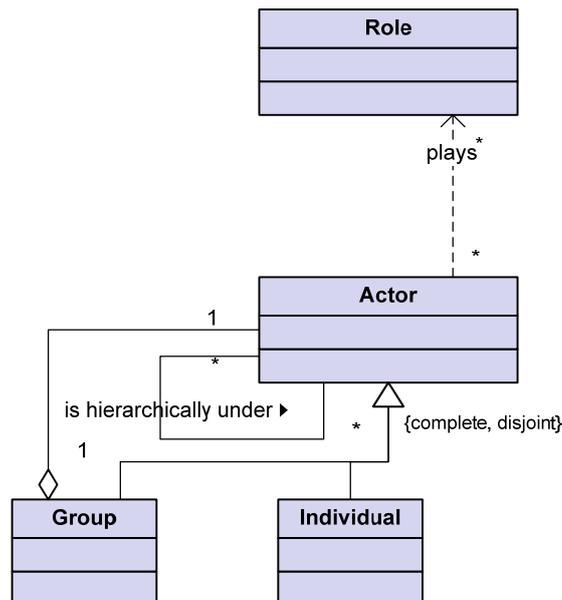


Figura 72. Metamodelo parcial de la *relación de agregación*

Relación de Jerarquía

Definición

Una *relación de jerarquía* es una *relación organizativa estructural* que identifica una dependencia de grado entre dos *actores*.

Definición Contextual

Actor

La *relación de jerarquía* muestra la dependencia de unos *actores* sobre otros, determinando así sus posiciones relativas de acuerdo a criterios de clase, poder, oficio, categoría, autoridad o cualquier otro asunto que conduzca a un sistema de clasificación.

Sinónimos

No.

Término Anglosajón

Hierarchy relationship.

Nota

La definición contextual relativa a actor está basada en la definición de jerarquía de la Wikipedia [Wikimedia, 2007].

Notación



Figura 73. Notación empleada para representar el concepto de *relación de jerarquía*

Metamodelo Parcial

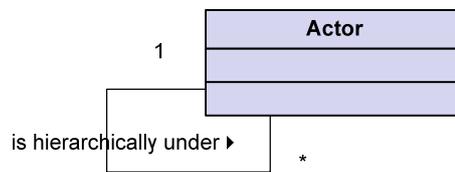


Figura 74. Metamodelo parcial de la *relación de jerarquía*

Relación Organizativa de Grupo

Definición

Una *relación organizativa de grupo* es una asociación entre *elementos organizativos* que describe las relaciones de interacción entre los *actores* del sistema.

Definición Contextual

No.

Sinónimos

Relación de grupo.

Término Anglosajón

Group relationship.

Nota

Se ha identificado como *relación de grupo* únicamente la *co-interacción*, pero el modelo está abierto y podría extenderse, si fuera necesario, para considerar otro tipo de *relaciones organizativas de grupo*.

Notación

No. Ver notaciones de cada una de las *relaciones organizativas de grupo*.

Metamodelo Parcial

No. Ver metamodelo parcial de cada una de las *relaciones organizativas de grupo*.

Co-interacción

Definición

La *co-interacción* es una *relación organizativa de grupo* entre dos *actores* que expresa una interacción entre ellos, encaminada al logro de un *objetivo* común que no podría alcanzarse sin dicha interacción.

Definición Contextual

Ver *Tarea*.

Ver *Objetivo*.

Tarea de Grupo, *Objetivo*

Dos *actores* del sistema podrían estar relacionados entre sí por medio de una *co-interacción*, indicando, de esta manera, que colaboran entre sí para llevar a cabo una *tarea de grupo* y alcanzar un *objetivo* común que no sería alcanzable de otra manera.

Tarea

La *co-interacción* refleja una colaboración existente entre dos *actores* del sistema. Para que esta interacción se dé es necesario que ambos *actores* realicen una *tarea, compuesta o atómica*. La composición de esas *tareas* dará lugar a una *tarea de grupo*. La composición de diferentes *co-interacciones*, entre varios *actores*, dará lugar a *tareas de grupo* más complejas.

Sinónimos

No.

Término Anglosajón

Co-interaction.

Nota

No es lo mismo una *co-interacción* que una *tarea cooperativa* tradicional. Lo que tradicionalmente se ha entendido como *tarea cooperativa* en esta ontología se corresponde con el concepto de *tarea de grupo*. Podría darse el caso que varias *co-interacciones* entre actores compusieran una *tarea de grupo*.

La *co-interacción* podría estar relacionada con una o varias de las caracterizaciones de las *tareas* según sus características CSCW: *comunicación, cooperación, colaboración o coordinación*. Son interacciones entre *actores* en un sistema. Tanto en español como en inglés, las características anteriores comienzan por "co". Haciendo alusión a esta coincidencia, se ha determinado llamar *co-interacción* a este tipo de relación.

Notación

Una *co-interacción* se representa por medio de una línea dirigida o no entre dos elementos organizativos. En los extremos aparece el nombre de la *tarea* realizada por el actor correspondiente para que se pueda llegar a dar esa *tarea de grupo*. En el centro de la línea aparece el nombre de la *tarea* junto con una leyenda que alude a la caracterización de la *tarea de grupo* según sus características CSCW: *coordinación, comunicación (communication), cooperación / colaboración*; de tiempo: *síncrona / asíncrona*; y de espacio: *mismo espacio (same) / espacios diferentes (different)*.

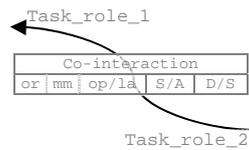


Figura 75. Notación empleada para representar el concepto de *co-interacción*

Metamodelo Parcial

El siguiente metamodelo parcial no representa la *co-interacción* en sí, sino los elementos que participan de esa relación. Para considerar esa *co-interacción*, dos actores del sistema deben realizar sendas *tareas*. Esas *tareas* las realizan porque desempeñan un *rol*. Probablemente la ejecución de esas *tareas* (y el consiguiente logro de los *objetivos* asociados) lleve a la consecución de un *objetivo* de una *tarea de grupo*.

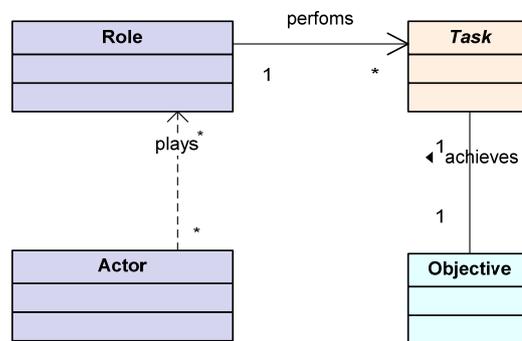


Figura 76. Metamodelo parcial de la *co-interacción*

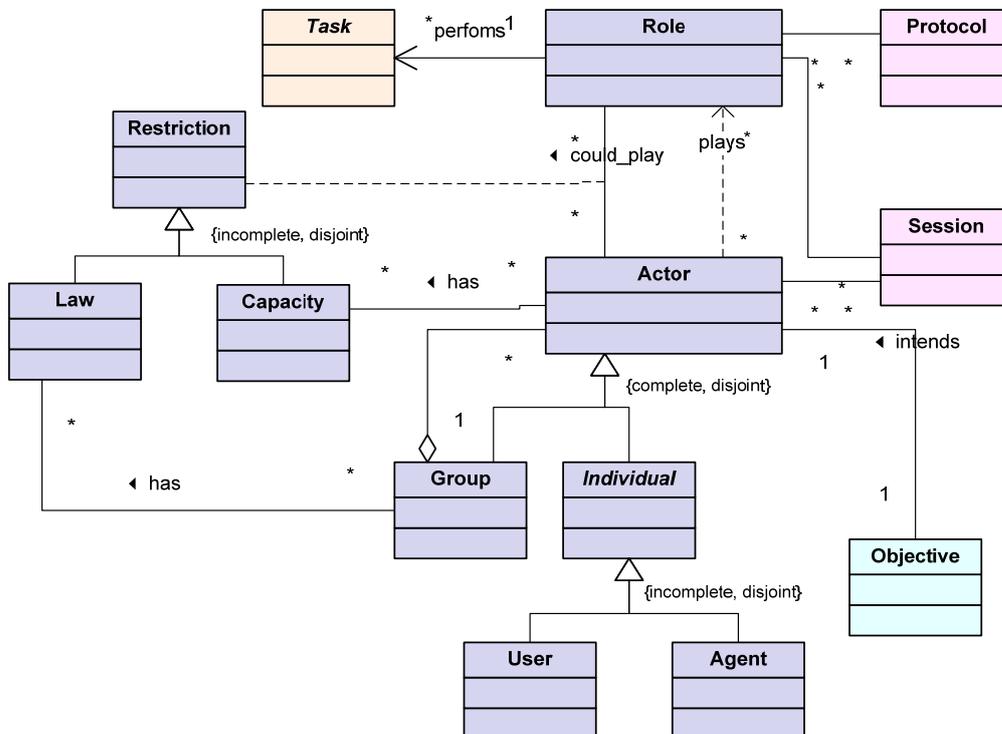


Figura 77. Metamodelo de Organización y sus relaciones con otros metamodelos

4.2.1.2 Metamodelo de Organización

La figura siguiente muestra el *metamodelo de organización*, incluyendo algunos elementos de otros metamodelos por su estrecha relación: *tarea*, *protocolo*, *sesión* y *objetivo*. En la descripción de cada uno de los conceptos se ha podido entender también su relación con los conceptos de otros metamodelos, por lo que no es necesario volver a comentarlo.

4.2.2 Modelo de Tareas

El desarrollo Web ha sufrido un espectacular cambio en los últimos años motivado por la mejora de la tecnología, las infraestructuras y el modo de crear aplicaciones, más centradas en las necesidades del usuario, que cada vez dan más importancia a la mejora de requisitos no funcionales como la usabilidad de los sistemas, para garantizar la calidad de los mismos.

Se ha pasado de tener portales Web formados por un conjunto de páginas escritas en HTML a tener verdaderos programas de un nivel de complejidad elevado, que permiten realizar todo tipo de acciones a través de Internet.

Todas esas acciones facilitan a los usuarios realizar tareas que por tiempo o por distancia espacial no serían posibles. La colaboración entre los distintos usuarios de la red es posible gracias al desarrollo de nuevas herramientas que, a través de la Web, facilitan la comunicación y la cooperación de las personas.

Para "orquestrar" todas esas tareas, para describir los pasos, las acciones que tienen que realizar los usuarios para lograr sus objetivos, se han utilizado los sistemas de análisis de tareas como. Estas notaciones proporcionan a los diseñadores la especificación exacta de estas tareas que se han de realizar, de manera que se facilita el desarrollo de la aplicación Web identificando los pasos a seguir en cada momento o los elementos de control, información, etc. [Lozano, 2002; Mori, 2004] que han de aparecer en la interfaz Web para poder llegar a ejecutar ciertas acciones.

El *Modelo de Tareas* describe las porciones de trabajo que se realizan los *actores* del sistema en la consecución de cada uno de los *objetivos* necesarios para alcanzar el *objetivo* de la aplicación. Para ello, el modelo conceptual se ha basado en los conceptos propuestos por autores relevantes en el análisis de tareas y de entornos colaborativos, de manera que cada tarea identificada y descrita para los diseñadores por medio de cualquier sistema de análisis de tareas, tenga en cuenta la caracterización tradicional de tareas más relevante.

Una buena caracterización de las tareas que se han de realizar en un portal Web facilitan la comprensión del dominio del problema a los diseñadores de la aplicación, lo que posibilita el desarrollo de una aplicación de calidad, en la que los pasos a seguir para cumplir los objetivos son más sencillos. El beneficiario final del buen desarrollo es el usuario final del sistema.

4.2.2.1 El Mundo de las Tareas

Como se ha comentado con anterioridad, el concepto de *tarea* es un término que se ha definido en multitud de ocasiones a lo largo de los años, pero siempre con pequeños matices que diferencian unas definiciones de otras. En este metamodelo, la definición de *tarea* está inspirada en una de las notaciones más utilizadas para el *modelado de tareas*: ConcurTaskTrees [Paternò, 1999].

Tarea

Definición

Una Tarea es una porción de trabajo necesaria para lograr un *Objetivo* determinado.

Definición Contextual

Objetivo

Tarea Compuesta, Tarea Atómica

Algunas *tareas* se descomponen en otras *tareas* más específicas, y éstas en otras, y así sucesivamente. Las últimas *tareas*, aquéllas que ya no puedan dividirse en otras, son *tareas atómicas*. Una *tarea* debe ser atómica o compuesta necesariamente. Adicionalmente, las *tareas* se caracterizan con otras características.

Abstracta, de Aplicación, de Interacción, de Usuario

Una *tarea* también se puede caracterizar como *abstracta*, de *aplicación*, de *interacción* o de *usuario*. Es una caracterización exclusiva, de manera que una *tarea* no podría ser de más de uno de esos tipos al mismo tiempo. Estos tipos de *tareas* son los que se proponen en CTT [Paternò, 1999]. Se ha considerado importante esta clasificación porque es ampliamente aceptada e identifica con claridad las tareas que se han de realizar en un sistema para alcanzar un objetivo.

Esta caracterización de *tareas* según las características previstas por paternò se ha dejado abierta por la ambigüedad que introduce la *tarea abstracta*, y porque no se descarta concretar otro tipo de *tarea* en este sentido. Es decir, estas características definen muy bien el tipo de *tarea* que se describe, pero no se descarta que pudiera surgir otro tipo, dentro de lo que se consideran *tareas abstractas* actualmente, que concretizara aún más dicha descripción.

Rol

Las *tareas* están asociadas a *roles*. Los *actores* que desempeñan esos *roles* son los que pueden llevar a cabo las *tareas*.

Sinónimos

Acción, Actividad.

Término Anglosajón

Task.

Nota

El concepto *tarea* en esta ontología tiene un significado muy general. Esta definición simplifica la concepción de qué es lo que tiene que realizar cada *actor* del sistema. Todas las porciones de trabajo se consideran *tareas*, desde la más compleja hasta

la más simple. Por tanto, otros términos como *acción* o *actividad* se consideran sinónimos en este caso.

Notación

No. Ver *tarea compuesta*, *tarea de grupo*, *abstracta*, *de aplicación*, *de interacción* y *de usuario*.

Metamodelo Parcial

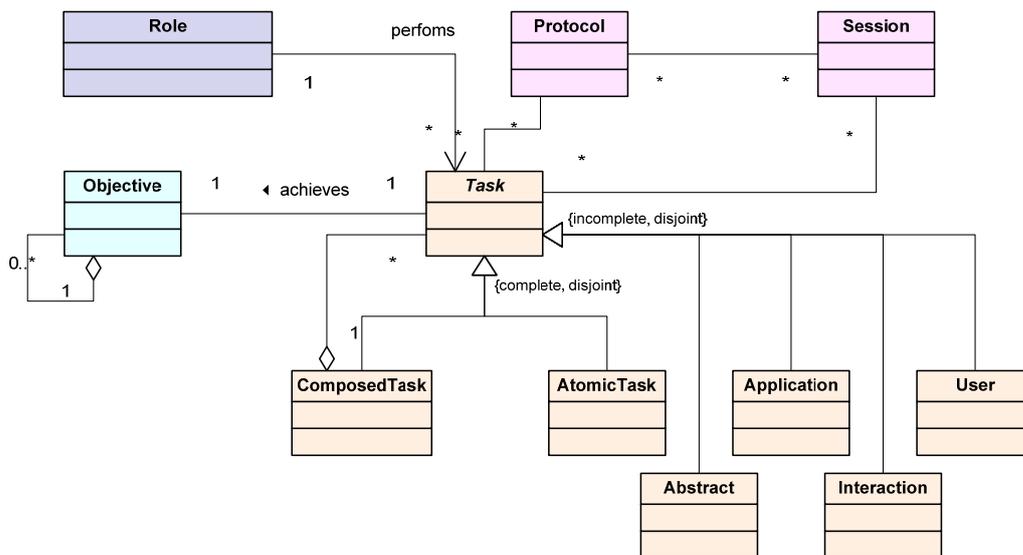


Figura 78. Metamodelo parcial de *tarea*

Tarea Abstracta

Definición

Las *tareas abstractas* son *tareas* complejas cuya descripción no encaja con alguno de los tres tipos previstos: de *aplicación*, de *interacción* o de *usuario*.

Definición Contextual

Ver *tarea*.

Sinónimos

No.

Término Anglosajón

Abstract task.

Nota

La notación proviene de [Paternò, 1999].

Notación



Figura 79. Notación empleada para representar el concepto de *tarea abstracta*

Metamodelo Parcial

Ver metamodelo parcial de *Tarea*.

Tarea de Aplicación

Definición

Una *tarea de aplicación* es aquella *tarea* que es ejecutada por el sistema, por *agentes*, en su totalidad, sin la intervención de un *usuario*.

Definición Contextual

Ver *tarea*.

Sinónimos

No.

Término Anglosajón

Application task.

Nota

Reciben información del sistema y pueden proporcionar información a los *actores*, pero son activadas por la aplicación.

La notación proviene de [Paternò, 1999]

Notación



Figura 80. Notación empleada para representar el concepto de *tarea de aplicación*

Metamodelo Parcial

Ver metamodelo parcial de *tarea*.

Tarea de Interacción

Definición

Una *tarea de interacción* es aquella *tarea* llevada a cabo por el *usuario* interactuando directamente con la aplicación.

Definición Contextual

Ver *tarea*.

Sinónimos

No.

Término Anglosajón

Interaction task.

Nota

Se trata de *tareas* activadas directamente por el *usuario*.

La notación proviene de [Paternò, 1999]

Notación



Figura 81. Notación empleada para representar el concepto de *tarea de interacción*

Metamodelo Parcial

Ver metamodelo parcial de *tarea*.

Tarea de Usuario

Definición

Una *tarea de usuario* es aquella *tarea* que lleva a cabo un *actor individuo* humano en su totalidad.

Definición Contextual

Ver *tarea*.

Sinónimos

No.

Término Anglosajón

User task.

Nota

Pueden ser actividades cognitivas o físicas en las que no se interactúa con el sistema.

La notación proviene de [Paternò, 1999]

Notación



Figura 82. Notación empleada para representar el concepto de *tarea de usuario*

Metamodelo Parcial

Ver metamodelo parcial de *tarea*.

4.2.2.2 Clasificación de Tareas según sean Atómicas o Compuestas

Tal y como se ha expresado con anterioridad, se considera un único término, *tarea*, para expresar las porciones de trabajo que realizan los *actores* del sistema. En otros trabajos se ha hecho uso de términos como "acción" o "actividad" para describir, componer o descomponer tareas.

El uso de un único término facilita al Ingeniero del Software la especificación del sistema. No necesita pensar cuál es el límite del nivel de granularidad a partir del cual se pasaría de hablar de acción, tarea o actividad.

En cualquier caso se habla de *tarea*, si bien se acepta que existen diferentes niveles de granularidad. Por este motivo, el concepto de *tarea* se "adorna" con el de *compuesta* o *atómica*, según sea el caso, naciendo de esta manera dos nuevos conceptos que complementan y caracterizan al primero.

Tarea Compuesta

Definición

Una *tarea compuesta* es una *tarea* formada por varias *tareas atómicas* u otras *tareas compuestas*.

Definición Contextual

Ver *tarea*.

Ver *tarea de grupo*.

Sinónimos

No.

Término Anglosajón

Composite task.

Nota

Conviene tener presente que una *tarea compuesta*, como *tarea* que es, se caracteriza también como *tarea abstracta*, de *aplicación*, de *interacción* o de *usuario*.

Notación

No existe una notación específica puesto que en los árboles CTT se puede observar claramente si se trata de una *tarea compuesta*, ya que sería un nodo del que salen más *tareas* en caso de estar desplegado, o un nodo con un símbolo '+' en caso de estar replegado.

Metamodelo Parcial

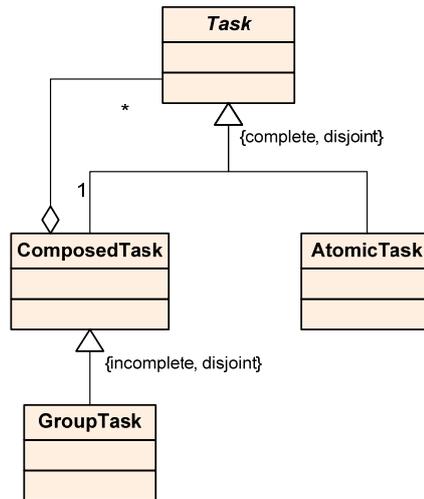


Figura 83. Metamodelo parcial de *tarea compuesta*

Tarea Atómica

Definición

Una *tarea atómica* es una *tarea* que no se puede descomponer en otras, sino que su ejecución es la solución en sí misma.

Definición Contextual

Ver *tarea*.

Sinónimos

No.

Término Anglosajón

Atomic task.

Nota

Las *tareas atómicas* son el nivel de granularidad más fino, son indivisibles y la solución en sí mismas.

Conviene tener presente que una *tarea atómica*, como *tarea* que es, se caracteriza también como *tarea abstracta*, de *aplicación*, de *interacción* o de *usuario*.

Notación

No.

Metamodelo Parcial

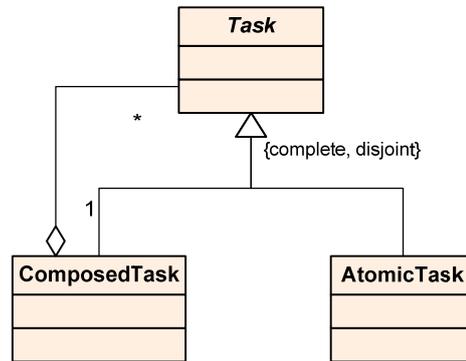


Figura 84. Metamodelo parcial de *Tarea Atómica*

4.2.2.3 Las Tareas de Grupo: una Especialización Frecuente de Tareas Compuestas en Sistemas Web Multi-Usuario

En el análisis de tareas para sistemas Web, cada vez resulta de mayor importancia tener en cuenta las tareas cooperativas que se puedan dar en un sistema. Esto se debe, principalmente, a que la red se ha convertido en el medio más importante desde el que las personas pueden realizar acciones en conjunto. Se pueden comunicar en tiempo real por medio de chats o sistemas de telefonía VoIP, o de forma asíncrona por medio del tradicional correo electrónico o de los foros. Se establecen nuevos medios de colaboración y comunicación como los blogs. Todos los miembros de una organización, aun distribuidos geográficamente se pueden coordinar por medio de agendas y calendarios compartidos que operan a través de la Web, etc.

Todas estas posibilidades sugieren un grupo especial de taras que, tradicionalmente, se han llamado cooperativas. En esta caracterización hemos preferido llamarlas tareas de grupo, puesto que se hace referencia a tareas CSCW [Greif, 1988], Computer-Supported Cooperative Work, cuyas bases son la coordinación, la comunicación y la información compartida [Poltrack, 1999]. Antes de esta clasificación, la categoría compartición de la información era denominada cooperación [Poltrack, 1994], sin embargo, carece de sentido tratar de explicar que las bases de un concepto (CSCW) son de esa manera (coordinación, comunicación y *cooperación*) puesto que esta misma palabra se encuentra en CSCW, el concepto a definir.

Por otro lado, el uso del término cooperativo, a veces, se ha confundido con el término colaborativo. Es cierto que son palabras, conceptos muy próximos, pero esconden una pequeña diferencia semántica que merece la pena destacar.

Por todo ello, en este modelo conceptual se utiliza un término más genérico cuando se habla de *tareas* en las que participan varios usuarios para lograr un *objetivo* que sería inalcanzable de forma individual. Este término es *tarea de grupo*, que, lógicamente, es una *tarea compuesta*, que se descompone en diferentes *tareas* realizadas por distintos *actores* del sistema.

Tarea de Grupo

Definición

Una *tarea de grupo* es una *tarea compuesta* en la que participan dos o más *actores* del sistema.

Definición Contextual

Ver *tarea compuesta*.

Características CSCW

Una *tarea de grupo* se puede caracterizar atendiendo a los pilares del CSCW [Greif, 1988], Computer-Supported Cooperative Work: *Coordinación*, la *Comunicación* y la *Información Compartida* [Pollock, 1999].

Una *tarea de grupo* será de al menos uno de estos tipos, pudiendo ser caracterizada como de varios de ellos a la vez.

Características de Johansen

Una *tarea de grupo* se puede caracterizar atendiendo a matriz espacio-temporal establecida por Johansen [Johansen, 1988]: *síncrona*, *asíncrona*, en el *mismo-lugar* o en *lugares-diferentes*.

Una *tarea de grupo* será *síncrona* o *asíncrona*, y se llevará a cabo en el *mismo-lugar* o en *lugares-diferentes*. Es decir, se clasificará espacial y temporalmente, pero no podría ser *síncrona* y *asíncrona* al mismo tiempo, ni se puede llevar a cabo en el *mismo-lugar* y en *lugares-diferentes* a la vez.

Sinónimos

No.

Término Anglosajón

Group task.

Nota

No.

Notación

Las *tareas de grupo* se representan por medio de co-interacciones.

Metamodelo Parcial

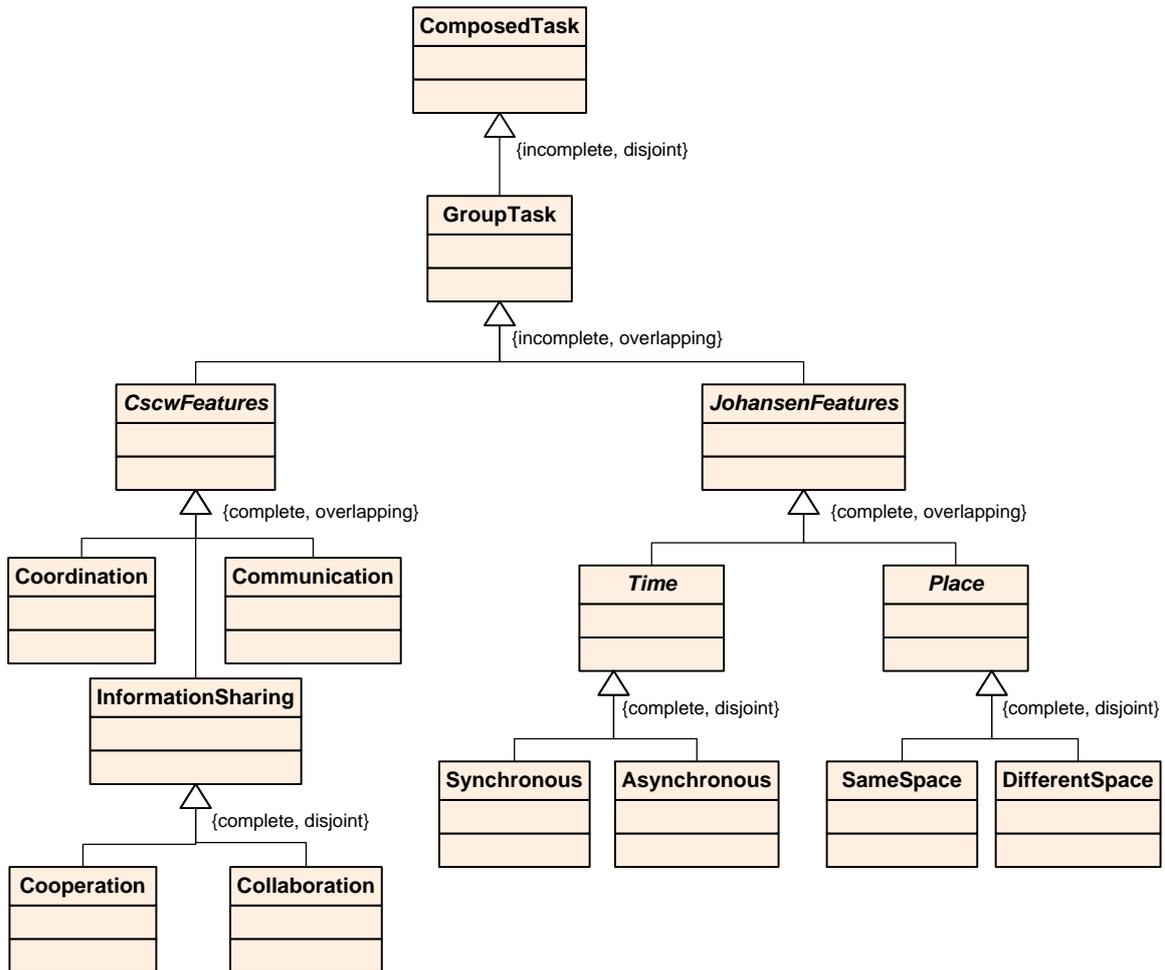


Figura 85. Metamodelo parcial de la *tarea de grupo*

4.2.2.4 CSCW Features

Tras esta discusión acerca de las *tareas de grupo*, han surgido una serie de conceptos típicos del CSCW que nos permiten caracterizar las tareas según sean orientadas a la *coordinación*, a la *comunicación* o a la *compartición de la información* [Poltrock, 1999].

Muchos investigadores consideran *cooperación* y *colaboración* como conceptos diferentes [Flores, 1994; Grudin, 1994] y así se ha hecho en este modelo conceptual, a pesar de que la diferencia sea mínima.

En el capítulo anterior se han discutido ampliamente estos conceptos.

Coordinación

Definición

Una *tarea de grupo* se caracteriza como de *coordinación* cuando dos o más *actores* del sistema armonizan esfuerzos para lograr un *objetivo* común.

Definición Contextual

No.

Sinónimos

No.

Término Anglosajón

Coordination.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

Compartición de la Información

Definición

En este caso, se habla de *compartición* de la información cuando dos o más *actores* llevan a cabo *tareas de grupo cooperativas* o *colaborativas*, puesto que en el dicho proceso de grupo se produce un acceso común a recursos del sistema sobre los que se colabora o se coopera.

Definición Contextual

Ver *Colaboración*.

Ver *Cooperación*.

Características CSCW, Colaboración, Cooperación

Las características típicas del CSCW son tres: (1) *coordinación*, (2) *comunicación* y (3) *compartición de la información, colaboración o cooperación* [Poltrack, 1997; Poltrack, 1999].

a) Puesto que en la mayoría de las investigaciones del campo se hace diferenciación entre *cooperación* y *colaboración*, por pequeña que sea, se han mantenido ambos términos.

b) Puesto que muchos investigadores piensan que el tercer punto es la *cooperación* y, otros tantos, que es la *compartición de la información*, se ha decidido contemplar los dos términos.

De los dos puntos anteriores a) y b) se concluye contemplar los tres términos detectados en la explicación tradicional del tercer "pilar" del CSCW (3). tanto la *cooperación* como la *colaboración* son *compartición de la información* de alguna manera.

Normalmente no se empleará este concepto puesto que se habla con mayor frecuencia de *colaboración* y/o de *cooperación* que de *compartición de la información*.

Características CSCW

Una *tarea de grupo* puede caracterizarse como de *coordinación* y/o de *comunicación* y/o de *cooperación* o de *colaboración*. Es decir, una tarea de grupo podría ser a la vez caracterizada por cualquiera de las anteriores de forma concurrente, pero nunca podrían concurrir *cooperación* y *colaboración*: una *tarea de grupo* no puede caracterizarse como de *colaboración* y de *cooperación* al mismo tiempo.

Sinónimos

No.

Término Anglosajón

Information sharing.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

Cooperación

Definición

Una *tarea de grupo* se caracteriza como de *cooperación* cuando dos o más *actores* llevan a cabo otras *tareas* de menor nivel de granularidad de forma conjunta y concurrente para lograr un *objetivo* común.

Definición Contextual

Características CSCW

una *tarea de grupo* puede caracterizarse como de *coordinación* y/o de *comunicación* y/o de *cooperación* o de *colaboración*. Es decir, una tarea de grupo podría ser a la vez caracterizada por cualquiera de las anteriores de forma concurrente, pero nunca podrían concurrir *cooperación* y *colaboración*: una *tarea de grupo* no puede caracterizarse como de *colaboración* y de *cooperación* al mismo tiempo.

Ver *Colaboración*.

Sinónimos

No. Sin embargo, en ocasiones se utilizan los términos *cooperación* y *colaboración* como sinónimos. En esta ontología se consideran como conceptos diferentes.

Término Anglosajón

Cooperation.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

Colaboración

Definición

Una *tarea de grupo* se caracteriza como de *cooperación* cuando dos o más *actores* llevan a cabo otras *tareas* de menor nivel de granularidad de forma conjunta, pero no concurrente, para lograr un *objetivo* común.

Definición Contextual

Características CSCW

una *tarea de grupo* puede caracterizarse como de *coordinación* y/o de *comunicación* y/o de *cooperación* o de *colaboración*. Es decir, una *tarea de grupo* podría ser a la vez caracterizada por cualquiera de las anteriores de forma concurrente, pero nunca podrían concurrir *cooperación* y *colaboración*: una *tarea de grupo* no puede caracterizarse como de *colaboración* y de *cooperación* al mismo tiempo.

Ver *Cooperación*.

Sinónimos

No. Sin embargo, en ocasiones se utilizan los términos *cooperación* y *colaboración* como sinónimos. En esta ontología se consideran como conceptos diferentes.

Término Anglosajón

Colaboration.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

Comunicación

Definición

Una *tarea de grupo* se caracteriza como de *comunicación* cuando dos o más *actores* llevan a cabo otras *tareas* de menor nivel de granularidad de manera que dichos actores envían, reciben o intercambian información.

Definición Contextual

Características CSCW

Una *tarea de grupo* puede caracterizarse como de *coordinación* y/o de *comunicación* y/o de *cooperación* o de *colaboración*. Es decir, una *tarea de grupo* podría ser a la vez caracterizada por cualquiera de las anteriores de forma concurrente, pero nunca podrían concurrir *cooperación* y *colaboración*: una *tarea de grupo* no puede caracterizarse como de *colaboración* y de *cooperación* al mismo tiempo.

Sinónimos

No.

Término Anglosajón

Communication.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

4.2.2.5 Johansen Features

Adicionalmente, una *tarea de grupo* se podría caracterizar según sus características espacio-temporales que definirán el modo en el que los usuarios que participan de ella la realizan de acuerdo al dónde y al cuándo.

Johansen [Johansen, 1988] estableció una matriz espacio-temporal que aclara estos conceptos y que expresa que una *tarea* puede ser llevada a cabo en el mismo lugar o en sitios diferentes, así como de forma síncrona (tiempo real) o asíncrona. Estos conceptos, así como dicha matriz, han sido presentados previamente en capítulos anteriores.

Recordemos que las aplicaciones groupware, por sus características espacio-temporales, se pueden clasificar según sean: *síncronas / en el mismo lugar*, *síncronas / en lugares diferentes*, *asíncronas / en el mismo lugar* o *asíncronas / en lugares diferentes*.

Síncrono

Definición

El término *síncrono/-a* indica que una *tarea de grupo* se realiza en tiempo real, al mismo tiempo, por parte de los *actores* que participan de ella.

Definición Contextual

Asíncrona Una *tarea de grupo* no puede ser al mismo tiempo *síncrona* y *asíncrona*.

Lugar (Mismo Lugar / Lugares Diferentes)

Una *tarea de grupo* puede ser a la vez *síncrona* y ejecutarse en el mismo lugar o bien ser *síncrona* y ejecutarse en *lugares diferentes*.

Sinónimos

No.

Término Anglosajón

Synchronous.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

Asíncrono

Definición

El término *asíncrono/-a* indica que una *tarea de grupo* no se realiza en tiempo real. Los *actores* que participan de ella llevan a cabo sus *tareas* en tiempos diferentes.

Definición Contextual

Síncrono Una *tarea de grupo* no puede ser al mismo tiempo *síncrona* y *asíncrona*.

Lugar (Mismo Lugar / Lugares Diferentes)

Una *tarea de grupo* puede ser a la vez *asíncrona* y ejecutarse en el mismo lugar o bien ser *síncrona* y ejecutarse en *lugares diferentes*.

Sinónimos

No.

Término Anglosajón

Asynchronous.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

Mismo Lugar

Definición

El término *mismo lugar* indica que los *actores* que participan en una *tarea de grupo* están ubicados geográficamente en el mismo sitio.

Definición Contextual

Lugares Diferentes Una *tarea de grupo* no puede llevarse a cabo en el *mismo lugar* y en *lugares diferentes* al mismo tiempo.

Tiempo (Síncrono / Asíncrono)

Los *actores* que llevan a cabo una *tarea de grupo* en un *mismo lugar* lo pueden hacer de forma *síncrona* o bien de forma *asíncrona*.

Sinónimos

No.

Término Anglosajón

Same space.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

Lugares diferentes

Definición

El término *lugares diferentes* indica que los *actores* que participan en una *tarea de grupo* están ubicados geográficamente en sitios distintos.

Definición Contextual

Mismo Lugar Una *tarea de grupo* no puede llevarse a cabo en el *mismo lugar* y en *lugares diferentes* al mismo tiempo.

Tiempo (Síncrono / Asíncrono)

Los *actores* que llevan a cabo una *tarea de grupo* en lugares diferentes lo pueden hacer de forma *síncrona* o bien de forma *asíncrona*.

Sinónimos

No.

Término Anglosajón

Different space.

Nota

No.

Notación

Ver *tarea de grupo*.

Metamodelo Parcial

Ver metamodelo parcial de *tarea de grupo*.

4.2.3 Modelo de Objetivos

Para la representación de los objetivos identificados tras la etapa de elicitación de requisitos se pueden plantear diferentes posibilidades.

Desde la perspectiva de la sociología se plantean tres términos diferentes en torno a esta idea:

- El *fin* o *finalidad* como propósito último, como el logro más general que se pretende alcanzar.
- La *meta* y el *objetivo*. Son los logros que se pretenden alcanzar. No son dos terminos diferentes, sino que la meta es algo más concreto, más específico. Se dice que la meta cuantifica y operacionaliza el objetivo.
- Se emplea el término *producto* para referirse al nivel más básico de resultados, obtenidos a partir de la realización de ciertas actividades.

Tradicionalmente, para resolver problemas de gran envergadura se ha recurrido a la técnica del "divide y vencerás" para obtener sub-problemas de menor tamaño que se puedan resolver separadamente de un modo más sencillo.

Hay diferentes posibilidades a la hora de realizar este modelo. A continuación se discuten las que se han estudiado hasta seleccionar la definitiva.

4.2.3.1 Posibilidad 1

En un principio se podría entonces optar por resolver el problema colaborativo planteado dividiéndolo en tres niveles de granularidad: *fin*, *meta* y *objetivo*, de mayor a menor nivel respectivamente. Con algunas diferencias terminológicas con el campo de la sociología. Concretamente no se emplea el término *producto* para evitar confundirlo con el resultado obtenido tras la etapa de desarrollo en Ingeniería del Software.

El *fin* representaría el mayor nivel de granularidad. Hace referencia al problema a resolver, es decir, el mayor y más general de los problemas que resuelve la aplicación que se pretende diseñar. La *estructura organizativa* de los *actores* del sistema se ha organizado para resolver este *fin*.

Previamente se ha definido el concepto de *grupo* como un conjunto de *actores* que necesitan interactuar y colaborar para alcanzar un propósito (*objetivo*) común. Dichos propósitos (*objetivos*) comunes no serían realizables sin tal colaboración

Ese nuevo concepto, *objetivo*, sería el siguiente nivel de granularidad en la descomposición de problemas. Cada *grupo* resuelve un *objetivo*, *objetivo* que es parte del *fin* general del sistema. Es un sub-problema en el que se ha dividido el problema principal. Por tanto, un *grupo* (los *actores* que desempeñan los *roles* que pertenecen a este *grupo*) lleva a cabo las *tareas* necesarias para lograr un *objetivo*.

A su vez, cada uno de esos *objetivos* se podría descomponer en *metas*, cuyo nivel de granularidad es el menor. Una *meta* es un pequeño sub-problema que es solucionado por una *tarea*. Sería también posible estar hablando de problemas atómicos tan sencillos como pulsar un botón.

Por consiguiente, un *actor* podría lograr diferentes *metas* realizando diferentes *tareas*. *tareas* que tales *actores* sólo podrían ejecutar si desempeñan los *roles* adecuados. La colaboración entre los *actores* en los *grupos* propicia la consecución de los *objetivos*. La consecución de todos los *objetivos* hace factible la consecución del *fin* para el que la aplicación colaborativa se construye.

Lógicamente, un *actor* podría participar en la consecución de diferentes *metas* y *objetivos* porque podría desempeñar *roles* que pertenecieran a *grupos* diferentes.

De esta manera, el metamodelo parcial quedaría como se muestra en la Figura 86.

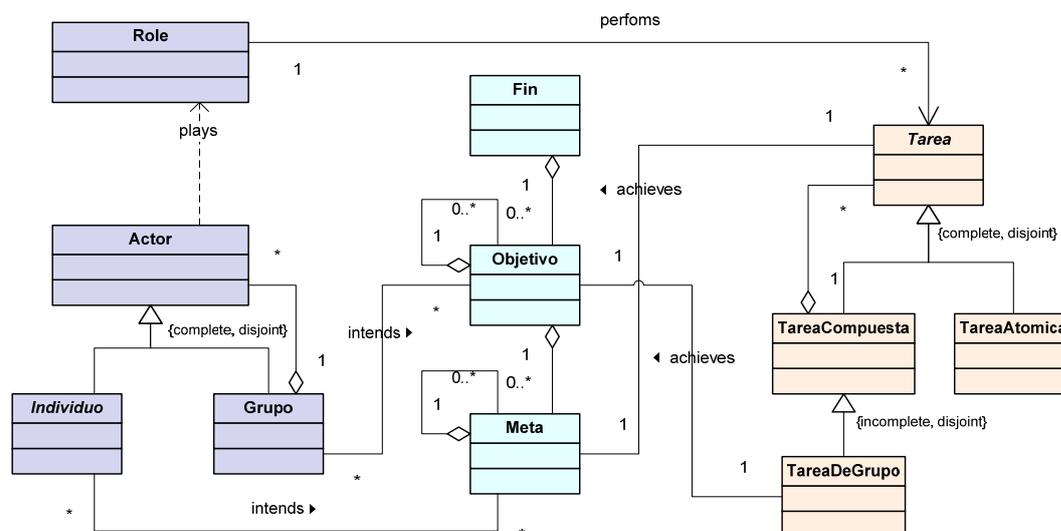


Figura 86. Posible metamodelo parcial del *Modelo de Objetivos* del sistema: primera opción

En el metamodelo parcial se puede observar la división del *fin* en *objetivos*, la división de *objetivos* en otros *objetivos* y/o en *metas* y la división de *metas* en otras *metas*, que serían finalmente propósitos atómicos.

Las *metas* son los propósitos de menor nivel de granularidad y se puede apreciar en el metamodelo su relación con los *actores individuos* que intentarían abordarlas (relación *intends*). Así mismo, estas *metas* serían llevadas a cabo por medio de las *tareas compuestas* o *atómicas* correspondientes (relación *achieves*).

Por otro lado, los *objetivos* son abordados por los *grupos* de *actores* (relación *intends*) que realizarían las *tareas de grupo* correspondientes para la consecución de dichos *objetivos* (relación *achieves*).

La consecución de las *metas* y de los *objetivos* lleva a lograr el *fin* para el que se ha diseñado el sistema.

4.2.3.2 Posibilidad 2

Sin embargo, con el propósito de simplificar el metamodelo y el diseño final del sistema se ha optado por considerar la meta y el objetivo como semánticamente idénticos. Es decir, como propósitos de un cierto nivel de granularidad que pueden estar compuestos por otros objetivos y así sucesivamente hasta llevar a objetivos atómicos que no puedan ser descompuestos.

Esta decisión viene provocada de forma similar a aquélla considerada en la definición de *tareas*, donde no se ha basado dicho concepto en otros como acción o actividad con el mismo objetivo de la simplicidad.

En la especificación de las *tareas* sí se ha tenido en cuenta explícitamente los conceptos de *tarea compuesta* y *tarea atómica*. Algo que en la definición de *objetivos* no se tendrá en cuenta de forma explícita. Esto se debe a que se ha considerado hacer así en las *tareas* para una especificación más completa que es necesaria para el diseño del sistema. Ese nivel de especificación no será necesario en los *objetivos*.

Dicho esto, el metamodelo parcial que expresaría los objetivos de un sistema sería tal y como se muestra en la Figura 87.

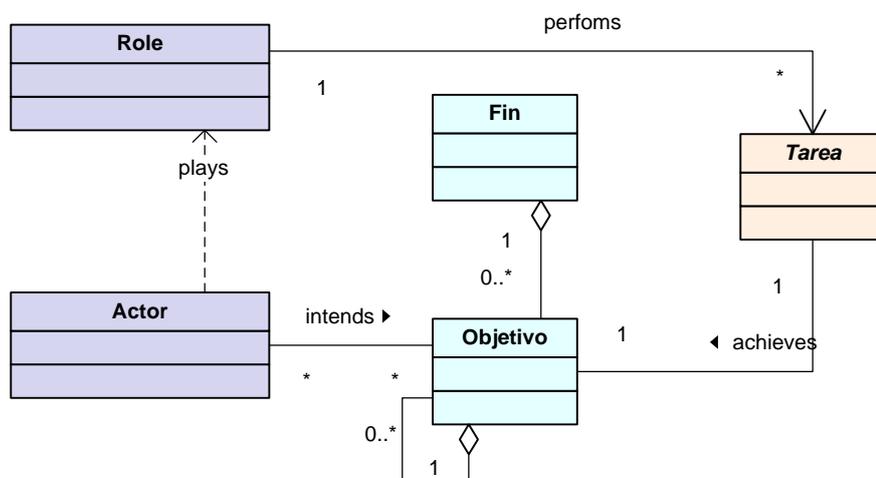


Figura 87. Posible metamodelo parcial del *Modelo de Objetivos* del sistema: segunda opción

Los *actores* abordan los objetivos del sistema (relación *intends*) ejecutando las *tareas* que permiten lograr dichos objetivos (relación *achieves*). En este caso no es necesario especificar que los actores Individuos o los Grupos abordan un tipo específico de objetivo. Todo son objetivos. Lo mismo sucede con las tareas.

La consecución de los *objetivos* lleva a lograr el *fin* para el que se ha diseñado el sistema.

4.2.3.3 Posibilidad Seleccionada

En los subapartados anteriores se ha afirmado lo siguiente:

- La consecución de los *objetivos* lleva a lograr el *fin* para el que se ha diseñado el sistema.
- El *fin* representa el mayor nivel de granularidad. Hace referencia al problema a resolver, es decir, el mayor y más general de los problemas que resuelve la aplicación que se pretende diseñar.

Teniendo en cuenta estas afirmaciones se podría considerar el *fin* también como un *objetivo*, el mayor de todos los *objetivos*.

De esta manera se simplifica el *Modelo de Objetivos* hasta el punto de considerar un único concepto que engloba los tres mencionados hasta ahora. Se trata pues del concepto de *objetivo*, que puede tener diferentes niveles de granularidad y puede estar compuesto por otros objetivos.

Abarca desde el mayor de todos los propósitos, es decir, aquél para el que se ha diseñado el sistema y que se ha venido llamando fin en las posibilidades consideradas con anterioridad; hasta el menor de todos los propósitos, aquellos que son atómicos por no poder dividirse en otros más sencillos.

Por otro lado, desde la Ingeniería de Requisitos también se definen los conceptos *requisito* y *objetivo*. Por los motivos que se discuten en el apartado 5.2.6.1, en esta ontología se consideran sinónimos, pero con alguna salvedad.

El metamodelo parcial del *Modelo de Objetivos* que se ha adoptado es el que se muestra en la Figura 88.

Objetivo

Definición

Un *objetivo* es un propósito que aborda un *actor* del sistema por medio de la ejecución de la *tarea* que logra la consecución de dicho *objetivo*.

Definición Contextual

Objetivo Los *objetivos* pueden descomponerse en otros objetivos de un nivel de granularidad menor, y así sucesivamente hasta que no se puedan descomponer más porque se traten de propósitos atómicos. El *objetivo* de mayor nivel de granularidad es el que resuelve el problema para el que se ha diseñado el sistema.

Actor Los *actores* abordan los objetivos identificados en el sistema, es decir, tratan de lograrlo.

Tarea Las *tareas* son el medio por el cual se pueden lograr los *objetivos* en el sistema. Las *tareas* se diseñan para lograr *objetivos*.

Sinónimos

Requisito.

Término Anglosajón

Objective.

Nota

Existe una salvedad a la sinonimia entre los términos de *requisito* y *objetivo*. En la etapa de *elicitación de requisitos* se empleará el término *objetivo*, y no el de *requisito*, para hablar de los *objetivos del sistema*. Este es el único caso en el que no se puede emplear el término *requisito*. Si se habla de este tipo de objetivos, de alto nivel, de alto nivel de granularidad, de alto nivel de abstracción, objetivos más cercanos al vocabulario y necesidades del cliente que al software y al vocabulario empleado por los desarrolladores, entonces, no se puede hablar de *requisitos* sino de *objetivos*. Normalmente se acompañará con "coletilla" "*del sistema*" para aclarar que no se trata de requisitos, sino de los *objetivos del sistema*.

En el resto de etapas, a pesar de que siempre se hace referencia a los *requisitos*, se emplea el término *objetivo*. Esto se debe a que por la propia definición de *tarea* y por su uso extendido se ha considerado oportuno hacerlo así.

Por lo tanto, sólo se emplea el término *requisito* en la primera etapa de la metodología (*Elicitación de Requisitos*), salvo que se haga referencia a los *objetivos del sistema*.

Notación

No.

Metamodelo Parcial

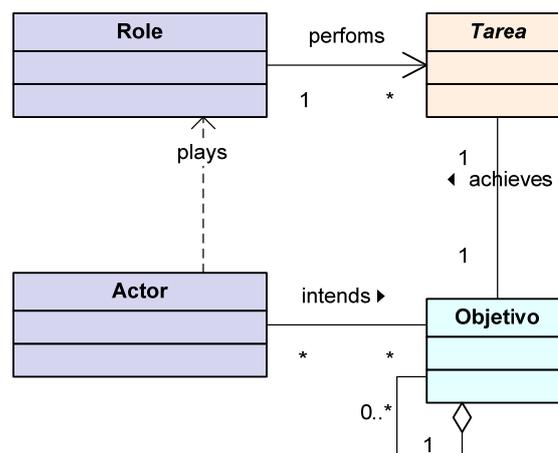


Figura 88. Metamodelo parcial del *Modelo de Objetivos* del sistema

4.2.4 Modelo de Sesión

Según [Ellis, 1991], uno de los investigadores del área más relevantes, una sesión es un periodo de interacción síncrona en un sistema groupware. Para este trabajo hemos adoptado el modelo de sesión propuesto en AMENITIES [Garrido, 2003] por varios motivos. En primer lugar porque encaja en la estructura y el metamodelo propuestos en este trabajo. En segundo lugar, porque esta tesis no está centrada en el concepto de sesión, por lo que no se ha profundizado en su estudio. Por ello, en este apartado simplemente se muestra dicho modelo y la relación del metamodelo con conceptos de otros metamodelos: *rol*, *actor* y *tarea*.

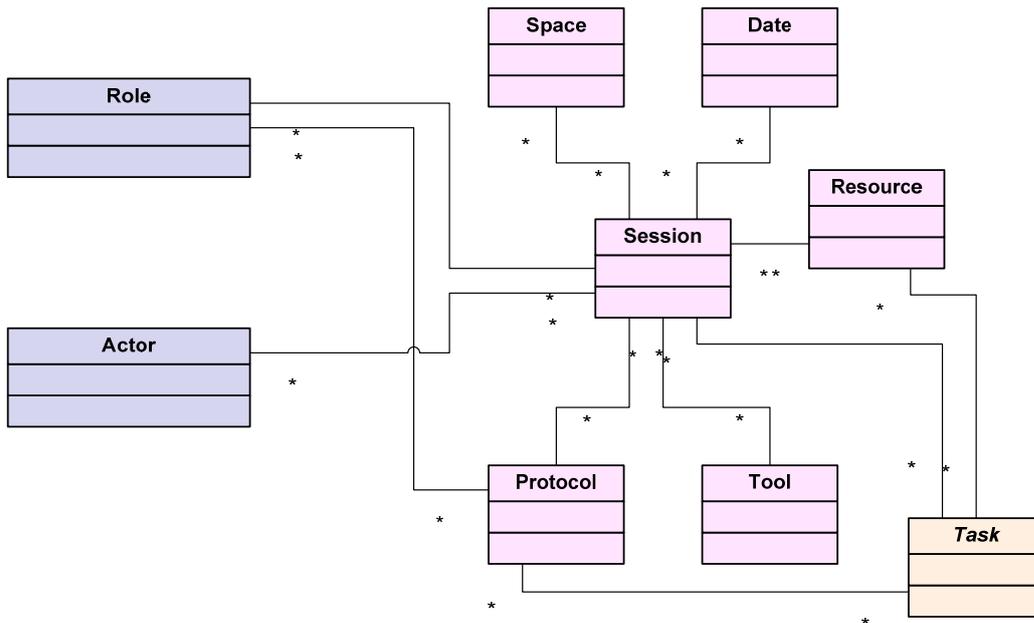


Figura 89. Metamodelo parcial de Sesión

CAPÍTULO 5.

**TOUCHE: TASK-ORIENTED AND USER-
CENTRED PROCESS MODEL FOR
DEVELOPING INTERFACES FOR HUMAN-
COMPUTER-HUMAN ENVIRONMENTS**

En este capítulo se describen las etapas del modelo de proceso propuesto en esta tesis, así como la metodología seguida en cada una de ellas. A este punto se ha llegado tras realizar el estudio del estado del arte general de los sistemas CSCW, el estudio sobre los conceptos más concretos, la implementación de algunas herramientas (Anexo II) que han permitido profundizar en el estudio del groupware y tras establecer las bases por medio de un vocabulario específico descrito en la ontología y modelo conceptual presentados en el capítulo anterior.

Es en el siguiente capítulo donde se muestra un ejemplo de funcionamiento del modelo de proceso, por lo que en éste sólo se describe de forma más teórica.

5.1 Descripción general

El número de técnicas a emplear en las etapas del modelo de proceso puede ser muy numeroso. No se descarta la posibilidad de emplear cualquier otra técnica, no contemplada en esta metodología, que pudiera complementar la información de la especificación, pero esa posibilidad, ese gran número de técnicas, podría llegar a complicar en exceso la descripción del sistema.

Si un analista se encuentra con una serie de pasos y técnicas a seguir, que le resulten más o menos viables por sencillez y tiempo, es posible que la especificación que haga sea más exitosa que si encuentra un número excesivo de posibilidades que le hagan "perderser en la abundancia".

Por este motivo se concretan las técnicas a emplear en la metodología, sugiriendo aquéllas que se consideran oportunas de entre las que son ampliamente utilizadas y aceptadas. En otras ocasiones, se extienden, se modifican o se crean nuevas técnicas que permitan mejorar y completar la especificación, normalmente aportando una visión colaborativa.

El modelo de proceso desarrollado para el estudio, análisis y diseño de entornos colaborativos consta de cuatro pasos interrelacionados y dependientes. Por un lado son independientes porque en cada etapa se abordan asuntos concretos y específicos correspondientes a dicha etapa y no a otra.

Sin embargo, el análisis y diseño de los sistemas informáticos en general, no solo de entornos colaborativos, hace imposible una independencia absoluta entre etapas. Existe pues una relación entre ellos de manera que se pueda pasar de una etapa a otra o de manera que un cambio en una etapa puede tener una repercusión en otras.

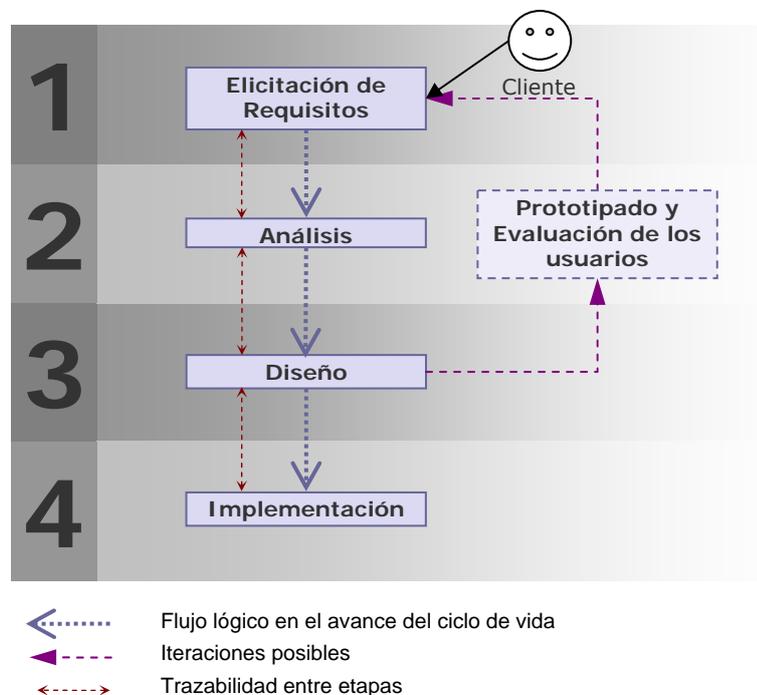


Figura 90. Etapas de la Metodología

Por lo tanto la trazabilidad entre las etapas identificadas es tan importante como cada una de las etapas para mantener la coherencia del modelo y de la especificación del sistema.

El proceso de desarrollo (Figura 90) que se sigue es el definido tradicionalmente en Ingeniería del Software, que incluye las etapas de:

- Elicitación de Requisitos
- Análisis
- Diseño
- Implementación

Este modelo de proceso incluye las prácticas y estrategias que se consideran necesarias para mejorar el proceso de desarrollo. Como prácticamente todos los modelos empleados en Ingeniería del Software también éste es iterativo, lo que permite refinar cada uno de los pasos hasta obtener el producto final deseado.

Tras el diseño se considera la realización de un prototipo que permita la evaluación de los usuarios. Esto permite que sean ellos los que validen cómo va la aplicación, dándoles mayor participación y mayor valor a su opinión.

5.2 Etapa 1.- Elicitación de Requisitos

"We need to model what we learn from our users, to confirm with them and with our clients our understanding of the work to be supported and to incorporate that understanding into the software we build" [Constantine, 1999]

La primera etapa del modelo de proceso consiste en la elicitación de los requisitos del sistema que se quiere implementar para resolver los objetivos planteados por los clientes.

Se trata de un proceso complejo puesto que es el punto en el que una persona, probablemente ajena al mundo de la computación, trata de expresar sus necesidades en lenguaje natural. Estas necesidades se han de plasmar de una manera adecuada para conseguir una especificación total del sistema que se aproxime al máximo a lo que el cliente quiere. Fruto de esa especificación surgirá la aplicación que resuelva sus necesidades.

La Ingeniería de Requisitos trata de resolver esta problemática.

Para esta etapa nos hemos basado en la experiencia de una tesis doctoral centrada precisamente en la elicitación de requisitos (así como su análisis y validación), donde se considera la Ingeniería del Software como un proceso de descubrimiento y comunicación de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades [Durán, 2000]. El trabajo de Amador Durán es un modelo de proceso que además propone una serie de técnicas en cada etapa para obtener así una metodología que facilite la obtención de los requisitos del sistema.

En este mismo trabajo se considera que *"el aspecto más importante en la Ingeniería de Requisitos es la comunicación, característica que hace de la ingeniería de requisitos una disciplina especialmente compleja ya que hay un factor que, aunque lleva siendo estudiado mucho tiempo, apenas se conoce aún: el factor humano. Este factor es el responsable de que la ingeniería de requisitos tenga aspectos sociales y culturales y no sólo técnicos [Goguen, 1994]"*.

En el apartado 5.2.1 se define el concepto de requisito tal y como se plantea tradicionalmente en la Ingeniería de Requisitos. En el apartado 5.2.2 se describe

brevemente el modelo de proceso seguido normalmente. En el apartado 5.2.3 se apuntan las técnicas de elicitación más empleadas. En el apartado 5.2.4 se describen las tareas propuestas por [Durán, 2000] en la etapa elicitación de requisitos. En el apartado 5.2.5 se muestra cómo debería ser el *Documento de Requisitos del Sistema (DRS)*, único producto resultante de la etapa de elicitación de requisitos. Tras el descripción y estudio de la Ingeniería de Requisitos y más concretamente de la metodología empleada por [Durán, 2000], en el apartado 5.2.6 se discuten algunas consideraciones realizadas para adaptar dicha esta técnica al análisis y diseño de entornos colaborativos así como a la metodología planteada en este trabajo. Una vez discutidos esos términos, en el apartado 5.2.7 se describen los pasos adaptados necesarios para realizar la primera etapa de la metodología presentada en esta tesis.

5.2.1 Definición de Requisito

Un **requisito** es, según la Real academia de la Lengua [RAE, 2007], una *circunstancia o condición necesaria para algo*.

En [Durán, 2000] se ha realizado un estudio exhaustivo sobre el concepto de requisito en Ingeniería del Software. La definición del mismo no es un estándar, como suele ocurrir en muchos casos. Se apuntan algunas de las definiciones más importantes:

- Para la IEEE [IEEE, 1990]:

(a) Una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo. (b) Una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal. (c) Una representación en forma de documento de una condición o capacidad como las expresadas en (a) o en (b).

- En [DoD, 1994]:

Característica del sistema que es una condición para su aceptación.

- En [Goguen, 1994]:

Propiedad que un sistema debería tener para tener éxito en el entorno en el que se usará.

Además realiza un resumen de las propiedades que los requisitos deben tener, que se consideran deseables y que aparecen en la bibliografía. Un requisito debería cumplir con las siguientes propiedades:

- *Comprensible por clientes y usuarios*, puesto que se trata de un canal de comunicación. Por ello se debería prescindir de notaciones formales y complejas y no deberían ser ambiguos. Una especificación de requisitos debería prepararse, en la medida de lo posible, de forma conjunta con el cliente.
- *Corrección*. Una especificación de requisitos ha de definir aquello que debe hacer el sistema, pero además debe prescindir de requisitos innecesarios.
- *No ambigüedad*. Tratando de proporcionar una única interpretación para cada requisito.
- *Compleitud*. Una especificación de requisitos completa incluye todos los requisitos identificados por el cliente así como los necesarios para la definición del sistema.

- *Consistencia*. Los requisitos especificados no deben entrar en conflicto entre ellos.
- La especificación de requisitos ha de ser *verificable*, con lo que se debería poder comprobar que el sistema final cumple los requisitos.
- La especificación de requisitos debe ser *modificable*, de manera que se puedan realizar cambios de forma fácil, completa y consistente.
- Los requisitos han de ser *rastreables*. Para ello han de poder referenciarse de manera única.
- Cada requisito del sistema ha de anotarse con la *importancia* que tiene para el cliente en relación al resto de requisitos y la *estabilidad* que se le presupone una vez alcanzado en el sistema, es decir, la probabilidad de que cambie.
- La especificación de requisitos debería ser *independiente del diseño y la implementación*. Los clientes han de expresar lo que quieren y los diseñadores y programadores se encargan de los aspectos relativos al diseño e implementación.

5.2.2 Modelo de Proceso de Ingeniería de Requisitos

El modelo de proceso de Ingeniería de Requisitos propuesto en [Durán, 2000] es un modelo iterativo que consta de las actividades de *elicitación*, *análisis* y *validación*.

La primera actividad, la *elicitación*, consiste en recoger los requisitos necesarios para la implementación del sistema como fruto de una continua interacción entre el cliente, usuarios e ingenieros de requisitos. En esta actividad se trata de conocer el dominio del problema, descubrir las necesidades reales de clientes y usuarios y consensuar los requisitos entre los propios clientes y usuarios.

La segunda actividad, el *análisis*, es a la que más importancia se le ha dado tradicionalmente. En esta metodología se le da más importancia a la elicitación, puesto que es el momento en el que se recogen los requisitos necesarios para que la aplicación tenga éxito. En esta actividad se trata de detectar conflictos en los requisitos, profundizar en el conocimiento del dominio del problema y establecer las bases para el diseño.

En la tercera actividad, la *validación*, los clientes y usuarios comprueban que la aplicación hace lo que debe, de acuerdo con los requisitos previamente establecidos. En esta actividad se trata de asegurar que los requisitos describen el producto deseado.

5.2.3 Técnicas de Elicitación de Requisitos

Existen numerosas técnicas para la elicitación de requisitos que se han venido utilizando tradicionalmente, incluso antes de que la Ingeniería de Requisitos existiera como tal. Lógicamente, son necesarios una serie de pasos para la consecución de la aplicación que quiere el cliente, y para llegar a ella, antes es necesario saber qué es lo que quiere.

Probablemente la técnica asumida naturalmente desde la antigüedad para llevar a cabo esta tarea haya sido la entrevista. Sin embargo, el número de técnicas ha ido aumentando considerablemente y son muchas las empleadas para esta primera actividad de la Ingeniería de Requisitos.

A continuación se describen las técnicas recogidas en [Durán, 2000] para la elicitación de requisitos:

- Entrevistas.

Las entrevistas son un modo natural de comunicación entre personas. Se identifican tres fases [Piattini, 1996]: *preparación, realización y análisis*.

- Joint Application Development (JAD) o Desarrollo Conjunto de Aplicaciones.

El JAD es una técnica ideada por IBM en 1977 que suele dar buenos resultados, pero que no se utiliza con frecuencia porque requiere demasiado tiempo de demasiadas personas. Consiste en una serie de reuniones de grupo en las que participan diferentes personas, donde se ayuda al cliente a expresar lo que quiere, ayudado por especialistas.

Esta técnica se basa en [Raghavan, 1994] dinámicas de grupo, ayudas visuales (diagramas, transparencias, elementos multimedia, etc.), mantenimiento de un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que se ve es lo que se obtiene).

Se identifican tres fases [Raghavan, 1994]: *adaptación* de la técnica al proyecto particular, *celebración de las sesiones JAD* para exponer, discutir, analizar y refinar las ideas de cada participante; y *conclusión* de la información generada en documentos formales, cuyas copias tendrán todos los miembros de las entrevistas.

- Brainstorming o tormenta de ideas.

La tormenta de ideas es una técnica muy utilizada por dos motivos principalmente: no requiere aprendizaje para su uso y se generan ideas con facilidad.

Consiste en una reunión moderada por un jefe de sesión que no controla sino que fomenta la participación. Cuantas más ideas surjan, por descabelladas que pudieran parecer, más ideas podrán darse a partir de éstas. Por eso no se limita, ni critica, ni cohibe, a ningún participante. El jefe de sesión debe fomentar esa participación y la combinación de ideas.

Las fases identificadas en el brainstorming [Raghavan, 1994] son: *preparación* de la sesión por parte del jefe de sesión, que seleccionará los participantes adecuados para la reunión; *generación*, donde el jefe de sesión expone la primera idea, que será la semilla a partir de la cual surgirán el resto de ideas; *consolidación*, donde se revisan, descartan y priorizan las ideas generadas en la fase anterior; y *documentación* de las ideas priorizadas por parte del jefe de sesión.

- Casos de Uso.

Los Casos de Uso son una técnica utilizada especialmente en Ingeniería del Software, propuesta por [Jacobson, 1993], que se ha empleado en UML [Booch, 1999] y que se sigue usando en UML 2.0 [OMG, 2005].

Un caso de uso es "*a mean for specifying required usages of a system*", es decir, un comportamiento, un uso requerido del sistema. Los casos de uso se definen de acuerdo a las necesidades de los actores, que son entidades externas al sistema y que interactúan con el mismo. Los casos de uso se representan por medio de diagramas de casos de uso [OMG, 2005].

- Observación in situ, estudio de documentación, cuestionarios, inmersión.

Estas y otras técnicas también se pueden emplear para completar la elicitación de requisitos.

5.2.4 Tareas en la Elicitación de Requisitos

Amador Durán en su tesis doctoral [Durán, 2000] propone toda una serie de tareas a realizar en la elicitación de requisitos, tareas que se resumen en la Figura 91.

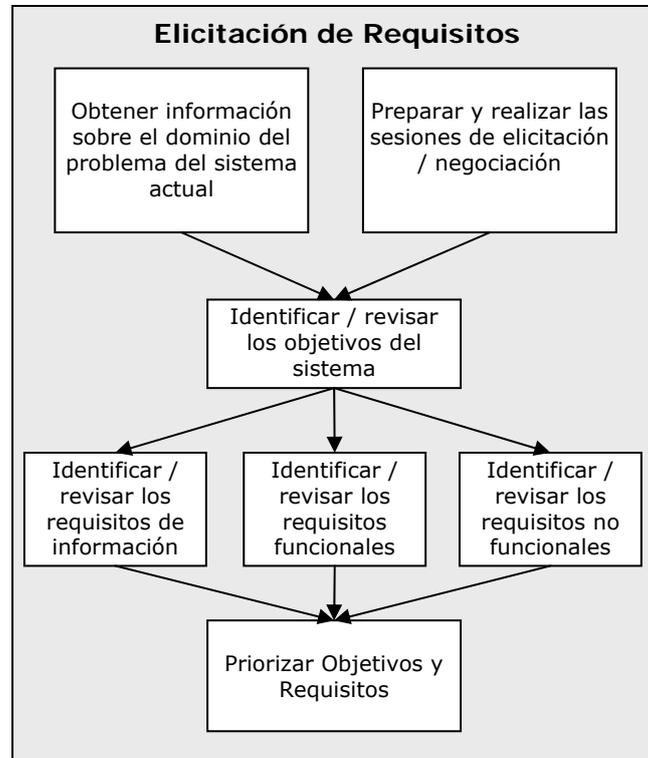


Figura 91. Tareas de elicitación de requisitos en [Durán, 2000]

Este conjunto de tareas y técnicas a emplear en cada una de ellas componen la metodología a seguir en la etapa de elicitación de requisitos del modelo de proceso que propone.

A grandes rasgos, los productos obtenidos, y las técnicas empleadas para obtenerlos en cada una es las tareas, se muestran a continuación:

- Tarea 1.- Obtener información sobre el dominio del problema del sistema actual

Es una tarea opcional que permite conocer el dominio del problema y la situación actual antes de comenzar propiamente con la recolección de requisitos.

- Técnicas Empleadas:

Recopilación de documentación, entrevistas, reuniones en grupo, cuestionarios, inmersión en el negocio del cliente, aprendizaje del negocio, modelado del sistema actual, etc.

- Productos Resultantes:

Modelos del sistema actual, documentación de la organización, resultados de entrevistas iniciales, resultados de cuestionarios exploratorios, documentación de desarrollos previos sobre el mismo dominio de problemas, información proveniente de expertos, etc.

- Tarea 2.- Preparar y realizar las sesiones de elicitación / negociación
- Técnicas Empleadas:

Técnicas de elicitación de requisitos (entrevistas, JAD, brainstorming); plantillas de objetivos, requisitos y conflictos (descritas ampliamente en la metodología); y técnicas de negociación.
- Productos Resultantes:

Notas tomadas durante las sesiones, transcripciones o actas de las sesiones, grabaciones de audio o vídeo, etc.

Así mismo, se detectan los participantes en el proyecto, objetivos, requisitos y conflictos como parte del Documento de Requisitos del Sistema (DRS).
- Tarea 3.- Identificar / revisar los objetivos del sistema
- Técnicas Empleadas:

A partir de la información obtenida en la tarea anterior se pueden identificar los objetivos que se esperan alcanzar con el despliegue del sistema a implementar.
- Técnicas Empleadas:

Técnicas como el análisis de factores críticos de éxito [MAP, 1995] u otras técnicas de identificación de objetivos. En la metodología se utilizan especialmente las plantillas para especificar los objetivos del sistema.
- Productos Resultantes:

Se obtienen, como resultado, los objetivos del sistema como parte del DRS.
- Tarea 4.- Identificar / revisar los requisitos de información
- Técnicas Empleadas:

Existen tres tipos diferentes de requisitos. Los de información, los funcionales y los no funcionales. Mediante esta tarea se identifican los primeros, aquellos relativos al almacenamiento de información que deberá cumplir el sistema software a desarrollar.
- Técnicas Empleadas:

Plantilla para requisitos de almacenamiento de información descrita en la metodología.
- Productos Resultantes:

Requisitos de almacenamiento de información como parte del DRS.
- Tarea 5.- Identificar / revisar los requisitos funcionales
- Técnicas Empleadas:

Existen tres tipos diferentes de requisitos. Los de información, los funcionales y los no funcionales. Mediante esta tarea se identifican los segundos, los funcionales (casos de uso) que deberá cumplir el sistema software a desarrollar. Se identifican previamente los actores que interactuarán con el sistema.
- Técnicas Empleadas:

Casos de uso, y plantillas para actores y para los requisitos funcionales descritas en la metodología.
- Productos Resultantes:

Requisitos funcionales como parte del DRS.

- Tarea 6.- Identificar / revisar los requisitos no funcionales

Existen tres tipos diferentes de requisitos. Los de información, los funcionales y los no funcionales. Mediante esta tarea se identifican los segundos, los no funcionales (normalmente de carácter técnico o legal) que deberá cumplir el sistema software a desarrollar.

- Técnicas Empleadas:

Plantilla para requisitos no funcionales descrita en la metodología.

- Productos Resultantes:

Requisitos no funcionales como parte del DRS.

- Tarea 7.- Priorizar Objetivos y Requisitos

5.2.5 Documento de Requisitos del Sistema (DRS)

El único documento que se obtiene como producto para la elicitación de requisitos tras el uso de esta metodología de Ingeniería de Requisitos es el *DRS* o *Documento de Requisitos del Sistema*, cuya estructura se muestra en la Figura 92.

Portada
Lista de cambios
Índice
Lista de figuras
Lista de tablas
1 Introducción
2 Participantes en el proyecto
3 Descripción del sistema actual [opcional]
4 Objetivos del sistema
5 Catálogo de requisitos del sistema
5.1 Requisitos de almacenamiento de información
5.2 Requisitos funcionales
5.2.1 Diagramas de casos de uso por objetivos del sistema
5.2.2 Descripción casos de uso más importantes por medio de las plantillas
5.3 Requisitos no funcionales
6 Matriz de rastreabilidad objetivos/requisitos
7 Conflictos pendientes de resolución [opcional, pueden ir en un documento aparte]
8 Glosario de términos [opcional]
Apéndices [opcionales]

Figura 92. Estructura del Documento de Requisitos del Sistema (DRS) a partir de [Durán, 2000]

Uno de los puntos clave del *DRS* es la *matriz de rastreabilidad* entre objetivos y requisitos. Dicha matriz muestra si existe relación entre los objetivos planteados en el sistema y los requisitos identificados. De esta manera, se puede estudiar si todo objetivo tiene requisitos asociados, si todo requisito está justificado por algún objetivo. Esta relación explícita facilita el estudio de los cambios y sus consecuencias, así como la trazabilidad entre las diferentes tareas de la metodología. La Tabla 16 muestra un ejemplo de *matriz de rastreabilidad*.

Tabla 16. Ejemplo de matriz de rastreabilidad [Durán, 2000]

	OBJ-1	OBJ-2	...	OBJ-n
RI-01	X	X		
RI-02		X		X
...				
RF-01	X			X
RF-02	X	X		
...				

5.2.6 Consideraciones para la Etapa de Elicitación de Requisitos en esta Metodología

Para el proceso de elicitación de requisitos adoptado, aun basado en la metodología de [Durán, 2000], se consideran algunos detalles particulares, bien por adaptación al modelo de proceso y la metodología presentados en este trabajo, bien por razones relativas a entornos colaborativos. Este apartado muestra las consideraciones realizadas sobre la metodología en la elicitación de requisitos presentada por [Durán, 2000]. Dichas consideraciones serán aplicadas en la primera etapa del modelo de proceso presentado en este trabajo, encaminada a realizar la elicitación de requisitos teniendo en cuenta de forma explícita las características CSCW que pudieran darse (apartado 5.2.7).

5.2.6.1 Ingeniería del Software y HCI: Objetivos, Requisitos y Tareas

Ingeniería del Software y HCI

Durante varias décadas, la *Ingeniería del Software* ha trabajado por establecer los modelos de proceso y las metodologías necesarios para conseguir abordar las necesidades de los clientes con éxito.

Para ello se han descrito una serie de pasos y técnicas a emplear en los mismos que permitieran describir el dominio del problema y el dominio de la solución de la forma más completa y coherente posible.

Se trata de un área donde el sistema a desarrollar es el protagonista. Todo gira en torno a él.

Todos los modelos de proceso en Ingeniería del software suelen comenzar con una primera etapa que tiene en cuenta las necesidades de los clientes. Es una etapa de elicitación de requisitos en la que es práctica habitual utilizar la técnica de casos de uso para recogerlos y analizarlos (un análisis de los requisitos, no del sistema). Los casos de uso fueron introducidos por [Jacobson, 1992] para capturar la funcionalidad del sistema desde el punto de vista del usuario.

El *HCI*, cuya aparición es posterior a la Ingeniería del Software, también tiene como propósito último conseguir sistemas de calidad. Sin embargo, el modo de abordar el problema es diferente. En este caso, el desarrollo del sistema está centrado en el usuario, en la interacción del usuario con el sistema.

Los modelos de proceso más extendidos en HCI normalmente comienzan con una etapa de análisis de tareas. Dichas tareas están encaminadas a satisfacer los requisitos de los clientes.

En este trabajo se ha visto la necesidad de contemplar la experiencia aportada por el campo de la Ingeniería del Software. Abordar el problema desde un análisis de

tareas en HCI puede ser un paso demasiado brusco para comenzar el desarrollo de un problema.

En este sentido, el modelo presentado en este trabajo comienza con una etapa de elicitación de requisitos como las propuestas en Ingeniería del Software. Se trata de un modelo de proceso de HCI, o más concretamente de CSCW, en el que se incluye esta primera etapa por considerarla fundamental en todo proceso de desarrollo sea el que fuere.

Lógicamente, existe una trazabilidad entre todas las etapas incluida esta primera.

Objetivos, Requisitos y Tareas

Debido al uso de metodologías y modelos de proceso de líneas diferentes, en este caso Ingeniería del Software y HCI, aparecen incoherencias terminológicas que deben ser precisadas puesto que el modelo de proceso presentado no es un híbrido. Es decir, no se trata de varios modelos y metodologías entrelazadas, sino que se presenta un modelo de proceso único, que debe tener un lenguaje coherente en todo momento (apartado 4.2).

Desde la perspectiva de la Ingeniería del Software, como se ha anticipado anteriormente centrada en el sistema, el concepto de **requisito** se podría considerar como una necesidad del cliente. En el apartado 5.2.1 se ha definido ampliamente el concepto según la Ingeniería del Requisitos y esas definiciones son ampliamente aceptadas y válidas para este trabajo:

Retomando parte de la definición de requisito elaborada por la IEEE [IEEE, 1990] un requisito es una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo.

Por otro lado, si el desarrollo del sistema se aborda desde el punto de vista del HCI, en este caso centrado en el usuario, se consideran entonces las **tareas** que los usuarios deben realizar para satisfacer sus necesidades.

Como se discutió en el apartado 4.2.2, existen multitud de aproximaciones a la definición de tarea, y por los motivos aportados en dicho apartado, en este trabajo:

Una Tarea es una porción de trabajo necesaria para lograr un objetivo determinado.

Tarea y requisito pueden ser considerados equivalentes desde puntos de vista diferentes. Tarea desde el punto de vista del usuario (HCI) y requisito desde el punto de vista del sistema (Ingeniería del Software). En ambos casos la semántica que hay detrás es la misma.

En ambas definiciones, así como en la tesis de [Durán, 2000], aparece el término **objetivo**.

Según la Real Academia de la Lengua [RAE, 2007], un objetivo es un *fin o intento a que se dirige o encamina una acción u operación*.

Uno de los propósitos de las primeras etapas en la Ingeniería de Requisitos es la identificación de los **objetivos** del sistema que se pretende desarrollar. Tanto la Ingeniería del Software como el HCI coinciden en este concepto como el o los propósitos que se pretenden abordar ya sea satisfaciendo los requisitos del sistema, ya sea llevando a cabo las tareas correspondientes.

[Durán, 2000] utiliza adicionalmente el término *objetivo del sistema* como una técnica para dividir el problema principal en varios subsistemas más sencillos de afrontar.

Como se explicó en el apartado 4.2.3.3, con el fin de simplificar el Modelo de Objetivos (Figura 88) presentado y utilizado en la metodología se considera un único concepto, el concepto de *objetivo*, que engloba otros como fin o meta:

Un objetivo es un propósito que aborda un actor del sistema por medio de la ejecución de la tarea que logra la consecución de dicho objetivo.

Los objetivos pueden descomponerse en otros objetivos de un nivel de granularidad menor, y así sucesivamente hasta que no se puedan descomponer más porque se traten de propósitos atómicos. El objetivo de mayor nivel de granularidad es el que resuelve el problema para el que se ha diseñado el sistema.

Por todo ello, en este trabajo se emplearán los términos de la siguiente manera:

- El término *requisito* sólo se empleará en la primera etapa del modelo de proceso: Elicitación de Requisitos. Se hace uso de él desde la perspectiva de la Ingeniería del Software, puesto que se trata de una etapa propia de esta línea, y además no se suele emplear en HCI.
- El término *tarea* se usa a partir de la siguiente etapa a la Elicitación de Requisitos, donde se "vuelve" a la línea de HCI para comenzar con el análisis del sistema que, entre otras técnicas, utiliza las de análisis de tareas.
- En cualquier caso los conceptos *requisito del sistema* y *tarea de usuario* son semánticamente equivalentes en este trabajo.
- El término *objetivo* es un concepto de un nivel de granularidad mayor y como se comentó con anterioridad, los objetivos son propósitos que se pretenden alcanzar ya sea satisfaciendo los requisitos del sistema, ya sea llevando a cabo las tareas correspondientes.
- También se considera interesante la idea de identificar, en primera instancia, los *objetivos del sistema* como plantea [Durán, 2000] a modo de facilitar la especificación del mismo: división del sistema en subsistemas.

5.2.6.2 Modelo de Proceso de la Ingeniería de Requisitos

[Durán, 2000] propone un modelo de proceso que comprende las etapas de elicitación de requisitos, análisis de requisitos y validación de requisitos. En cada una de esas etapas propone una serie de técnicas que conforman la metodología a seguir.

En el modelo de proceso que se presenta en este trabajo, la primera etapa es la de Elicitación de Requisitos. Dicha etapa está basada en la etapa de elicitación de requisitos propuesta por [Durán, 2000].

Se puede profundizar más en esta y en otras etapas. Una manera de profundizar sería aplicar las etapas de análisis y validación propuestas en el modelo de proceso de [Durán, 2000] teniendo en cuenta los aspectos de colaboración introducidos en la modificación a la etapa planteada en esta tesis.

Sin embargo, el objetivo de este trabajo es presentar una notación, una ontología, un modelo de proceso y una metodología para entornos CSCW. Debido a su envergadura, algunos puntos del modelo de proceso se indican como trabajos futuros que pudieran extenderlo. Este es uno de esos puntos.

El modelo de proceso presentado es completo, lo que no quiere decir que no pudiera llevarse a cabo el análisis de los requisitos y su validación en este punto; estas etapas podrían dar incluso mayor consistencia a la especificación del sistema. En esta metodología nos centraremos más en los aspectos relativos a la colaboración.

El estudio de cómo afectaría el análisis y la validación de los requisitos teniendo en cuenta las modificaciones realizadas por adaptación a esta metodología, así como por la extensión de las plantillas de elicitación para objetivos colaborativos, sería un trabajo futuro a realizar que daría mayor consistencia a los pasos a seguir desde el inicio.

5.2.6.3 Tipos de requisitos

La Ingeniería de Requisitos trata, como se ha comentado con anterioridad, de resolver el problema de la identificación de los requisitos que se plantean en un sistema. Se tienen en cuenta los tres tipos de requisitos existentes: *de información, funcionales y no funcionales*.

La información aportada sobre los requisitos descubiertos para el sistema se podría enriquecer desde esta etapa considerando si se trata de algún requisito que tenga que ver con colaboración y de qué manera.

Los sistemas antiguos estaban ideados para su uso por parte de una única persona frente a un único computador. Actualmente, la infraestructura de red, la capacidad de las nuevas máquinas y sistemas y otros avances tecnológicos hacen casi impensable el desarrollo de una aplicación que no sea colaborativa o que no incluya algún elemento típico CSCW.

Los requisitos para las primeras aplicaciones y para las aplicaciones actuales siguen siendo *de información, funcionales y no funcionales*, a pesar de la aparición del CSCW. A pesar de todo, no parece lógico agregar un nuevo tipo de requisito que fuera *de colaboración*. Más bien, un requisito de información, funcional o no funcional podría estar adicionalmente relacionado con CSCW.

Este es el motivo que nos lleva a pensar en una extensión en la recopilación de información sobre los requisitos, una extensión sobre las plantillas, sobre esos metadatos. Esta extensión contemplaría metadatos adicionales para el caso en el que hubiera que describir un requisito CSCW, independientemente de su tipo.

En este trabajo se hablará entonces de *requisitos de información, funcionales, y no funcionales*; que podrían además extenderse con aspectos CSCW. Según lo visto hasta ahora (apartado 4.2), podríamos extender la especificación de un requisito CSCW con características propias de estos entornos: *coordinación, colaboración, cooperación, comunicación*; y con características espacio-temporales: *síncrono / asíncrono, mismo lugar / lugares diferentes*. Estos y otros metadatos se describirán en el apartado siguiente.

5.2.6.4 Plantilla para Objetivos y Requisitos

En [Durán, 2000] se plantea una plantilla para recabar información sobre cada uno de los siguientes elementos:

- objetivo del sistema
- requisito de información
- requisito funcional
- requisito no funcional
- actor del sistema
- conflicto

Cada plantilla tiene una serie de metadatos asociados al elemento en cuestión, así como unos patrones lingüísticos (patrones-L) que facilitan cumplimentar esos datos (ver Anexo I).

No se ha considerado la plantilla ni el concepto de *conflicto* en esta metodología. Se trata de realizar una especificación clara y no ambigua de los elementos del sistema.

El concepto de *actor* se desarrollará más al tratarse de entornos CSCW donde se tiene en cuenta otro tipo de conceptos como *grupo*, *usuario* o *agente*. Ver apartado 4.2.1.

Estudiando los campos empleados en las plantillas para objetivos y requisitos de [Durán, 2000], se puede observar que muchos de los contenidos en los *objetivos*, *requisitos de información*, *funcionales* y *no funcionales* coinciden. Los metadatos que coinciden en las cuatro plantillas son: *Identificador* (OBJ-<id>, RF-<id>, RNF-<id>, RI-<id>), *Versión*, *Autores*, *Fuentes*, *Descripción*, *Importancia*, *Urgencia*, *Estado*, *Estabilidad*, *Comentarios*. El único patrón-L que varía ligeramente es el de *Descripción*. Los demás son iguales.

Los campos *Objetivos asociados* y *Requisitos asociados* coinciden en los *requisitos funcionales*, *no funcionales* y *de información*. En los *objetivos* aparece sin embargo el campo *Subobjetivos* que, de alguna manera, pone en relación los *objetivos*.

Algunos campos son particulares de cada plantilla:

- Requisitos de información: Datos específicos, Intervalo temporal.
- Requisitos funcionales: Precondición, Secuencia normal, Postcondición, Excepciones, Rendimiento, Frecuencia esperada.

Resumiendo, y llamando a partir de ahora a los campos *metadato*, puesto que es el término que se emplea en esta tesis normalmente, se podría decir que:

$$\forall \text{metadato} \subset \text{OBJ} \wedge \neg \{\text{Subobjetivos}\} \Rightarrow \text{metadato} \subset \text{RF}$$

$$\forall \text{metadato} \subset \text{RNF} \Rightarrow \text{metadato} \subset \text{RF}$$

$$\forall \text{metadato} \subset \text{RI} \wedge \neg \{\text{Datos_específicos}, \text{Intervalo_temporal}\} \Rightarrow \text{metadato} \subset \text{RF}$$

Siendo,

OBJ el conjunto de metadatos de los objetivos

RI el conjunto de metadatos de los requisitos de información

RNF el conjunto de metadatos de los requisitos no funcionales

RF el conjunto de metadatos de los requisitos funcionales

Del estudio de estos metadatos empleados en las plantillas se desprende que existen algunos genéricos, que están en todas ellas y otros más específicos, según aquello sobre lo que se esté recopilando información.

En este trabajo, para la elicitación de los requisitos de un sistema colaborativo se realiza una modificación al modo de emplear estas plantillas.

Existe una plantilla para la recopilación de *requisitos* y *objetivos*. Esta plantilla está formada por tres partes: una *plantilla general*, una *extensión específica* a la *plantilla general para requisitos u objetivos* y una *extensión CSCW* a la misma. La *extensión CSCW* sólo se da en el caso de que existan indicios de colaboración. La *plantilla* está formada, pues, por al menos la *plantilla general* y una *extensión específica*.

Para cada objetivo del sistema se crea su correspondiente Diagrama de Casos de Uso. Después se describirán los casos de uso más importantes por medio de la *plantilla para objetivos y requisitos* y su *extensión para requisitos funcionales*.

Plantilla general

La **plantilla general** (ver Tabla 17) facilita un marco para recoger la información relativa a los *objetivos del sistema* y a los *requisitos*.

En su mayoría, esta parte de la *plantilla para requisitos* está formada por los metadatos que se repiten en las plantillas para *objetivos del sistema*, *requisitos de información*, *funcionales* y *no funcionales* de [Durán, 2000], como se ha comentado anteriormente: *identificador* ($\{OBJ-<id>, RI-<id>, RF-<id>, RNF-<id>\}$), *Versión*, *Autores*, *Fuentes*, *Importancia*, *Urgencia*, *Estado*, *Estabilidad*, *Comentarios*. Ver Anexo I.

La *Descripción* del requisito seguirá uno de los cuatro patrones-L según sea un *objetivo*, o bien un *requisito de información*, *funcional* o *no funcional*. Por este motivo se ha dejado como metadato específico.

El identificador será $OBJ-<id>$ cuando identifique a un objetivo del sistema, $RI-<id>$ si se trata de un *requisito de información*, $RF-<id>$ si es un *requisito funcional*, o bien $RNF-<id>$ si hace referencia a un *requisito no funcional*.

Los metadatos **Objetivos asociados** y **Requisitos asociados** ponen en relación los *objetivos del sistema* y los *requisitos* entre sí, no por relaciones de jerarquía, sino más bien porque semánticamente estén relacionados y convenga expresarlo de forma explícita para facilitar la comprensión del sistema. Verlo de forma explícita evita tener que "repensar innecesariamente".

Adicionalmente se ha incluido un nuevo metadato necesario para saber quién ha de estar informado de qué y en qué momento. Se trata del metadato **Necesidad de percepción**. La percepción o awareness está cobrando cada vez mayor importancia debido al número de participantes de los entornos colaborativos y de las cantidades de información manejadas. Un primer paso para proporcionar esta percepción es incluirla como metadato a cumplimentar en la elicitación de requisitos. Probablemente se completará tras el análisis del sistema.

Resulta importante determinar qué es lo que tiene que saber quién (descripción de la información en sí), cómo ha de saberlo (notificaciones por web, por correo electrónico, avatares, etc.), cuándo ha de ser consciente de ello, dónde ha de ser informado y por qué ha de ser consciente de ello.

El metadato **Participantes** indica quién realiza el cometido. Es de vital importancia en sistemas CSCW, pero se introduce en la general puesto que un requisito o un objetivo siempre lo realiza algún actor aunque no sea CSCW. Si se trata de un requisito es colaborativo, seguro que en su consecución participan más de un actor del sistema. No sólo se precisa indicar quiénes participan, sino también cuál es su cometido particular para lograr el requisito en conjunto. El resto de metadatos particulares para requisitos CSCW se detallan más adelante en la extensión CSCW de la plantilla general.

Tabla 17. Plantilla general

{OBJ-<id>, RI-<id>, RF-<id>, RNF-<id>}	<nombre descriptivo>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) • ...
Fuentes	<ul style="list-style-type: none"> • <fuente de la versión actual> (<organización de la fuente>) • ...
Objetivos asociados	<ul style="list-style-type: none"> • #OBJ-<id> (<nombre del objetivo>) • ...
Requisitos asociados	<ul style="list-style-type: none"> • #{RI-<id>, RF-<id>, RNF-<id>} (<nombre del requisito>) • ...
Importancia	<importancia del objetivo o requisito>
Urgencia	<urgencia del objetivo o requisito >
Estado	<estado del objetivo o requisito >
Estabilidad	<estabilidad del objetivo o requisito >
Necesidad de percepción	<p>De este {objetivo, requisito} deberían estar informados los siguientes actores:</p> <ul style="list-style-type: none"> • #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>): <ul style="list-style-type: none"> - Qué: <de qué debe ser informado> - Cómo: <cómo ha de ser informado> - Cuándo: <cuándo ha de ser informado> - Dónde: <dónde ha de ser informado> - Por qué: <por qué ha de ser informado> - ... • ...
Participantes	<p>Los actores que participan en la consecución del {objetivo, requisito} son:</p> <ul style="list-style-type: none"> • #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>): <Descripción de su participación> • ...
Comentarios	<comentarios adicionales sobre el objetivo o requisito >

Extensión específica a la plantilla general

Para describir los objetivos del sistema se extiende la plantilla general con la que se muestra en la 0. Como resultado se obtiene la plantilla que propone [Durán, 2000] con alguna pequeña modificación.

Puesto que los objetivos, por definición, *pueden descomponerse en otros objetivos de un nivel de granularidad menor, y así sucesivamente hasta que no se puedan descomponer más porque se traten de propósitos atómicos*, la mayoría de los objetivos tendrán dependencias tipo padre-hijo. En la plantilla original se presenta el campo *Subobjetivos* para mostrar los objetivos que dependen del que se describe en la plantilla concreta.

Este campo se ha modificado denominándose en las nuevas plantillas *Dependencias-I*. Adicionalmente se ha añadido el metadato análogo. Los metadatos *Dependencias-S* y *Dependencias-I* relacionan los *objetivos del sistema* con otros superiores o inferiores respectivamente en ese sentido. La 0 muestra la extensión a la *plantilla general* que da lugar a la *plantilla para objetivos*.

Tabla 18. Extensión específica para objetivos

Descripción	El sistema deberá <objetivo a cumplir por el sistema>
Dependencias-S	<ul style="list-style-type: none"> • #OBJ-<id> (<nombre del objetivo>) • ...
Dependencias-I	<ul style="list-style-type: none"> • #OBJ-<id> (<nombre del objetivo>) • ...

Tabla 19. Extensión específica para requisitos de información

Descripción	El sistema deberá almacenar la información correspondiente a <concepto relevante>. En concreto:
Datos específicos	• <datos específicos sobre el concepto relevante>
Intervalo temporal	{ pasado y presente, sólo presente }

Tabla 20. Extensión específica para requisitos funcionales

Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso { durante la realización de los casos de uso <lista de casos de uso>, cuando <evento de activación> }	
Precondición	<precondición del caso de uso>	
Secuencia Normal	Paso	Acción
	p ₁	El {actor #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>), sistema} <acción/es realizada/s por actor / sistema>
	p ₂	Se realiza el caso de uso <caso de uso (#RF-x)>
	p ₃	Si <condición>, el {actor #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>), sistema} <acción/es realizada/s por actor / sistema>
	p ₄	Si <condición>, se realiza el caso de uso <caso de uso (#RF-x)>

Postcondición	<postcondición del caso de uso>	
Excepciones	Paso	Acción
	p _i	Si <condición de excepción>, el {actor #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>), sistema} <acción/es realizada/s por actor / sistema>, a continuación este caso de uso {continúa , termina}
	p _j	Si <condición de excepción>, se realiza el caso de uso <caso de uso (#RF-x)>, a continuación este caso de uso {continúa , termina}

Rendimiento	Paso	Cota de tiempo
	q	m <unidad de tiempo>
Frecuencia esperada	<nº de veces> veces / <unidad de tiempo>	

Tabla 21. Extensión específica para requisitos no funcionales

Descripción	El sistema deberá <capacidad del sistema>.
--------------------	--

La extensión de la *plantilla general* con los metadatos correspondientes según sean *requisitos de información*, *funcionales* o *no funcionales*, completa la información relativa a la elicitación de los requisitos particulares por su tipo.

Como se puede observar en la Tabla 19, en la Tabla 20 y en la Tabla 21, los metadatos empleados para la extensión específica por tipo de requisito son los mismos que se utilizan particularmente en cada uno de los tipos de requisitos en [Durán, 2000]. Para más información, consultar el Anexo I.

Extensión CSCW de la plantilla general para requisitos

Por último, cada plantilla se extiende con una serie de metadatos adicionales si se trata de un *objetivo del sistema* o un *requisito* relacionados con la colaboración. Esta información es adicional y sólo se extiende la plantilla en el caso de que se trate de un *objetivo del sistema* o *requisito* colaborativos. Ver Tabla 22.

En la descripción de los metadatos empleados en la extensión CSCW se utiliza el término *requisito*, pero es igualmente aplicable a los objetivos del sistema. Se ha seguido la notación para los patrones-L empleados por [Durán, 2000]. Los metadatos de la *extensión CSCW de la plantilla general* son los siguientes:

- **Descripción CSCW.** Además de la descripción realizada para el requisito en el campo anterior de la plantilla general, se puede complementar la especificación del requisito con información exclusivamente relativa a su enfoque CSCW. Es una explicación textual que trata de describir la naturaleza colaborativa del requisito.
 - **Descripción del entorno.** El entorno donde tendría lugar el requisito es fundamental para una buena comprensión del mismo. Es necesario describir de la mejor manera posible la o las salas donde se ejecutará este objetivo / requisito, qué materiales son necesarios, recursos, etc.
 - **Características CSCW. Coordinación, Cooperación, Colaboración, Comunicación.** Se trata de las cuatro características típicas del CSCW que han sido contempladas en esta tesis según se ha explicado en el apartado 4.2.2.4. No siempre se caracterizará como todas, pero sí como al menos una de ellas. Además, la intersección de todas estas características, junto con los metadatos *Espacio* y *Tiempo*, pueden clasificar el sistema según la matriz de clasificación definida en el apartado 2.4.3.
 - **Características espacio/temporales. Espacio, Tiempo.** Se trata de las características expresadas por Johansen explicadas en el apartado 4.2.2.5 para clasificar las herramientas groupware según estas cualidades. Según el *Espacio*, el requisito puede tener lugar en el mismo espacio o en espacios diferentes. Según el *Tiempo*, el requisito puede realizarse de forma síncrona o de forma asíncrona.
- Junto con las características CSCW pueden clasificar el sistema según la matriz de clasificación definida en el apartado 2.4.3.
- **Nivel de exigencias.** Se trata de explicar factores como, por ejemplo, si en una aplicación síncrona el tiempo real ha de ser extremadamente parecido a la realidad (mucha precisión) o por el contrario admite retrasos, de cuánto podrían ser esos retrasos, pérdidas, etc.

Es importante destacar que los subobjetivos (dependencias-I) de un objetivo colaborativo serán necesarios para la consecución del primero.

Tabla 22. Extensión CSCW de la plantilla general

Descripción CSCW	Por la naturaleza colaborativa del {objetivo, requisito}, se debería tener en cuenta lo siguiente: • <información colaborativa adicional si necesario> • ...
Descripción del entorno	El entorno de ejecución del sistema será: • <información relativa al dónde, a lo que hay alrededor, etc.> • ...
Coordinación	{Sí, No}. <Si sí, descripción de la coordinación>
Cooperación	{Sí, No}. <Si sí, descripción de la cooperación>
Colaboración	{Sí, No}. <Si sí, descripción de la colaboración>
Comunicación	{Sí, No}. <Si sí, descripción de la comunicación>
Espacio	{Mismo, Diferente} <Nota adicional si necesario>
Tiempo	{Síncrono, Asíncrono} <Nota adicional si necesario>
Nivel de exigencias	<Descripción del nivel de exigencias en cuanto a características espacio/temporales, etc.>

El metadato *Necesidad de percepción* es información relacionada con los objetivos y requisitos colaborativos, sin embargo es necesario que se encuentre en la *plantilla general* puesto que puede requerirse de *objetivos del sistema* y *requisitos* no colaborativos en sí.

5.2.6.5 Plantillas de Participantes

Con el objeto de identificar los participantes del sistema y de hacerlo de manera que se pueda reflejar la estructura organizativa de los mismos, también se ha tenido en cuenta la plantilla empleada en [Durán, 2000] para recoger información relativa a los actores del sistema. En cualquier caso, se ha ampliado y modificado puesto que en esta metodología se tienen en cuenta, explícitamente, el manejo de *grupos, usuarios y agentes*.

Como se ha detallado en el apartado 4.2.1, la "*estructura organizativa de los participantes de un sistema colaborativo es la expresión de la distribución de sus elementos, incluyendo sus relaciones de pertenencia a grupos y sus relaciones de desempeño de roles... Se define de acuerdo a algunos conceptos que se conocen como elementos organizativos (grupo, rol, actor, individuo, usuario y agente) así como en base a algunas relaciones organizativas entre dichos elementos...*".

Por tanto, tener una idea clara de cómo se estructuran los participantes en el sistema es fundamental y se puede tener en cuenta también desde el principio, aunque vayan sufriendo modificaciones con el análisis del sistema.

En primer lugar se utilizará una plantilla para recoger la información, en modo textual, de esos participantes, de sus agrupaciones y relaciones: *plantilla para la estructura organizativa* de los participantes de un sistema. Debe ser una descripción sencilla que será ampliada en las siguientes plantillas relacionadas con los actores del sistema. Actores que se entienden según lo definido en esta metodología (apartado 4.2.1).

Así mismo, las plantillas contienen información que es delicada puesto que deberían, en todo momento, ser coherentes entre sí. Por ejemplo, los actores que aparecen en la plantilla relativa a la estructura organizativa, deben aparecer detallados en una plantilla específica para dicho actor.

Plantilla para la Estructura Organizativa

Tal y como está planteado en esta tesis, la estructura organizativa de los participantes de un sistema es única. Los participantes se organizan de una determinada manera para llevar a cabo el conjunto de objetivos para el que se diseña el sistema. Esa organización incluye agrupaciones, jerarquías, relaciones de colaboración, etc.

Los metadatos *Versión, Autores, Fuentes y Comentarios* coinciden con los presentados en [Durán, 2000] (descritos en el Anexo I).

Los metadatos **Actores** e **Individuos** se corresponden con participantes abstractos. Cabe la posibilidad de no tener muy claro en un principio si quien o quienes deben desempeñar ciertas funciones deben ser un grupo o un individuo. También sería posible no conocer con certeza si un participante del sistema que ha de realizar un conjunto de tareas será un participante humano o por el contrario un agente que realice las tareas de forma automática. Estos son dos ejemplos que conducen a la necesidad de poder utilizar *Actores* e *Individuos* como participantes.

Probablemente se tenga una idea concebida de antemano de qué es lo que serán esos *Actores* e *Individuos*. Es decir, si los *Actores* serán finalmente *Grupos, Usuarios* o *Agentes*, o si los *Individuos* serán *Usuarios* o *Agentes*. Esa información se incluye igualmente en la plantilla dando información al respecto.

Los **Grupos, Usuarios** y **Agentes** son participantes algo más concretos, ya se sabe que esos actores son eso en concreto. Es deseable tener el menor número de *Actores* y de *Individuos* puesto que cuando el sistema esté completamente especificado este número llegará a cero.

Para los metadatos *Actores*, *Individuos*, *Grupos*, *Usuarios* y *Agentes*, se requiere únicamente que sean identificados. Para cada uno de ellos habrá una plantilla que recabe su información más detallada.

La **Descripción** de la *Estructura Organizativa* permite tener una idea rápida de su forma. Se describe cómo se organizan los grupos y si existen relaciones entre ellos. Además se incluye la información relativa al resto de participantes que se considere oportuna (información en torno a la *Estructura Organizativa*).

La Tabla 23 muestra la plantilla diseñada para recabar la información relativa a la Estructura Organizativa de los participantes de un sistema.

Tabla 23. Plantilla para la Estructura Organizativa de los participantes del sistema

Estructura Organizativa	
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) • ...
Fuentes	<ul style="list-style-type: none"> • <fuente de la versión actual> (<organización de la fuente>) • ...
Actores	<ul style="list-style-type: none"> • #A-<id> (<nombre descriptivo del actor>). Tiende a {Grupo, Usuario, Agente} porque <explicación de la hipótesis> • ...
-- Grupos	<ul style="list-style-type: none"> • #G-<id> (<nombre descriptivo del grupo>) • ...
-- Individuos	<ul style="list-style-type: none"> • #I-<id> (<nombre descriptivo del actor individuo>). Tiende a {Usuario, Agente} porque <explicación de la hipótesis> • ...
-- Usuarios	<ul style="list-style-type: none"> • #U-<id> (<nombre descriptivo del usuario>) • ...
-- Agentes	<ul style="list-style-type: none"> • #S-<id> (<nombre descriptivo del agente>) • ...
Descripción	Los participantes están organizados en los siguientes grupos:<listado de grupos y descripción de sus jerarquías y relaciones>. <información relativa al resto de participantes que se considere oportuna>
Comentarios	<comentarios adicionales sobre la Estructura Organizativa>

Plantilla para Actores

La plantilla diseñada para recabar información sobre los participantes del sistema está formada por una plantilla general y varias específicas, de forma similar a como se ha diseñado para los objetivos y requisitos.

Especialmente al inicio de la especificación de un sistema se tiene una idea difusa de los participantes del mismo, que con el tiempo, análisis y otras discusiones se va clarificando en diversas iteraciones. Se pasa entonces a concretar esos *Actores* en *Grupos* o *Individuos* y esos *Individuos* en *Usuarios* o *Agentes*.

Por tanto, siempre se podrá tomar una información genérica a todos los participantes puesto que tanto *Grupos*, como *Usuarios* o *Agentes* son *Actores*. La plantilla general para los participantes del sistema es una plantilla con información sobre los participantes como *Actores* (en general) del sistema.

Esta plantilla general se extiende a medida que se conozca más sobre los *Actores*. Es decir, si se sabe que un *Actor* concreto es un *Grupo*, la *plantilla para Actores* se extenderá con la *extensión específica para Grupos* (ver Tabla 25). Si por el contrario el *Actor* es un *Individuo* se extenderá con la extensión específica para *Individuos*.

La *plantilla de Actores* contiene los siguientes metadatos (Tabla 24):

- **Versión, Autores, Fuentes, y Comentarios** coinciden con los presentados en [Durán, 2000] (ver Anexo I).
- **Descripción del Actor.**
- **Supergrupos.** Generalmente los *Actores* pertenecen a *Grupos*. Este metadato expresa explícitamente esa relación de pertenencia a otros *Grupos* y la descripción necesaria en caso de que sea necesaria.
- **Jerarquía superior y Jerarquía inferior.** En muchas ocasiones existen relaciones de jerarquía superior e inferior entre *Actores*. Por ejemplo, un *Usuario* U-1 puede ser jefe de otro U-2. El *Usuario* U-1 tiene una relación de jerarquía inferior con U-2 y U-2 tiene una relación de jerarquía superior con U-1. De nuevo, este tipo de metadatos ayudan a comprender mejor el sistema en su globalidad.
- **Otras asociaciones.** Posiblemente existan relaciones no explícitas de unos actores con otros, que no ayuden directamente al desarrollo del sistema, pero cuya percepción explícita por parte de los analistas, diseñadores, desarrolladores, etc. del sistema pueda facilitar su comprensión. Esa mejor comprensión global del sistema ayuda a un desarrollo de mayor calidad. Este metadato posibilita expresar esta información, esas asociaciones (quizás semánticas) con otros actores.
- **Capacidades.** Metadato basado en el concepto de capacidad de la tesis de [Garrido, 2003]. Las Capacidades de los Actores son habilidades o responsabilidades asociadas al mismo que le permiten desempeñar roles y llevar a cabo tareas. Permite conocer al Actor en una nueva dimensión. Permite conocerlo por aquello de lo que es capaz, independientemente de lo que haga.

Tabla 24. Plantilla para *Actores*

A- <id>	<nombre descriptivo>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) • ...
Fuentes	<ul style="list-style-type: none"> • <fuente de la versión actual> (<organización de la fuente>) • ...
Descripción	<descripción relativa al grupo>
Supergrupos	<p>Este Actor es parte de los siguientes grupos:</p> <ul style="list-style-type: none"> • #G-<id> (<nombre descriptivo del actor>): <descripción de la relación de pertenencia si procede> • ...
Jerarquía superior	<p>Este Actor depende jerárquicamente de los siguientes Actores:</p> <ul style="list-style-type: none"> • #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>): <descripción de la relación de jerarquía> • ...
Jerarquía inferior	<p>Los siguientes actores dependen jerárquicamente de este Actor:</p> <ul style="list-style-type: none"> • #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>): <descripción de la relación de jerarquía> • ...
Otras asociaciones	<p>Existen estas otras relaciones:</p> <ul style="list-style-type: none"> • #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>): <descripción de la asociación el actor con este otro> • ...
Capacidades	<p>Habilidades o responsabilidades del Actor:</p> <ul style="list-style-type: none"> • {H-<id>, R-<id>}: <descripción de la capacidad del actor> • ...
Comentarios	<comentarios adicionales sobre el actor>

De las definiciones de los conceptos de *Actor*, *Grupo*, etc. (ver apartado 4.2.1) se comenta que los *Actores* se agrupan entre sí y que ese conjunto de *Actores* logra el objetivo común del *Grupo* llevando a cabo las *Tareas* que le vienen asignadas según los *Roles* que juegan. Tanto *Roles* como *Tareas* son una información que no se corresponde con la etapa de elicitación de requisitos, sino con el posterior análisis. Por tanto no aparecerán metadatos relativos a estos conceptos.

Tabla 25. Extensión específica para Grupos

Grupo	
Objetivo común	#OBJ-<id> (<nombre del objetivo>)
Pertenencia	Este Grupo está formado por los siguientes participantes: <ul style="list-style-type: none"> • #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>): <descripción de la relación de pertenencia si procede> • ...
Leyes	Normas impuestas por el grupo: <ul style="list-style-type: none"> • L-<id>: <descripción de la ley> • ...

Extensión específica para Grupos

Si se identifica que un *Actor* es un *Grupo* se habrá de extender la plantilla para *Actores* con la *extensión específica para Grupos* que se muestra en la Tabla 25.

Puesto que un *Grupo* es un *conjunto de Actores, individuos o colectivos, que desempeñan Roles para lograr un Objetivo común*, los *Actores* identificados como *Grupos* tienen además los siguientes metadatos:

- **Objetivo común.** Por la propia definición de *Grupo*, éste no tendría sentido sin ese *Objetivo*. El *Objetivo* está descrito e identificado en su *plantilla para Objetivos* correspondiente (ver apartado 4.2.3).
- **Pertenencia.** Un *Grupo* es un conjunto de *Actores*, por lo que se habrá de indicar qué *Actores* forman parte de este *Grupo*. En su caso, se puede describir esta situación.
- **Leyes.** Metadato basado en el concepto de ley de la tesis de [Garrido, 2003]. Una ley es una norma impuesta por un grupo que restringe su funcionamiento en base a reglas sociales, culturales, capacidades de los actores, etc. Establece de algún modo restricciones o reglas de comportamiento que se presuponen para los miembros del grupo.

Extensión específica para Individuos, Usuarios y Agentes

Recordemos que según las definiciones establecidas por la ontología de uso de esta metodología (ver apartado 4.2.1):

Un elemento *Individuo* es un, y sólo un, *Actor* que desempeña roles.

Un *Usuario* es un elemento *Individuo* humano.

Un *Agente* es un elemento *Individuo* no humano.

En principio no se han identificado metadatos relevantes para estos actores, pero si se identifica que un *Actor* es un *Individuo* y se detecta algún metadato relevante, se habría de extender la plantilla para *Actores* de acuerdo con el modelo seguido hasta ahora y representado en la Tabla 26: modelo para la *extensión específica para individuos, usuarios y agentes*.

Tabla 26. Modelo para la extensión específica para *individuos, usuarios y agentes*

{Individuos, Usuarios, Agentes}	
Metadato	descripción

5.2.7 Elicitación de Requisitos en Entornos CSCW

Cuando se habla de un entorno colaborativo, aproximaciones a la elicitación de requisitos como la descrita en este apartado [Durán, 2000] son perfectamente válidas debido a su carácter genérico. El estudio del sistema colaborativo puede partir del conocimiento de los requisitos del sistema. Este conocimiento lo facilita la tarea de elicitación de requisitos que tiene como consecuencia el DRS o Documento de Requisitos del Sistema.

Por lo tanto, el DRS que puede ser el punto de partida para comenzar con el análisis del sistema colaborativo.

Sin embargo, se ha considerado oportuno introducir algunos puntos para completar algunas tareas relativas a la elicitación de requisitos. De esta manera, se puede hacer uso de la tarea de elicitación de requisitos según [Durán, 2000] para identificarlos, pero atendiendo a estas consideraciones sobre entornos colaborativos que se pueden tener desde el principio del ciclo de vida del software, puesto que facilitarán posetiormente el análisis del sistema si se tienen en cuenta desde tan temprana fase.

5.2.7.1 Aportaciones CSCW introducidas sobre la Elicitación de Requisitos

Las aportaciones introducidas sobre la elicitación de requisitos para adaptar la etapa propuesta por [Durán, 2000] a esta metodología y a la visión colaborativa del sistema que se pretende analizar y diseñar son las siguientes:

- Modificación de las plantillas de objetivos y requisitos: *plantilla general de objetivos y requisitos y extensiones específicas.*
- Extensión CSCW a la plantilla general de objetivos y requisitos.
- Inclusión de la plantilla para la Estructura Organizativa.
- Identificación enriquecida de actores del sistema: extensión de la plantilla para actores (*Actores, Grupos, Individuos, Usuarios y Agentes*).
- Discusión sobre el uso de los términos *requisito, tarea y objetivo* en los campos de Ingeniería del Software y HCI para su uso adecuado.
- La matriz de reastreabilidad podría verse modificada por las inclusiones realizadas al contemplar la colaboración de forma explícita.
- Como conclusión a todos estos puntos, un DRS enriquecido para contemplar de algún modo la posible colaboración desde el inicio.

5.2.7.2 Elicitación de Requisitos en Entornos CSCW

Puesto que la tesis de [Durán, 2000] ha supuesto un gran esfuerzo en el estudio de la elicitación de requisitos y los pasos necesarios que se han de hacer para hacerlo de forma correcta y esta tesis basa su primera etapa en ella, los pasos que se siguen en la Elicitación de Requisitos son similares.

Lógicamente se han de tener en cuenta todas las consideraciones del punto 5.2.6 con respecto a la adaptación de la metodología de [Durán, 2000] a esta metodología para entornos CSCW.

A continuación se describen los pasos (mejor que tareas) (Figura 93) de la etapa de Elicitación de Requisitos de forma similar a como se ha descrito en el apartado 5.2.4:

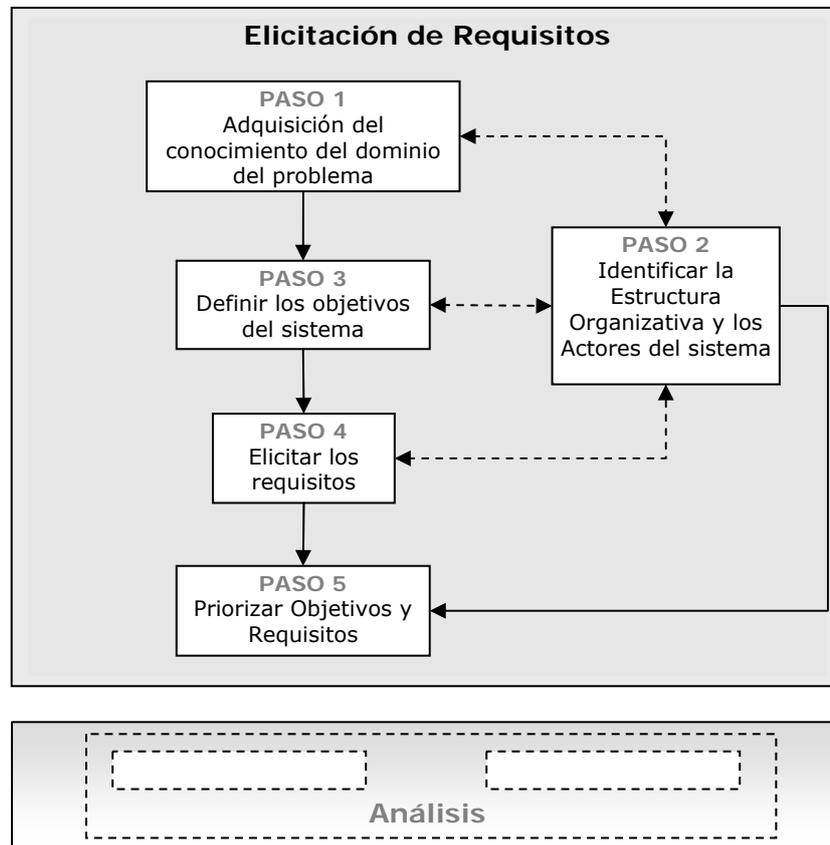


Figura 93. Pasos de la etapa de elicitación de requisitos

Paso 1: Adquisición del conocimiento del dominio del problema

Este paso de preparación a la elicitación de requisitos consiste en la recuperación de toda la información posible acerca del dominio del problema del futuro sistema y su situación actual incluso antes de comenzar su estudio y análisis.

Se trata pues de contar con la información suficiente como para comenzar a introducirse en el nuevo proyecto, entendiendo un poco mejor lo que se quiere resolver. Es un paso muy genérico común a cualquier proyecto que se quiera especificar, sea colaborativo o no.

Para este paso se emplean las técnicas usadas en las tareas 1 y 2 de la metodología de [Durán, 2000]: *Obtener información sobre el dominio del problema y el sistema actual* y *Preparar y realizar las sesiones de elicitación / negociación*.

En cualquier caso, se han introducido dos diferencias con respecto a la original, además de fundir las dos tareas en un solo paso: (1) En las técnicas empleadas para preparar y realizar las sesiones de elicitación, las plantillas que se emplean son las descritas en esta tesis con las modificaciones y consideraciones explicadas con anterioridad: *plantilla para objetivos y requisitos*; y (2) la *identificación de los*

participantes del sistema se realiza en el paso 2 puesto que gana peso en esta metodología.

- **Técnicas Empleadas:**

Recopilación de documentación, entrevistas, reuniones en grupo, cuestionarios, inmersión en el negocio del cliente, aprendizaje del negocio, modelado del sistema actual, etc.

Técnicas de elicitación de requisitos (entrevistas, JAD, brainstorming) y técnicas de negociación.

A modo de guía o de "orden del día" se pueden emplear las plantillas para objetivos y requisitos: plantilla general para objetivos y requisitos, más la extensión específica correspondiente y la extensión CSCW en su caso.

- **Productos Resultantes:**

Modelos del sistema actual, documentación de la organización, resultados de entrevistas iniciales, resultados de cuestionarios exploratorios, documentación de desarrollos previos sobre el mismo dominio de problemas, información proveniente de expertos, etc.

Notas tomadas durante las sesiones, transcripciones o actas de las sesiones, grabaciones de audio o vídeo, etc. *Se obtiene un borrador del Documento de Requisitos del Sistema (DRS) conteniendo toda esta información en forma textual, anexos, etc. Se comienzan a cumplimentar las plantillas empleadas como técnica.*

Paso 2: Identificar la Estructura Organizativa y los Actores del sistema.

Tras obtener información del dominio del problema, del sistema actual, preparar las sesiones de elicitación se tiene un pequeño background de lo que se pretende construir. Es decir, el analista o la persona encargada de realizar esta etapa de elicitación de requisitos comienza a saber de qué va el sistema antes de tener un primer contacto real.

Del mismo modo que las reuniones se empiezan con un orden del día, el paso 1 anterior supone un punto por donde empezar. En ese momento también se puede tener una idea de cómo será la Estructura Organizativa de los participantes del sistema que se quiere analizar y diseñar.

Por este motivo, se han de identificar los participantes desde el principio. Esta información que comienza a recabarse tras ese pre-estudio del sistema en el paso 1, se irá completando en los pasos 3 y 4.

El paso 2 no es secuencial. Se trata de un paso paralelo a los pasos 1, 3 y 4 que se irá enriqueciendo constantemente según el desarrollo de los mismos en diversas iteraciones.

- **Técnicas Empleadas:**

Plantilla para la Estructura Organizativa; plantilla para Actores: extensiones específicas para Grupos, Individuos, Usuarios y Agentes.

- **Productos Resultantes:**

Apartado 2 del documento de Requisitos del Sistema (DRS) conteniendo las plantillas empleadas como técnica.

Paso 3: Elicitar Objetivos del Sistema

Una vez realizadas las reuniones pertinentes y analizada toda la información que se tiene, conviene tratar de identificar clara y explícitamente cada uno de los *objetivos del sistema*.

Durante el desarrollo de este paso, el paso 2 donde se identifican los *actores* del sistema y la *Estructura Organizativa* de los mismos se enriquece y modifica según los resultados que se vayan obteniendo.

Tras la identificación de los *objetivos del sistema*, se puede tener una nueva reunión con el cliente para contrastar que dichos objetivos son los que realmente se quieren y si se han identificado correctamente los *actores* del sistema y su estructura hasta ahora. Se ha conseguido dividir el sistema en diferentes subsistemas más sencillos.

- **Técnicas Empleadas:**

Con el estudio de la información obtenida hasta ahora (en su mayoría en el borrador del DRS), se analiza qué introducir en las plantillas para objetivos como objetivos del sistema.

- **Productos Resultantes:**

En este momento se comienza a construir la versión v0.x del DRS que incluye los objetivos del sistema identificados. Esta versión se irá construyendo en los pasos 4 y 5 hasta llegar a un consenso final que eleve la versión a publicada (v1.x). En este caso la "x" permite incrementar el número de la versión cuando haya modificaciones significativas en el DRS.

Paso 4: Elicitar Requisitos

La elicitación de *requisitos de información, funcionales y no funcionales* también se realiza a partir del estudio y análisis de la información obtenida en las etapas anteriores (borrador y v0.x del DRS).

Se identifican los *requisitos de información*, relativos al almacenamiento de información que deberá cumplir el sistema software a desarrollar; los *requisitos funcionales*, (casos de uso) que deberá cumplir el sistema software a desarrollar; y los *requisitos no funcionales*, (normalmente de carácter técnico o legal) que deberá cumplir el sistema software a desarrollar. Todo ello teniendo en cuenta los metadatos incluidos en las plantillas y la *extensión CSCW* en caso de que sea necesaria.

- **Técnicas Empleadas:**

Plantillas para requisitos: plantilla general para requisitos, más la extensión específica correspondiente y la extensión CSCW en su caso.

Casos de uso

Se modifican las plantillas para participantes (paso 2).

- **Productos Resultantes:**

Requisitos de información, funcionales, funcionales, extensiones colaborativas como parte de la versión v0.x del DRS. Actualización de plantillas de participantes en el DRS (paso 2).

Paso 5: Priorizar Objetivos y Requisitos

Se trata únicamente de ordenar por prioridad los objetivos del sistema y los requisitos. Se obtiene el DRS en su versión v1.x, listo para ser utilizada en la siguiente etapa de la metodología.

5.3 Etapa 2.- Análisis

Una vez que el Documento de Requisitos del Sistema (DRS), resultado de la etapa anterior, ha llegado a su versión v1.x significa que ha sido contrastado con el cliente y con los analistas, desarrolladores, etc. correspondientes y se ha llegado a un acuerdo sólido sobre qué es lo que se quiere.

En principio, a partir del DRS se puede afirmar que hay una base firme sobre la que "construir" el sistema.

Cada nuevo cambio importante supondría pasar por esa etapa primera hasta llegar a la versión v1.x+1, lo que podría suponer, lógicamente, un cambio considerable en el resto de etapas. Por eso se trata de un modelo de proceso iterativo e incremental al estilo de otras metodologías como XP [Beck, 2000].

"Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar." Kent Beck.

La etapa de análisis en las metodologías se corresponde con el estudio del dominio del problema. Esta etapa trata de descubrir el *qué* describiendo los requisitos del sistema sin describir asuntos de implementación.

5.3.1 Análisis del Sistema en Entornos CSCW

Tras la elicitación de requisitos es el momento de analizar la información recabada. Es una etapa previa al diseño del sistema. Se estudian los elementos del dominio del problema y sus relaciones. Se trata de acercar la especificación del sistema recogida en la etapa anterior en un "lenguaje más cercano a la persona" a una especificación más cercana al desarrollador.

La Figura 94 muestra un esquema de los pasos que se siguen en la etapa de análisis del modelo de proceso que se presenta en este trabajo: identificación y descripción de roles, identificación y descripción de tareas, análisis de la estructura del sistema y análisis del comportamiento del sistema.

El análisis de un sistema colaborativo comienza con una identificación previa de roles y tareas a partir de la información recabada en el DRS de la etapa anterior. Este comienzo es fruto de la trazabilidad existente entre las etapas de elicitación de requisitos y análisis, que permite identificar roles a partir de actores y tareas a partir de requisitos. Esta trazabilidad será explicada posteriormente con mayor detalle.

Una vez identificados los primeros roles y tareas, se puede comenzar la descripción de los mismos y en paralelo los primeros diagramas de tareas (TD), colaboración (CD) y estructura organizativa (OSD). Estos diagramas se describirán en sucesivos apartados.

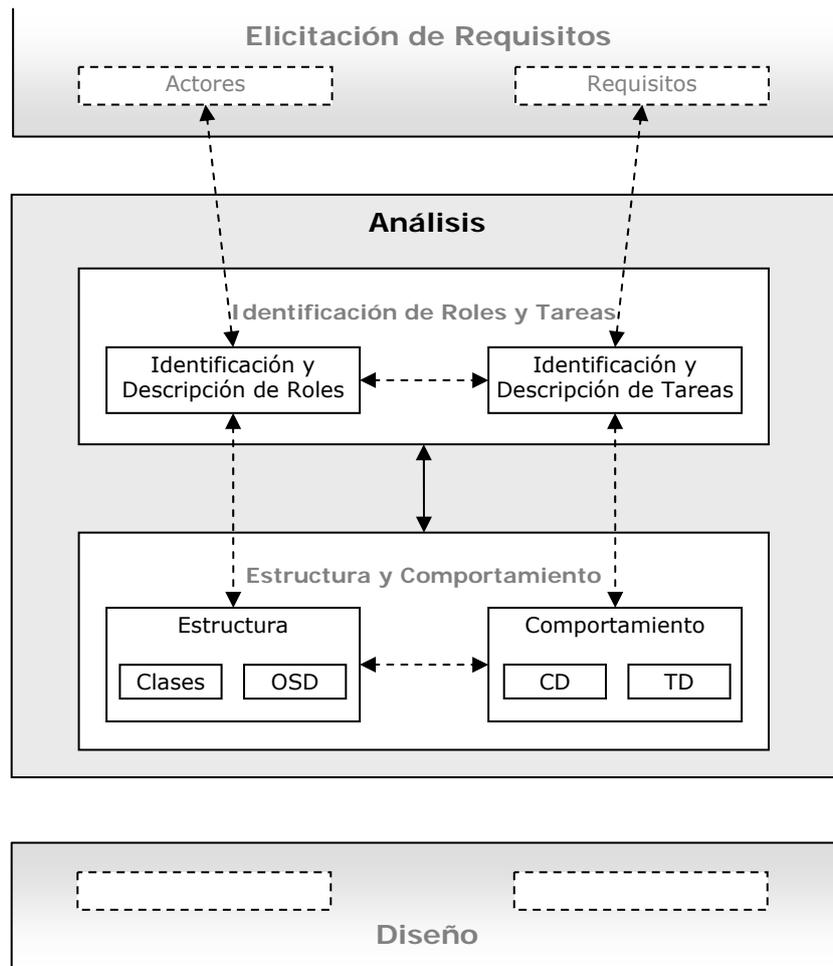


Figura 94. Pasos de la etapa de Análisis

Por un lado, la identificación de los primeros roles permite al analista realizar un primer esbozo de la estructura organizativa de los actores del sistema. Este diagrama, junto con el diagrama de clases de UML modelan la estructura del sistema. Así mismo, la identificación de las primeras tareas permite ordenarlas, creando así un primer esbozo de los diagramas de tareas y los diagramas de co-interacciones, que modelan el comportamiento de los actores del sistema.

Por otro lado, del análisis de las primeras versiones de los diagramas surgen nuevos roles y tareas, es decir, se identifican y describen nuevos roles y tareas como resultado de su estudio.

Tras esa primera identificación de roles y tareas, un proceso iterativo de refinamiento permitirá obtener un modelo de mayor calidad, que permita obtener una especificación completa del sistema.

Del mismo modo que la trazabilidad entre las etapas de elicitación de requisitos y análisis permiten obtener especificaciones coherentes y completas, es necesario tener en cuenta una trazabilidad intra-etapa entre los pasos de la misma. Dicha trazabilidad se detallará más adelante.

5.3.2 Identificación de Roles y Tareas

Tras la primera etapa del modelo de proceso se han recogido los requisitos del sistema. El DRS, documento resultante de esa etapa, contiene, entre otras cosas, los *actores* del sistema y la información relativa a los *requisitos*.

Para continuar con el análisis del sistema, es necesario tener en cuenta dos nuevos conceptos: *Rol* y *Tarea* (apartados 4.2.1.1 y 4.2.2.1 respectivamente). La relación entre roles y tareas es tan estrecha que la identificación y descripción de roles irá muy ligada y en paralelo a la identificación y descripción de tareas.

Conocidos los requisitos del sistema y los actores que intervienen en él, se pueden generar una serie de roles que determinen qué hace quién. Se trata de algo determinante en la organización del diseño de un sistema colaborativo.

Para describir los roles y las tareas identificados, se emplearán unas *plantillas* similares a las utilizadas en la elicitación de requisitos. Son plantillas más sencillas en cuanto a número de metadatos que describan el concepto en cuestión, pero necesarias para ubicar rápidamente todos los roles y tareas del sistema.

Además se emplearán *matrices de rastreabilidad* como la propuesta para poner en relación objetivos del sistema con los requisitos en la etapa de elicitación de requisitos de [Durán, 2000]. En este caso se ponen en relación actores con roles y requisitos con tareas. Las matrices de rastreabilidad simplemente muestran de forma explícita información que ya está recogida, con el objetivo de facilitar al analista, desarrollador, etc. el acceso a la misma. Una última matriz pone en relación las tareas asociadas a cada rol.

La identificación de los roles y las tareas, su descripción y el uso de las plantillas y las matrices de rastreabilidad se describen en los siguientes sub-apartados.

5.3.2.1 Identificación y Descripción de Roles

"Users are of interest to developers, not as people, but because of the roles they will play in relationship with the system to be designed and built" [Constantine, 1999]

El concepto de rol se ha definido como *conjunto de Tareas que puede desempeñar un Actor. Un Actor es lo que es debido al Rol que desempeña en cada momento en el sistema* (apartados 4.2.1.1).

En [Constantine, 1999] se dice que un rol (user role) es una colección abstracta de necesidades, intereses, expectativas, comportamientos y responsabilidades, y que un conjunto de estos elementos, constituyen roles diferentes. Un modelo de roles es una lista de los roles que hay en el sistema, con cada rol descrito en términos de las necesidades, intereses, expectativas, comportamientos y responsabilidades que lo caracterizan. Cada rol ha de tener un nombre asociado que capture su naturaleza.

La Figura 95 muestra los pasos que se siguen en la identificación y descripción de los roles en este trabajo, los pasos que se describen a continuación.

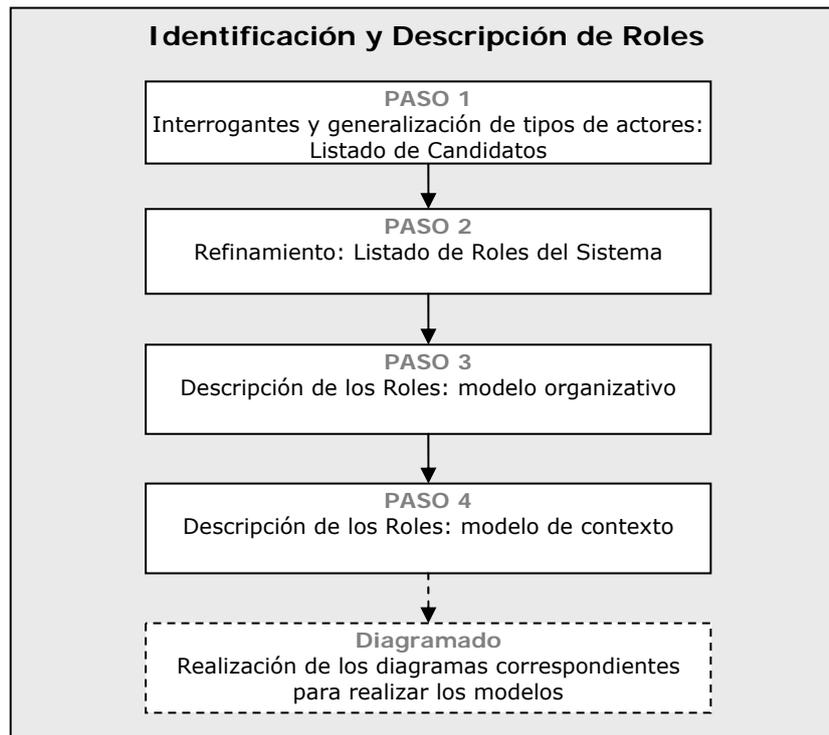


Figura 95. Pasos en la identificación y descripción de roles

Para construir ese listado de roles [Constantine, 1999] propone una serie de interrogantes previos:

- ¿Quiénes usarían o podrían usar el sistema?
- ¿Cuál es la clase general o el grupo al que pertenecen?
- ¿Qué distingue el cómo puedan utilizar el sistema?
- ¿Qué caracteriza su relación con la aplicación?
- ¿Qué necesitan de la aplicación?
- ¿Cómo se comportan en relación a la aplicación y cómo esperan que la aplicación se comporte?

Para identificar los roles se suele comenzar pensando en usuarios (agentes o grupos) particulares, reales, y después se va generalizando y abstrayendo hasta llegar al rol. [Constantine, 1999] propone la técnica de brainstorming, comentada en la etapa anterior, para, partiendo de una lista de roles candidatos, llegar al modelo de roles definitivo.

Del estudio de la información recogida y elaborada en la etapa anterior (elicitación de requisitos), se pueden extraer los diferentes roles que se necesitan en el sistema para que sean desempeñados por los usuarios (por los actores) finales que harán uso de la aplicación.

Los roles identificados se deben agrupar por roles similares con los objetivos de evitar tener roles demasiado parecidos que realmente pudieran ser uno solo, evitar duplicados, simplificar y generalizar el modelo, etc. [Constantine, 1999]. Es un paso de refinamiento.

Los roles se describirán empleando la plantilla de la Tabla 27, que muestra los metadatos utilizados en la descripción de los roles identificados. Con esta

información se obtienen los datos necesarios sobre los roles del sistema para el modelo de organización:

- Los tres primeros campos (**Versión**, **Autores**, **Fuentes**) son comunes a las plantillas de elicitación de requisitos y su significado es el mismo.
- Los roles pueden ser desempeñados por los actores si cumplen ciertas restricciones. Recordemos la definición de capacidad: *Una Capacidad es una habilidad o responsabilidad asociada a un actor que le permite desempeñar roles y llevar a cabo tareas*. Por lo tanto, para que algún actor pueda desempeñar un rol, éste debe tener unas determinadas **Habilidades**, y se le pueden exigir unas determinadas **Responsabilidades**, necesarias para que lo pueda hacer.
- Los actores que desempeñen un cierto rol, normalmente tendrán una serie de **Permisos** o privilegios para poder realizar ciertas acciones en el sistema.
- El metadato **Actor** permite identificar, de entre los actores del sistema, quiénes serán los que desempeñen este rol. Esto facilita la trazabilidad de la metodología.
- La Descripción esboza cómo es ese rol, mientras que los **Comentarios** dan algún tipo de información adicional.

Tabla 27. Plantilla para la descripción de los Roles del sistema: modelo organizativo

Rol- <id>	<nombre descriptivo>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) • ...
Fuentes	<ul style="list-style-type: none"> • <fuente de la versión actual> (<organización de la fuente>) • ...
Descripción	<descripción del rol>
Responsabilidades	<ul style="list-style-type: none"> • Las responsabilidades que se requieren de un actor para que pueda desempeñar este rol son las siguientes: • <Responsabilidad 1>: <descripción de la responsabilidad requerida> • ...
Habilidades	<ul style="list-style-type: none"> • Las habilidades que se requieren de un actor para que pueda desempeñar este rol son las siguientes: • <Habilidad 1>: <descripción de la responsabilidad requerida> • ...
Permisos	<normalmente se trata de los derechos que tiene el actor jugando este rol>
Actores	Los siguientes actores desempeñan este rol: <ul style="list-style-type: none"> • #{A-<id>, G- <id>, I-<id>, U-<id>, S-<id>} (<nombre del actor>) • ...
Comentarios	<comentarios adicionales sobre el rol>

Como se vio en el apartado 3.2, en [Constantine, 1999] se definen una serie de metadatos que podrían definir el rol con mayor precisión. Son metadatos más relacionados con la interacción del usuario en el entorno y no se han considerado en este trabajo para evitar que la definición y descripción de todos los elementos que se han de tener en cuenta no sea demasiado pesada.

Una vez que se ha identificado el conjunto de roles del sistema y se ha descrito, se pueden emplear para los siguientes pasos de la etapa de análisis, como se comentará posteriormente.

5.3.2.2 Identificación y Descripción de Tareas

El concepto de tarea se ha definido como *una porción de trabajo necesaria para lograr un objetivo determinado* (apartado 4.2.2.1). En el apartado 5.2.6.1 se ha discutido acerca del por qué los conceptos requisito y tarea son semánticamente equivalentes en este trabajo. Esto propicia la trazabilidad entre las etapas 1 y 2 del modelo de proceso, pero esto se comentará en el apartado 5.3.5.2.

Al iniciar el análisis del sistema se podría decir que siempre hay una tarea para resolver un requisito identificado. Existe una relación 1:1 entre los conceptos requisito y tarea, fruto de esa equivalencia semántica inicial. Siempre hay una tarea, aunque sea abstracta, que resuelve un requisito.

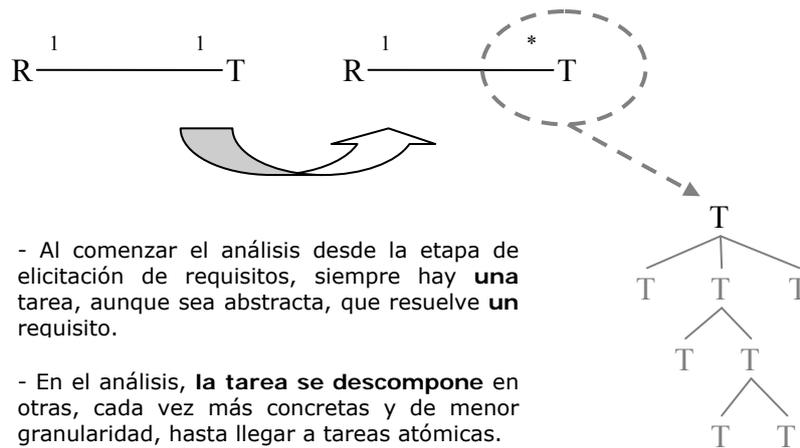


Figura 96. Relación entre requisitos del sistema y tareas de usuario en el análisis

En la etapa de análisis, algunas tareas se descomponen en otras y así sucesivamente hasta llegar a otras que son atómicas y no se pueden dividir más. Tal y como se expresa en la Figura 96, en esta etapa es probable que la relación entre requisitos y tareas ya no sea 1:1 sino más bien 1:M. Sí hay una tarea para resolver el requisito, solo que esta tarea es en realidad una tarea compuesta. Las tareas resultantes de esa descomposición surgen en esta etapa de análisis.

La Figura 97 muestra los pasos que se siguen en la identificación y descripción de las tareas en este trabajo, pasos que se describen a continuación.

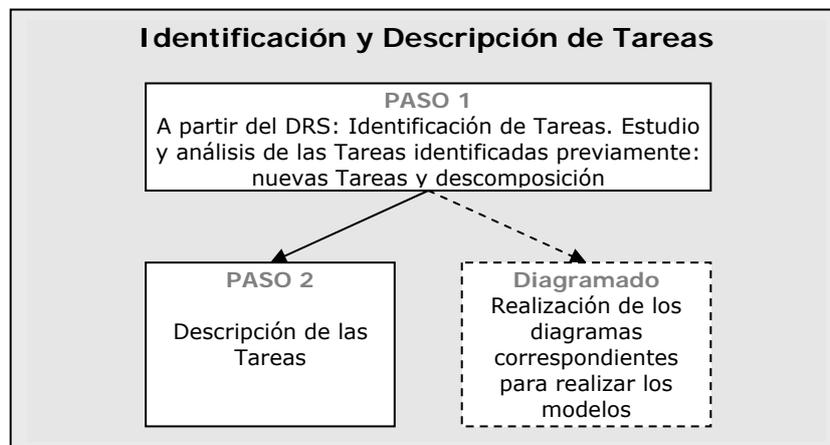


Figura 97. Pasos en la identificación y descripción de tareas

A partir del DRS de la etapa de elicitación de requisitos se han de identificar las tareas del sistema. El número de tareas identificadas se irá viendo incrementado con el estudio y el análisis de las mismas, principalmente por dos motivos: por descomposición de tareas complejas en otras más sencillas y por la identificación de nuevas tareas necesarias. Se trata de describir el trabajo que se debe realizar en el sistema. Para ello se pueden emplear diferentes aproximaciones: flujogramas, máquinas de estados finitos, modelos de workflow, diagramas de flujo de datos, etc. Como se ha comentado con anterioridad, la notación CTT para modelar tareas se ha utilizado extensamente y es el diagrama seleccionado en este trabajo para representar las tareas. Como se verá en los próximos subapartados, CTT permite modelar la interacción del usuario con el sistema, sin embargo la interacción persona-ordenador-persona, es decir, la interacción entre las personas a través de las máquinas no se modela con facilidad. Para solventar esta situación proponemos un nuevo diagrama que modela la colaboración entre usuarios. Así mismo, se describen las tareas por medio de plantillas similares a las descritas hasta ahora, y teniendo en cuenta la discusión establecida en torno al concepto de tarea en el apartado 4.2.2. La plantilla empleada puede verse en la Tabla 28.

Tabla 28. Plantilla para la descripción de las Tareas del sistema: modelo de tareas

Tarea- <id>	<nombre descriptivo>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) • ...
Fuentes	<ul style="list-style-type: none"> • <fuente de la versión actual> (<organización de la fuente>) • ...
Objetivo	La tarea está asociada al objetivo u objetivos siguientes: <ul style="list-style-type: none"> • #OBJ-<id> (<nombre del objetivo>)
Requisito	La tarea está asociada al requisito #{RI-<id>, RF-<id>, RNF- <id>} (<nombre del requisito>)
Descripción	<descripción de la tarea>
Tarea a la que pertenece	{La tarea es parte de la siguiente tarea: <ul style="list-style-type: none"> • #Tarea-<id>: <nombre descriptivo> , La tarea no es parte de ninguna otra}
Objetos	La tarea está relacionada o afecta a los siguientes objetos del sistema: <ul style="list-style-type: none"> • <nombre del objeto> • ...
Modo de Ejecución	Según su modo de ejecución la tarea es del tipo {Usuario, Interacción, Aplicación, Abstracta, otra}. <Descripción, si procede, del modo de ejecución>
Composición	Según su composición, la tarea es {Compuesta, Atómica}
Tarea Compuesta	
Tareas dependientes	La tarea está compuesta por las siguientes: <ul style="list-style-type: none"> • #Tarea-<id>: <nombre descriptivo> • ...
Descripción como Tarea Compuesta	<Información relevante como tarea Compuesta. Sólo si procede>
Tarea de Grupo	
CSCW	Según sus características CSCW, la tarea es de: <ul style="list-style-type: none"> • Coordinación: {Sí, No} • Comunicación: {Sí, No} • Cooperación: {Sí, No} • Colaboración: {Sí, No}
Tiempo	Según sus características temporales, la tarea es {Síncrona, Asíncrona}
Lugar	Según sus características espaciales, la tarea se lleva a cabo en {el mismo lugar, lugares diferentes}
Descripción como Tarea de Grupo	<Información relevante como tarea de grupo, probablemente según las características CSCW, Tiempo y Lugar. Sólo si procede>

Los campos que aparecen en la plantilla para la descripción de las tareas son los siguientes:

- Los tres primeros campos (**Versión**, **Autores**, **Fuentes**) son comunes a las plantillas de elicitación de requisitos y su significado es el mismo.
- **Objetivo** representa el subproblema o subproblemas asociados a la tarea. El conjunto de tareas asociadas a él, resuelven este objetivo del sistema, este sub-problema.
- **Requisito** muestra la relación de esta tarea con alguno de los requisitos identificados en la etapa anterior. La tarea sólo está asociada a un requisito. Las tareas que surgen como resultado de la división de otras según lo explicado anteriormente (Figura 96), tienen asociado el requisito de la tarea inmediatamente superior que lo tenga.
- **Descripción** permite introducir los comentarios necesarios para saber qué hace la tarea
- **Tarea a la que Pertenece** en caso de que forme parte de otra.
- **Objetos**. Las tareas están relacionadas o afectan a algunos objetos del dominio del sistema que deben mostrarse de manera que el diseñador sea consciente de ello. Los objetos del sistema y sus relaciones se representan, como se verá posteriormente, por medio del clásico diagrama de clases de UML.
- **Modo de Ejecución** representa una de las caracterizaciones introducidas en el modelo. Las tareas pueden ser de *Usuario*, de *Interacción*, de *Aplicación* o *Abstractas*. Esta clasificación está abierta, por lo que es posible caracterizarla como *otra*. Sería el caso, por ejemplo de las tareas compuestas.
- **Composición** indica si la tarea es *compuesta* o *atómica*.

En caso de que la tarea sea compuesta existen dos metadatos adicionales:

- **Tareas Dependientes** enumera aquellas tareas que componen la que se describe.
- **Descripción como Tarea Compuesta** proporcionaría información adicional, en caso de que fuera necesaria, relativa a su composición.

Si además la tarea es de grupo, se deben incorporar a la descripción de la tarea los siguientes metadatos:

- **CSCW** permite caracterizar la tarea según las características típicas del CSCW que han sido comentadas en el modelo de tareas. Así, una tarea podría caracterizarse como de *coordinación*, *comunicación*, *colaboración* y/o *cooperación*.
- **Tiempo** es el metadato que permite caracterizar la tarea según su ejecución sea en tiempo real o no.
- **Lugar** es el metadato que permite caracterizar la tarea según se lleve a cabo en el mismo lugar o los actores estén distribuidos geográficamente.
- **Descripción como Tarea de Grupo** proporcionaría información adicional, en caso de que fuera necesaria, relativa al hecho de que sea tarea de grupo.

5.3.3 Estructura y Comportamiento

Sin duda, el lenguaje más utilizado para modelar la realidad es UML [OMG, 2005]. UML 2.0 hace una distinción entre los paquetes empleados según permitan describir la estructura o el comportamiento del sistema:

- Parte I – *Estructura*. Paquetes que permiten describir la parte estática, estructural del sistema. Capturan la organización física de los elementos en el sistema, cómo se relacionan [Pilone, 2005].
 - Classes: Diagrama de Clases ,
Objetos,
Paquetes
 - Components: Diagrama de Componentes
 - Composite Structures: Diagrama de Composición
 - Deployments: Diagrama de Despliegue

- Parte II – *Comportamiento*. Paquetes que permiten describir la parte dinámica, el comportamiento de los elementos en el sistema.
 - Actions
 - Activities: Diagrama de Actividades
 - Interactions: Diagrama de Comunicación,
Visión de Cto de Interacciones,
Secuencias,
Tiempos
 - State machines: Diagrama de Estados
 - Use Cases: Diagrama de Casos de Uso

Esta división entre estructura y comportamiento se ha utilizado tradicionalmente en Ingeniería del Software para el modelado del sistema en todo el ciclo de vida del software. La etapa de análisis del modelo de proceso presentado en este trabajo contempla un conjunto de diagramas estructurales y de comportamiento que permiten el análisis del sistema.

Algunos de estos diagramas son ampliamente utilizados y aceptados por los profesionales de la materia. Otros, sin embargo, se han desarrollado con el objeto de aumentar la expresividad de la especificación y conseguir que sea más completa, especialmente desde el punto de vista del usuario como parte de un grupo y de las interacciones que puedan darse entre ellos.

Los elementos y relaciones que forman estos diagramas han sido descritos con anterioridad en el apartado 4.2.1, puesto que lógicamente son elementos que forman parte del modelo conceptual presentado.

En la Tabla 29 se muestran de forma resumida los elementos organizativos que se emplean en los diagramas junto con su notación. Del mismo modo, la Tabla 30 resume las posibles relaciones entre elementos organizativos.

Tabla 29. Elementos organizativos

Elemento Organizativo	Descripción	Notación
Actor	Un <i>actor</i> es una o varias personas, u otro u otros sistemas externos, que interactúa con el sistema. El <i>actor</i> puede interactuar directamente con el sistema colaborativo llevando a cabo tareas individuales, o bien, puede interactuar con otros <i>actores</i> a través del sistema realizando de este modo tareas cooperativas.	 ACTOR_1
Group	Un <i>grupo</i> es un conjunto de <i>actores</i> , individuos o colectivos, que desempeñan <i>roles</i> para lograr un <i>objetivo</i> común. Dichos <i>objetivos</i> comunes no serían realizables sin tal colaboración.	 GROUP_1
Individuo	Un elemento <i>individuo</i> es un, y sólo un, <i>actor</i> que desempeña <i>roles</i> .	 Individual_1
Usuario	Un <i>usuario</i> es un elemento <i>individuo</i> humano.	 User_1
Agente	Un <i>agente</i> es un elemento <i>individuo</i> no humano.	 Agent_1
Rol	Un <i>rol</i> es el conjunto de <i>tareas</i> que puede desempeñar un <i>actor</i> . Un <i>actor</i> es lo que es debido al <i>rol</i> que desempeña en cada momento en el sistema.	 ROLE_1

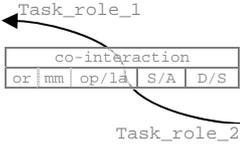
5.3.4 Diagramas Estructurales

Para representar la estructura del sistema se emplean dos diagramas estructurales.

En primer lugar el diagrama de clases correspondiente al *paquete Clases* de UML 2.0 [OMG, 2005] muestra los objetos del dominio que se utilizan en el sistema así como las relaciones que existen entre ellos. El diagrama de clases describe el conjunto de objetos del sistema en base a los atributos y operaciones que comparten. Esos objetos se representan mediante clases que abstraen los conceptos y que están relacionadas entre sí representando las asociaciones que se dan entre los objetos del dominio del problema. Estas clases se pueden identificar a partir de los RI (requisitos de información) y de los elementos de la estructura organizativa.

Para describir la estructura organizativa de los actores que participan en un sistema CSCW se ha desarrollado un diagrama que la representa: OSD. Es un diagrama estructural puesto que no especifica comportamiento, sino los participantes del sistema, su organización y los roles que desempeñan.

Tabla 30. Relaciones entre los elementos organizativos

Elemento Organizativo	Descripción	Notación
Relación de Desempeño	Una <i>relación de desempeño</i> es una <i>relación organizativa estructural</i> que identifica el <i>rol</i> que juega un <i>actor</i> .	
Relación de Agregación	Una <i>relación de agregación</i> es una <i>relación organizativa estructural</i> que identifica una asociación entre el todo y las partes.	
Relación de Jerarquía	Una <i>relación de jerarquía</i> es una <i>relación organizativa estructural</i> que identifica una dependencia de grado entre dos <i>actores</i> .	
Co-interacción	La co-interacción es una <i>relación organizativa de grupo</i> entre dos <i>actores</i> que expresa una interacción entre ellos, encaminada al logro de un <i>objetivo</i> común que no podría alcanzarse sin dicha interacción.	

En este trabajo no se describirá en profundidad el diagrama de clases. La descripción completa del mismo puede obtenerse en la fuente original [OMG, 2005] así como en otros muchos documentos que han aparecido con posterioridad debido a su uso extensivo. Sin embargo, el diagrama OSD sí se detalla a continuación.

La aplicación del caso de estudio (Capítulo 6) muestra ejemplos de aplicación de estos diagramas.

5.3.4.1 Diagrama para la Estructura Organizativa (OSD)

El diagrama para la estructura organizativa de los actores de un sistema (Organizational Structure Diagram, OSD) representa la distribución de los diferentes elementos organizativos del mismo, es decir, cómo se organizan los actores del sistema y cómo se relacionan (en términos estructurales) para constituir los grupos y jerarquías, qué roles desempeñan, etc.

El OSD se emplea para representar la estructura organizativa de todo el sistema. El sistema se construye con una finalidad y para llegar a alcanzar ese fin último, los actores del mismo han de organizarse de una determinada manera. Hay una estructura de grupos, actores desempeñando roles, relaciones de pertenencia, de jerarquía, etc. Todo esto es lo que representa el OSD, por lo que en principio habrá un OSD por sistema.

Si el sistema fuera demasiado complejo se podrían realizar diversos OSD complementarios, representando, por ejemplo, los subsistemas en los que se ha particionado el sistema principal en la etapa de elicitación de requisitos: objetivos del sistema (apartado 5.2.7).

El OSD es una representación del modelo de organización (apartado 4.2.1).

Puesto que el OSD es una representación del modelo de organización, los elementos organizativos y las relaciones entre ellos que se emplean para modelar esa parte del dominio del problema han sido descritos en el apartado 4.2.1. y están resumidos en la Tabla 29 (elementos organizativos) y en la Tabla 30 (relaciones).

En un OSD, podría aparecer cualquiera de los elementos organizativos descritos, sin embargo, en sucesivas iteraciones, elementos más abstractos como actor o individuo se irán sustituyendo por elementos más concretos como grupo, usuario o agente. En cualquier caso se muestra también el rol que juegan cada uno en el sistema.

En cuanto a las relaciones que se dan entre los elementos organizativos de un OSD, podrían aparecer relaciones de jerarquía, agregación o desempeño.

5.3.5 Diagramas de Comportamiento

Los diagramas de comportamiento de UML que podrían emplearse para el análisis de la dinámica de los elementos del sistema son el paquete Activities (diagrama de actividades), y los diagramas de comunicación y de secuencias del paquete Interactions.

Se ha definido un nuevo diagrama, el diagrama de co-interacciones, que esboza las colaboraciones que tienen lugar entre actores del sistema. Así mismo, el diagrama de tareas es un diagrama que se considera fundamental para representar lo que el usuario hace, las interacciones con el sistema. Como TD se ha adoptado la notación ConcurTaskTrees [Paternò, 1999] por ser ampliamente aceptada. Sin embargo, otras notaciones para modelar tareas [Pinelle, 2003; Van der Veer, 2000; etc.] podrían adaptarse igualmente.

5.3.5.1 Diagrama de Tareas (TD)

El diagrama de tareas permite modelar la parte de la realidad, del dominio del problema, que tiene que ver con el trabajo que realizan los actores en un sistema. Concretamente tiene en cuenta las tareas que realiza un actor del sistema de manera individual, su interacción con el sistema.

Como se ha comentado con anterioridad, existen diferentes aproximaciones al modelado del trabajo que realizan los usuarios de un sistema. Hemos adoptado CTT como notación para modelar las tareas que un actor realiza por ser ampliamente aceptado en la comunidad científica, pero otros modelos también podrían adaptarse.

Los tipos de tarea y los operadores que se pueden dar entre ellas en CTT ya han sido comentados con anterioridad (ver apartado 3.5.2). El orden entre las tareas y cómo se pasa de una a otra (operadores) se definen en esta etapa y con este diagrama.

5.3.5.2 Diagrama de Co-interacciones (CD)

El diagrama de co-interacciones (Co-Interactions Diagram, CD) representa las colaboraciones establecidas entre los diferentes elementos organizativos de una estructura organizativa, es decir, las interacciones que existen entre los actores (usuarios, agentes, grupos) de un sistema a través del mismo: tareas de grupo.

Estos diagramas se pueden emplear para representar las colaboraciones de todo el sistema o de parte de él. Si el sistema no es muy complejo, el CD puede mostrar todas las colaboraciones que se dan en él, conociendo de un vistazo cómo interactúan todos sus actores entre sí. Si por el contrario, el sistema es muy complejo, se pueden utilizar para representar partes de él. En cualquier caso, la intersección de todos los diagramas que representan esos subsistemas darían lugar a la

representación de las colaboraciones que se dan en el sistema en su totalidad. Esto siempre que se hayan realizado diagramas para todos los subsistemas.

Lo más adecuado, normalmente, es generar varios CDs por OSD que modelan la interacción persona-ordenador-persona entre los actores de esa estructura representada por el OSD.

Los elementos empleados para formar un CD son los mismos que los que se han utilizado para representar la estructura organizativa por medio de los OSD. Esto es lógico puesto que lo que se representa mediante un CD son las colaboraciones que tienen lugar entre los participantes de un sistema organizados como se muestra en el OSD.

Dichos elementos han sido descritos en el apartado 4.2.1 y están resumidos en la Tabla 29 (elementos organizativos). En cuanto a las relaciones posibles entre los elementos, aunque es factible mostrar algunas relaciones de jerarquía, agregación o desempeño por claridad, lo normal es que aparezcan relaciones de co-interacción que describe esa interacción entre dos participantes a través del sistema (ver apartado 4.2.1 y Tabla 30).

5.3.6 Trazabilidad Inter- e Intraetapa

Es fundamental mantener la coherencia y la consistencia entre los elementos que aparecen entre distintas etapas, así como en el interior de cada etapa.

Como se ha comentado con anterioridad, las matrices de rastreabilidad permiten relacionar diferentes conceptos de forma explícita, a pesar de que pueden estar ya relacionados de forma implícita en la especificación del sistema.

Para mantener la trazabilidad en el modelado del sistema, normalmente se hará uso de este tipo de matrices que relacionan algunos conceptos entre etapas diferentes, así como conceptos que aparecen en pasos diferentes de la propia etapa.

Mediante el uso de una herramienta CASE que soportara el modelo de proceso y la metodología que se presenta en el trabajo, proporcionaría un modo automático de visualizar toda esta información. Las matrices se generarían de forma automática a partir de la información de las etapas.

5.3.6.1 Trazabilidad entre las Etapas de Elicitación de Requisitos y Análisis

Para mantener la trazabilidad y la coherencia entre las etapas de elicitación de requisitos y análisis, se establecen dos matrices de rastreabilidad que ponen en relación explícita los actores y los requisitos identificados y descritos en la etapa de elicitación de requisitos con los roles y tareas identificados y descritos en la etapa de análisis respectivamente.

Matriz actores-roles

En el primer caso, la matriz permite identificar clara e inequívocamente qué puede hacer quién en el sistema. Se especifica qué *roles* desempeña cada *actor*.

Un actor puede desempeñar varios roles en el sistema, cada uno de los cuales le permitirá realizar unas funciones determinadas. En el otro sentido, un rol puede estar desempeñado por diferentes actores del sistema.

Tabla 31. Esquema de matriz actores-roles

	Rol- <i><id1></i> (nombre)	Rol- <i><id2></i> (nombre)	...	Rol- <i><idm></i> (nombre)
Actor- <i><id1></i> (nombre)	•	•		
Actor- <i><id2></i> (nombre)				•
...			...	
Actor- <i><idn></i> (nombre)		•		

Matriz tareas-requisitos

Por otro lado, la otra matriz pone en relación los requisitos identificados en la etapa de elicitación de requisitos con las tareas identificadas en el análisis que describen cómo se lleva a cabo en el sistema una porción de trabajo encaminada a resolver un determinado requisito.

En la etapa de análisis se ha comprobado que un requisito es alcanzado por la ejecución de una única tarea, pero esta tarea podría descomponerse en otras.

Por tanto, para un mismo requisito podría haber diferentes tareas asociadas. Una, la que lo resuelve, más todas las descendientes que la componen.

Cabe la posibilidad de que un bloque de tareas se repita para resolver una situación concreta que se dé en más de una ocasión. Un ejemplo podría ser la comprobación de que un usuario pertenece al sistema con una serie de permisos antes de realizar una determinada acción. Esto se podría repetir en diferentes requisitos, por lo que es posible encontrar una tarea en varios requisitos. Probablemente un requisito estaría incluido en el otro.

Tabla 32. Esquema de matriz tareas-requisitos

	Req- <i><id1></i> (nombre)	Req- <i><id2></i> (nombre)	...	Req- <i><idm></i> (nombre)
Tarea- <i><id1></i> (nombre)	•	•		
Tarea- <i><id2></i> (nombre)				•
...			...	
Tarea- <i><idn></i> (nombre)		•		

5.3.6.2 Trazabilidad Estructura-Comportamiento en la Etapa de Análisis

La trazabilidad para mantener la coherencia y la consistencia del modelo se ha de dar entre roles y tareas puesto que son conceptos muy relacionados pero que se identifican en un principio separadamente. Esta matriz les da la unidad que necesitan para tener sentido.

Así mismo, una vez identificados y descritos los roles y las tareas del sistema, los diagramas comentados con anterioridad permiten modelar la estructura organizativa de los actores del sistema, las colaboraciones que existen entre ellos y las interacciones de los usuarios con el sistema. Muchos de los elementos que se emplean en un diagrama se emplean también en los otros diagramas.

En principio, el uso de elementos unívocamente identificados en los distintos diagramas debería mantener la coherencia y la consistencia del modelo. Se debe tener especial cuidado con las implicaciones que tendría la modificación de un elemento en un modelo con respecto a los otros, que probablemente se verían afectados.

Matriz tareas-roles

El rol se ha definido en base a un conjunto de tareas, por lo que roles y tareas están íntimamente relacionados. El número de roles y, sobre todo, de tareas de un sistema puede ser muy elevado. Una matriz que ponga en relación los roles con las tareas asociadas, es decir, el rol identificado con la funcionalidad que le proporcionaría al actor que lo desempeñara, facilita la comprensión del sistema.

La *matriz tareas-roles* asocia el conjunto de tareas que define cada rol. El rol es la selección de un conjunto de tareas, por lo que éstas podrían estar asociadas también a varios roles.

También se habrá de tener en cuenta lo siguiente: si un actor y un requisito quedan relacionados en la etapa de elicitación de requisitos (Tabla 17), un rol y una tarea pueden quedar asociados en esta etapa; en caso contrario tal asociación no puede tener lugar.

Tabla 33. Esquema de matriz tareas-roles

	Rol-<id1> (nombre)	Rol-<id2> (nombre)	...	Rol-<idm> (nombre)
Tarea -<id1> (nombre)	•	•		
Tarea -<id2> (nombre)				•
...			...	
Tarea -<idn> (nombre)		•		

5.4 Etapa 3.- Diseño

Tras el análisis del sistema se conocen qué tareas se han de realizar en el sistema, quién las hace, de qué manera, cuál es la organización de los usuarios del sistema, qué roles desempeñan, cuáles son las tareas de grupo que implicarán colaboración, coordinación, comunicación, etc.

El proceso siguiente, el diseño del sistema, consiste en abordar la manera de cómo presentar la información (visualización, entradas, controles, etc.) al usuario. Se trata de un proceso en el que se traducirá toda la información obtenida hasta ahora de las etapas previas a una representación del software que se implementará.

El diseño todavía no proporciona detalles de implementación como la plataforma en la que mostrará la aplicación, etc. Simplemente se muestran los elementos que formarán parte de las UIs: son los AIOs definidos con anterioridad (apartado 3.7.2.1) y que se pueden ver desde TRIDENT [Bodart, 1990] hasta proyectos tan relevantes para el desarrollo de UIs en HCI como Cameleon [Calvary, 2003].

Una de las definiciones clásicas de *diseño* dice que es *el proceso de aplicar distintas técnicas y principios con propósito de definir un dispositivo, proceso o sistema con los suficientes detalles como para permitir su realización física* [Taylor, 1959].

El diseño de un sistema se puede abordar desde:

- Diseño de la estructura interna de los *datos*
- Diseño arquitectónico
- Diseño *procedimental* de los componentes software
- Diseño de la interfaz de usuario: establece la disposición y los mecanismos para la interacción hombre-máquina.

La etapa de diseño en las metodologías se corresponde con el estudio del dominio de la solución. Esta etapa trata de contestar al *cómo*, integrando los requisitos no funcionales y describiendo asuntos de implementación.

En el diseño presentado en esta tesis se aborda el desarrollo de la UI para entornos CSCW a partir de los trabajos realizados en HCI. La relación entre los distintos puntos desde los que se ha de abordar el diseño de un sistema es muy íntima, por lo que se tienen en cuenta, aunque con nivel de detalle mucho menor.

UML 2.0 [OMG, 2005] ofrece una serie de diagramas para facilitar el modelado del diseño de un sistema desde su estructura y desde su comportamiento:

- Estructura: Diagrama de Clases, Objetos, Diagrama de Componentes, Diagrama de Composición y Diagrama de Despliegue.
- Comportamiento: Diagrama de Actividades, Diagrama de Comunicación, Visión de Cto de Interacciones, Secuencias, Tiempos y Diagrama de Estados.

Como se ha discutido en el apartado 3.7, han habido muchas aproximaciones al diseño de la interfaz de usuario en el campo de la interacción persona-ordenador. Sin embargo, no son tan numerosas las aproximaciones realizadas que hayan tenido en cuenta la filosofía CSCW que rigen las aplicaciones groupware.

CSCW se considera un campo de investigación parte de HCI. En general, las aproximaciones HCI son válidas para UIs de aplicaciones groupware, pero la particularización de algunos aspectos metodológicos en el diseño de UIs para sistemas CSCW haría tener en cuenta esas particularidades desde el principio, con lo que aumentaría la calidad del desarrollo final.

En el diseño de un sistema CSCW se habrían de tener en cuenta especialmente sus aspectos relativos a la *comunicación, coordinación, cooperación y colaboración*, así como aspectos *espacio-temporales* que son definitivos en algunos casos. Así mismo se han de tener en cuenta aspectos relativos a la *percepción (awareness)* que el usuario tiene sobre lo que el resto de usuarios del sistema hacen. Este último aspecto resulta ser un requisito no funcional fundamental para el éxito de una aplicación groupware, porque aumenta la confianza del usuario en el sistema y en el trabajo en equipo.

Dos factores que pueden diferenciar el desarrollo de sistemas CSCW mediante metodologías dedicadas son: patrones de uso y elementos de implementación específicos.

Los *patrones* determinan la manera en la que se suelen realizar determinados procesos de grupo que suelen repetirse. Son una generalización que abstrae los conceptos básicos y sugiere realizar el proceso de una determinada manera estudiada. De este modo, el proceso de grupo en cuestión es un proceso ya estudiado y la manera de resolverlo no necesita ser reestudiada, sino que se hace uso de la reutilización del conocimiento que se tiene sobre la materia. Los patrones están más cerca del diseño procedimental del sistema, por lo que no se estudian en detalle en este trabajo.

Los *elementos de implementación* específicos que se empleen en el desarrollo de aplicaciones groupware pueden mejorar considerablemente la experiencia del usuario: uso de telepunteros, por ejemplo. Estos elementos están íntimamente relacionados con la interfaz de usuario por lo que se considerarán en detalle tanto en esta etapa de diseño como en la siguiente de implementación.

5.4.1 Diseño de la Interfaz de Usuario en Entornos CSCW

La mayor parte de la filosofía de diseño de aplicaciones interactivas subyacente en el campo de la interacción persona-ordenador se puede aplicar al diseño de aplicaciones groupware.

De hecho este tipo de aplicaciones son fuertemente interactivas, con el añadido de la interacción existente entre los usuarios del sistema a través de las máquinas. Es decir, al fin y al cabo, para que los usuarios se comuniquen entre sí, colaboren, cooperen, se coordinen, lo hagan en tiempo real o de forma asíncrona, hacen uso de UIs interactivas por medio de las que realizan esas tareas de grupo.

La diferencia fundamental entre los dos tipos de aplicaciones radica en el modo en el que trabajan los usuarios. En aplicaciones interactivas sin más, los usuarios no necesitan tener en cuenta a otros miembros del sistema. Sin embargo, en aplicaciones groupware los usuarios deben tener presente la interacción del resto de usuarios.

En el apartado anterior se define el diseño del sistema como una etapa que *trata de contestar al cómo, integrando los requisitos no funcionales y describiendo asuntos de implementación*. La clave para realizar un diseño apropiado en aplicaciones groupware es tener en cuenta los requisitos no funcionales adecuados. Concretamente el requisito no funcional por excelencia que determina el éxito de una aplicación colaborativa es el **awareness**, concepto discutido en el apartado 3.6.

Un adecuado tratamiento del *workspace awareness* en la metodología seguida para el diseño de la interfaz de usuario en sistemas HCI da soporte al diseño de la interfaz de usuario de los sistemas CSCW.

La Figura 98 muestra los elementos que se han de tener en cuenta en los métodos HCI para el diseño de la interfaz de usuario de los sistemas CSCW. Estos elementos posibilitan el awareness en un contexto compartido: workspace awareness.

Los cuatro puntos clave del concepto de *awareness* son: (1) es el *conocimiento* que un usuario tiene sobre lo que los demás hacen en el sistema, (2) es un *requisito no funcional* que (3) *cambia* con el paso del tiempo y (4) los usuarios lo *mantienen por medio de interacciones* con el sistema.

Estos puntos se aplican sobre los *objetos*, y las *acciones* que se pueden realizar sobre ellos, del espacio común de trabajo, es decir, sobre un *contexto compartido*.

La noción que se tiene hasta el momento actual (*pasado o historia y presente*) sobre la interacción de otras personas en el espacio de trabajo común define el *workspace awareness*, que es el requisito no funcional que se busca cubrir en los modelos de proceso de diseño de interfaces de usuario en HCI para realizar con éxito el diseño de la interfaz de usuario de los sistemas CSCW.

Las técnicas empleadas para posibilitar el workspace awareness han de tener presentes los elementos históricos y presentes que se muestran en la Figura 98 [Gutwin, 1997; Gutwin, 2004] relativos a quién hace qué, dónde lo hace, cómo, etc.

Estas técnicas computacionales están clasificadas en tres categorías: *embodiment* (personificación), *expressive artifact* (artefactos) y *visibility* (visibilidad), tal y como se explicó en el apartado 3.6.

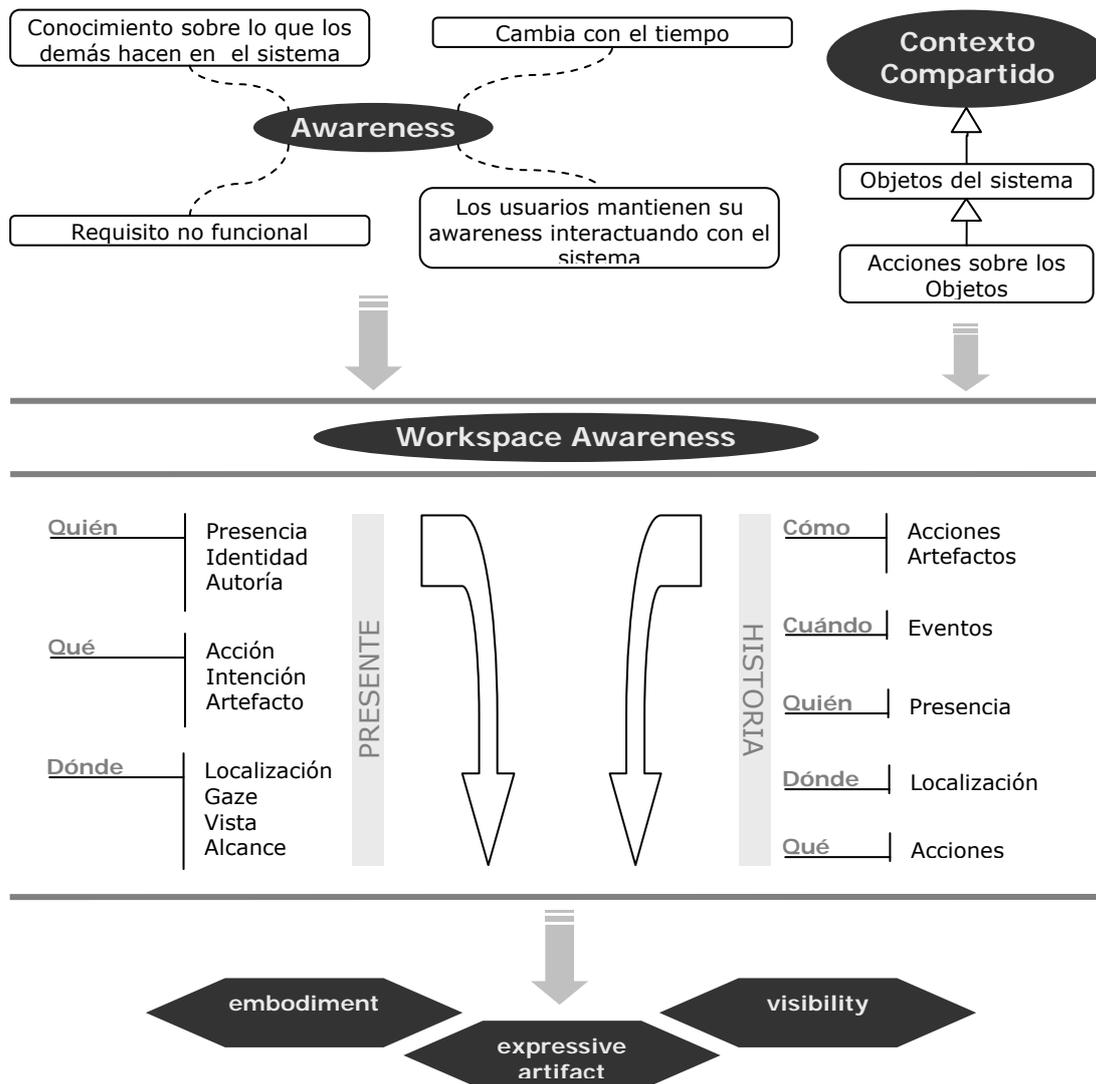


Figura 98. Elementos a tener en cuenta en los métodos HCI para el diseño de la interfaz de usuario de los sistemas CSCW

5.4.2 Metodología del Diseño

Como se ha comentado con anterioridad, la metodología que se sigue en esta etapa se hereda de algunos métodos seguidos en HCI. Adicionalmente se tendrán en cuenta los elementos comentados en el apartado anterior referentes a lo que es necesario considerar para obtener el diseño de la interfaz de usuario de los sistemas CSCW.

Siguiendo con el proceso típico propuesto en el marco de referencia Cameleon para el desarrollo de aplicaciones interactivas se trata de diseñar un AUI, una interfaz de usuario abstracta como proceso de reificación a partir de las tareas y demás conceptos abordados en la etapa de análisis. Para ello se han de identificar los AIOs correspondientes, que a su vez son una abstracción de los CIOs que concretarían la interfaz de usuario (CUI) en la fase de implementación.

Es decir, mientras que en la fase de diseño se identifican los AIOs independientes de la modalidad (interacción gráfica, vocal, reconocimiento y síntesis del habla, interacción basada en video, virtual, realidad mixta o aumentada) y de la plataforma (tanto software como hardware: sistema operativo y dispositivo), en la fase de implementación se reifican los CIOs a partir de esos AIOs. Estos CIOs son todavía independientes de la plataforma.

Se tomará como base de esquema conceptual el de UsiXML para el modelado de la UI abstracta (ver Figura 44) [Limbourg, 2004] puesto que la incorporación de las facetas es una de las últimas evoluciones en el diseño de la UI de sistemas interactivos como se puede apreciar en varios trabajos [Vanderdonckt, 2005; Jaquero, 2005; Montero, 2005].

Este modelo conceptual se enriquece como aparece en la Figura 99 para dar soporte a estas nuevas características:

- *AWAC*. Un Abstract Container (AC) podría ser además un *AWAC* o *Abstract Workspace Awareness Container* si se trata de un contenedor que facilitará un contexto compartido donde el usuario interactuará y mediante el que obtendrá awareness del resto de usuarios del sistema, es decir, es algo más que una simple ventana, cuadro de diálogo, marco, etc. Es una instancia que probablemente reflejará un cambio realizado en ella en el resto de instancias remotas.
- *Embodiment, expressive artifact* y *visibility* son las técnicas computacionales que darán soporte al awareness de alguna manera teniendo en cuenta los elementos presentes e históricos previamente comentados. En esta etapa de diseño se representan como facetas adicionales de los AICs (Abstract Individual Components). Es decir, habrá nuevos AICs que con estas facetas den soporte al awareness en el sistema. Por ejemplo un telepuntero es una reificación de un AIC que representa los usuarios que están interactuando en un AWAC.

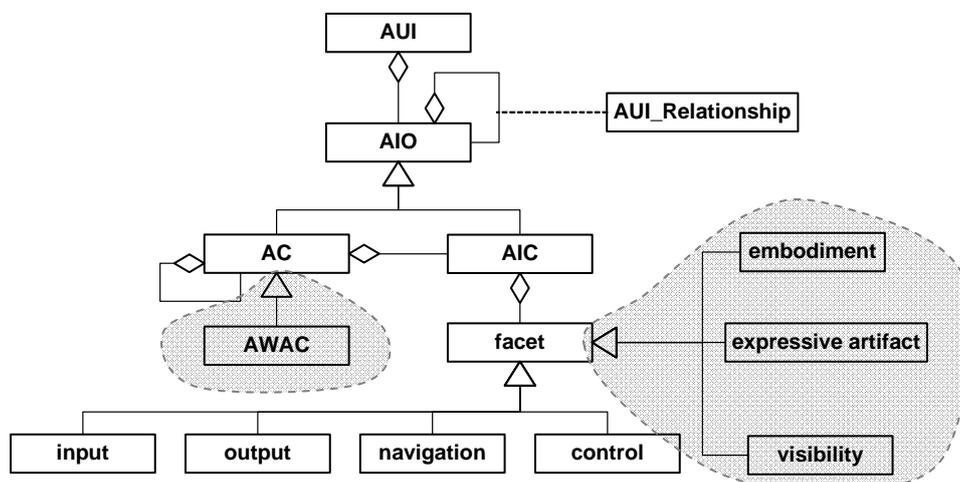


Figura 99. Modelo conceptual basado en el modelo conceptual para interfaces de usuario abstractas de UsiXML para el modelado de la UI abstracta de los sistemas CSCW. Las nuevas inclusiones aparecen sombreadas.

La Tabla 34 muestra la notación empleada para representar cada uno de los conceptos en los diagramas de los modelos como se describirá posteriormente. La notación está basada en la de [Montero, 2005], extendiéndola con los nuevos conceptos.

Tabla 34. Notación para los elementos en el diseño de la UI de aplicaciones groupware

AIO	Descripción	Notación
AC	Un <i>Abstract Container (AC)</i> es un AIO, una entidad que permite la agrupación lógica de otros ACs u otros AICs.	 Nombre_AC
AWAC	Un <i>Abstract Workspace Awareness Container (AWAC)</i> es un AC, un contenedor que facilitará un contexto compartido donde el usuario interactuará y mediante el que obtendrá awareness del resto de usuarios del sistema.	 Nombre_AWAC
AIC	Un <i>Abstract Individual Component (AIC)</i> es un AIO, una abstracción que representa un objeto de interacción.	 Nombre_AIC
input	<i>Input</i> es una faceta que describe una acción de entrada soportada por un AIC.	 Nombre
output	<i>Output</i> es una faceta que describe qué datos se pueden presentar al usuario por medio de un AIC.	 Nombre
navigation	<i>Navigation</i> es una faceta que describe una posible transición que un AIC puede habilitar.	 Nombre
control	<i>Control</i> es una faceta que describe enlaces entre un AIC y funciones del sistema, como por ejemplo, métodos del modelo de dominio.	 Nombre
embodiment	<i>Embodiment</i> es una faceta que describe la presencia de un usuario en un contexto compartido.	 Nombre
expressive artifact	<i>Expressive artifact</i> es una faceta que describe acciones de otros usuarios sobre un objeto del sistema. Suele mostrar las acciones sobre el propio AIC.	 Nombre
visibility	<i>Visibility</i> es una faceta que describe acciones de otros usuarios en el sistema. Suele mostrar las acciones en un AWAC aparte.	 Nombre

Un Modelo de Navegación basado en IDEAS (ver apartado 3.7.2.2) y adaptado a las nuevas necesidades completa la metodología de esta etapa proporcionando una representación de la navegación de los usuarios por las distintas UIs del sistema.

La Figura 100 muestra un esquema de la metodología descrita para esta etapa.

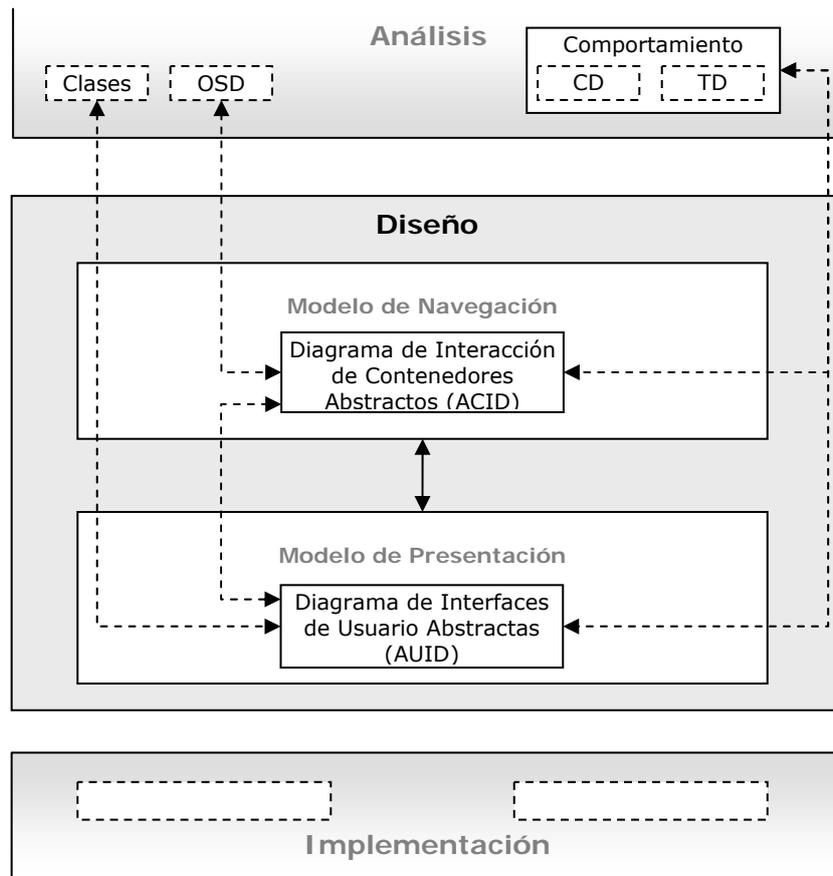


Figura 100. Pasos de la etapa de Diseño

5.4.3 Modelo de Navegación

El Modelo de Navegación proporciona una representación del camino que toman los usuarios entre las distintas interfaces en función de sus acciones. Para su representación se ha tomado como base el Diagrama de Interacción de Diálogos de IDEAS (ver apartado 3.7.2.2) adaptado al nuevo modelo conceptual para el diseño de UIs para sistemas CSCW. El diagrama resultante se conoce como *Diagrama de Interacción de Contenedores Abstractos (ACID, Abstract Container Interaction Diagram)*.

Debido a esta adaptación la idea de diálogo se extiende a la de AC (Abstract Container). Es decir, ya no se habla de ventana o diálogo como las "pantallas" por las que navega el usuario, sino de ACs o AWACs en su caso. De esta manera (1) se generaliza el concepto y (2) se permite la representación de navegaciones internas como las que proporcionan tecnologías como AJAX (Asynchronous JavaScript And XML, técnica de desarrollo Web para crear aplicaciones interactivas). En este sentido el Diagrama de Estados Internos propuesto en IDEAS no sería necesario puesto que el mismo Diagrama de Interacción de ACs podría representarlo.

5.4.3.1 Diagrama de Interacción de Contenedores Abstractos (ACID)

La representación de una navegación por medio de un ACID comienza con un *nodo inicial* y termina por un *nodo final*. Si se utiliza un ACID para representar navegaciones intermedias, los nodos iniciales y finales podrían ser ACs.

Los elementos que se emplean para formar el ACID son los que aparecen en la Tabla 34. Principalmente se emplean ACs y AWACs puesto que son abstracciones de lo que finalmente serán las UIs, pero se pueden emplear otros elementos como un AIC con una faceta *navigation* con el objeto de clarificar un esquema.

Las transiciones entre ACs y AWACs se realizan mediante arcos dirigidos nominados. El nombre en cuestión es la tarea que provoca el paso de un AC a otro. Los ACs contienen adicionalmente en su interior los iconos de los roles que pueden visualizar esos ACs.

Ver ejemplo en el caso de estudio del apartado 6.4, Figura 115.

5.4.4 Modelo de Presentación

El Modelo de Presentación proporciona una representación de las UIs con las que interaccionan los usuarios. Para su representación se ha tomado como base la notación empleada en [Montero, 2005] para representar los elementos del modelo conceptual de UsiXML. Puesto que dicho modelo se ha extendido para dar soporte a la representación de las UIs de entornos CSCW, también la notación se extiende.

El diagrama resultante se conoce como *Diagrama de Interfaces de Usuario Abstractas (AUID, Abstract User Interface Diagram)*. Este diagrama representa ACs que contienen AWACs y/o AICs. Lógicamente los AWACs también contienen AICs. De hecho como ACs que son, se pueden representar por medio de estos diagramas.

5.4.4.1 Diagrama de Interfaces de Usuario Abstractas (AUI)

Cada UI abstracta se representa por la composición de los elementos de la Tabla 34. Al menos hay un AC puesto que es el elemento que abstrae objetos como una ventana, un cuadro de diálogo, etc. en una implementación posterior.

Cada AC puede contener a su vez otros ACs y otros AICs. Los AWACs representan espacios de trabajo compartidos donde los usuarios interaccionan entre sí, obtienen información de lo que los demás usuarios están haciendo en el sistema, etc.

Los AICs son elementos con los que el usuario interactúa. Estos AICs pueden estar caracterizados por una o varias facetas que especifican qué hacen exactamente. Las facetas especificadas en [Limbourg, 2004] caracterizan los AICs según permitan al usuario introducir información en el sistema (*input*), visualizar información del sistema (*output*), navegar por el sistema (*navigation*) o lanzar alguna tarea (*control*).

Adicionalmente se añaden **tres nuevas facetas** según las técnicas computacionales descritas con anterioridad que facilitan el awareness en un sistema. Los AICs caracterizados con estas facetas aparecen en AWACs:

- Los AICs marcados con una faceta de **personificación** (*embodiment*) representan la presencia de un usuario en el sistema que interactúa con él. Son abstracciones de telepunteros, avatares, imágenes de video, etc.
- Los AICs marcados con una faceta de **artefectos expresivos** (*expressive artifact*) representan acciones que otros usuarios están realizando sobre esos

AICs. De esta manera el usuario es consciente de estas acciones. Por ejemplo, sería una abstracción de un botón que cambia de color en la interfaz de usuario cuando otro usuario lo pulsa desde otra interfaz remota.

- Los AICs marcados con una faceta de **visibilidad** (*visibility*) representan elementos que muestran en un espacio aparte las acciones que otros usuarios puedan estar realizando en el sistema. Por ejemplo, sería una abstracción de una vista de radar que mostrara los movimientos de otros usuarios por una determinada zona del sistema.

También aparecen en los AWACs los AICs con facetas relacionadas con tareas de grupo, como por ejemplo un AIC con una faceta de *control* que describa una tarea de este tipo.

Los AIOs manipulan objetos del dominio del sistema. En IDEAS la relación entre los objetos de interfaz y objetos del dominio se resuelve por medio de la idea de contrato (apartado 3.7.2.2). Por simplificar el uso de diagramas para modelar esta relación se incluye entre paréntesis el nombre del objeto u objetos manipulados por los AIOs junto al nombre del AIO.

Ver ejemplo en el caso de estudio del apartado 6.4, Figura 116 y Figura 117.

5.4.5 Trazabilidad Inter- e Intraetapa

Como se ha comentado con anterioridad en la etapa de análisis, es fundamental mantener la coherencia y la consistencia entre los elementos que aparecen entre distintas etapas, así como en el interior de cada etapa.

La trazabilidad en esta etapa está garantizada de la siguiente manera:

- *Intra-etapa*. El uso de los mismos elementos en los modelos de Navegación y de Presentación facilita la trazabilidad dentro de la etapa manteniendo la coherencia y consistencia del modelo.
- *Inter-etapa*. Por otro lado, los elementos de la etapa anterior que tienen presencia en la etapa actual son:
 - Los *objetos del dominio* del sistema manipulados por los AIOs correspondientes. Éstos aparecen junto al nombre del AIO que los manipula en el AUID del modelo de presentación.
 - Los actores y los roles del sistema tienen su presencia en el ACID del Modelo de Navegación, puesto que los ACs contienen los actores y roles que pueden visualizarlos.
 - Las tareas, incluidas las tareas de grupo, están representadas en el ACID del Modelo de Navegación, necesarias para pasar de un UI a otro; así como en el AUID del Modelo de Presentación donde AICs con facetas como las de *control* representan precisamente esas tareas.

5.5 Etapa 4.- Implementación

La etapa de implementación del sistema consiste en generar la UI final (FUI) del usuario en base a los AIOs definidos en la etapa de diseño. Es un proceso de reificación de cada uno de dichos componentes que concretizarán los elementos a emplear según detalles de implementación, plataforma objetivo, personalizaciones de usuarios, etc.

La diversidad de dispositivos, sistemas operativos, etc. hacen que un mismo AUI (Abstract User Interface) se pueda presentar a un usuario de formas muy diversas. La idea de qué es lo que hay que mostrar y cómo hacerlo es lo que se ha estudiado en las etapas anteriores. Ahora, los detalles concretos determinarán diferentes representaciones.

Debido al desarrollo que se ha seguido en la metodología y las decisiones tomadas especialmente en la etapa de diseño, IDEAS [Lozano, 2001], UsiXML [Limbourg, 2004] y el marco de referencia Cameleon [Calvary, 2003], así como los trabajos mencionados en dicha etapa de diseño, han influenciado directamente esta etapa de implementación de UIs para entornos CSCW.

El trabajo presentado en esta tesis se centra en el desarrollo de interfaces de usuario para entornos de trabajo en grupo y no está centrada en que se puedan generar de manera independiente a la plataforma por lo que no se ha considerado una representación de la CUI, que por otro lado es muy similar, y se realiza el proceso en un mismo paso. Por tanto, la etapa de implementación consiste fundamentalmente en la traducción de AIOs de la etapa de diseño a CIOs y su uso según el toolkit seleccionado.

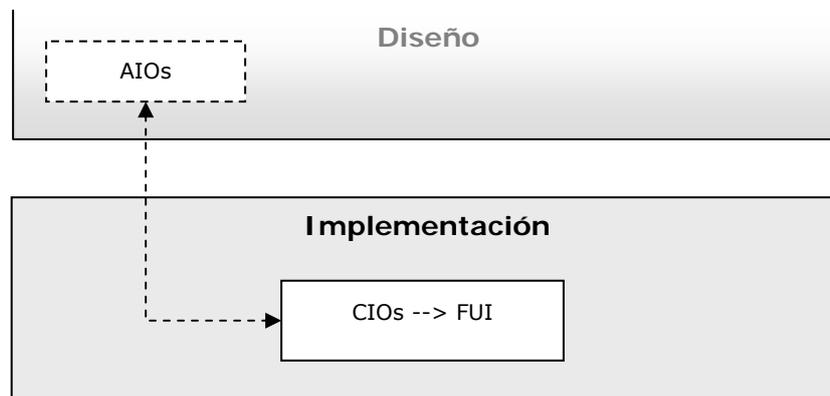


Figura 101. Pasos de la etapa de Implementación

5.5.1 Concrete User Interface y Final User Interface

Mientras que en IDEAS se pasa de AIOs a CIOs concluyendo así el diseño de la UI, el marco de referencia Cameleon descompone el último paso.

Tanto en IDEAS como en Cameleon los AIOs de la etapa de diseño son independientes de la modalidad (interacción gráfica, vocal, reconocimiento y síntesis del habla, interacción basada en video, virtual, realidad mixta o aumentada) y de la plataforma (software y hardware, es decir, entendida como sistema operativo y dispositivo). Sin embargo, los CIOs de IDEAS son directamente widgets que el usuario manipula en la interfaz correspondiente. En las aproximaciones que se enmarcan en Cameleon, como UsiXML, los CIOs siguen manteniendo un cierto nivel de abstracción (independencia de la plataforma) y conforman la Interfaz de Usuario Concreta (Concrete User Interface, CUI). La última concretización son los elementos del FUI (Final User Interface).

IDEAS se basa en XUL (XML-based User-Interface Lenguaje) y los objetos de interacción concretos propuestos para este lenguaje para la generación de la interfaz final. XUL es un lenguaje de marcado multiplataforma basado en XML para el desarrollo de UIs. A pesar de no ser un estándar, se emplea como un medio para

definir GUIs de forma simple y portable. Inicialmente se empleó para desarrollar productos de Mozilla.

Este lenguaje proporciona una serie de controles definidos mediante etiquetas. Son los CIOs empleados en IDEAS. La Figura 102 muestra un par de ejemplos de dichos objetos y las etiquetas correspondientes.

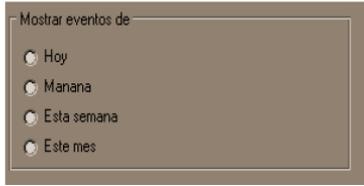
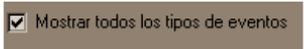
<p>Grupo de botones de exclusión 'radio' metidos en una 'box' con título</p> 	<pre><titledbox orient="vertical" > <title> <text value="Mostrar eventos de"/> </title> <radiogroup id="radiogroup1" > <radio id="Btm1" value="Hoy"/> <radio id="Btm2" value="Manana"/> <radio id="Btm3" value="Esta semana"/> <radio id="Btm4" value="Este mes"/> </radiogroup> </titledbox></pre>
<p>Botones de verificación 'checkbox'</p> 	<pre><checkbox checked="true" value="Mostrar todos los tipos de eventos" /></pre>

Figura 102. Ejemplos de Objeto de Interacción Concreto en XUL empleados en IDEAS

Otros trabajos como el de Pinheiro da Silva, UMLi [Silva, 2002] utilizan aproximaciones similares. En el modelado de la presentación concreta se describe la interfaz en términos de dependencia del entorno, de manera que se presentan directamente los objetos que los usuarios manipulan en la UI. Por ejemplo, la Figura 103 muestra los elementos del modelo de presentación concreto para los componentes de Java Swing, uno de los toolkits más empleados para el desarrollo de todo tipo de aplicaciones.

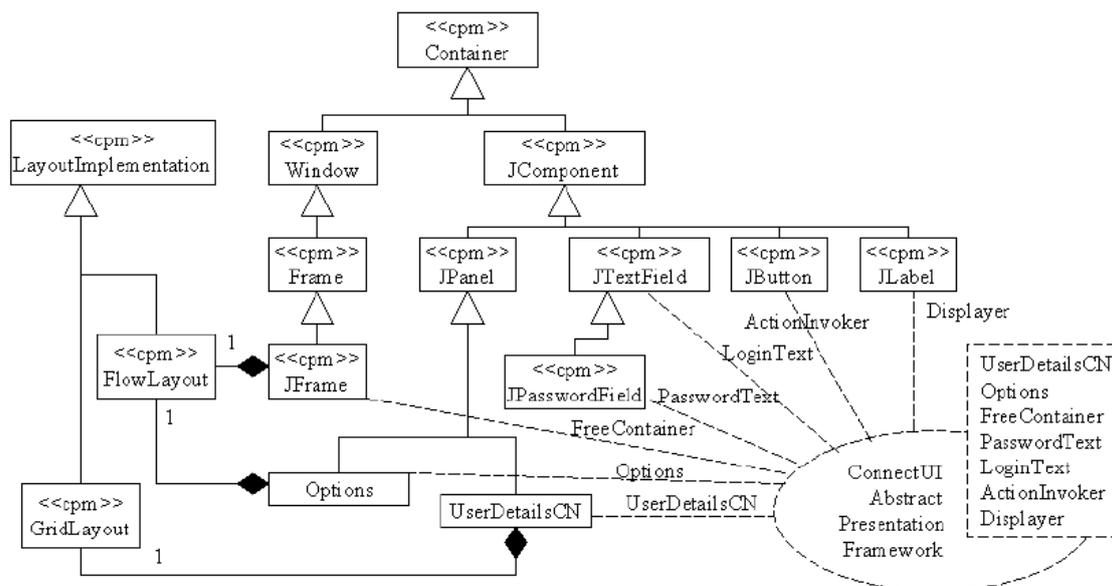


Figura 103. Modelo de presentación concreto para elementos del toolkit Java Swing en UMLi

Puesto que la última evolución en el diseño de interfaces de usuario para entornos HCI se enmarca dentro del proyecto Cameleon, en esta tesis la etapa de implementación sigue también esta aproximación.

La definición completa de CIO se puede obtener de [Limbourg, 2004]. Un CIO es una entidad que el usuario puede percibir o manipular. Un CIO es una abstracción de un conjunto de widgets encontrados en los toolkits más populares y estándares. En otras palabras, un CIO trata de generalizar elementos que aparecen en diferentes herramientas de desarrollo que el usuario final manipula o percibe. Una *etiqueta* puede aparecer en diferentes toolkits, pero la esencia es la misma en todos ellos.

Los CIOs son dependientes de la modalidad, pero todavía son independientes de la plataforma, de los toolkits concretos que se vayan a emplear en su implementación.

Limbourg propone el modelo CUI (Figura 104a) como un conjunto de CIOs y sus relaciones. Además distingue entre CIOs *gráficos* y *auditivos* según la modalidad. A su vez, estos se pueden ser *contenedores* o *componentes individuales*. Así, y respetando la nomenclatura anglosajona, hay cuatro tipos de CIO: *Graphical Container (GC)*, *Graphical Individual Component (GIC)*, *Auditory Container (AC)* y *Auditory Individual Component (AudIC)*. Los CIOs gráficos son descritos con más detalle por su frecuencia y complejidad, frente a los auditivos.

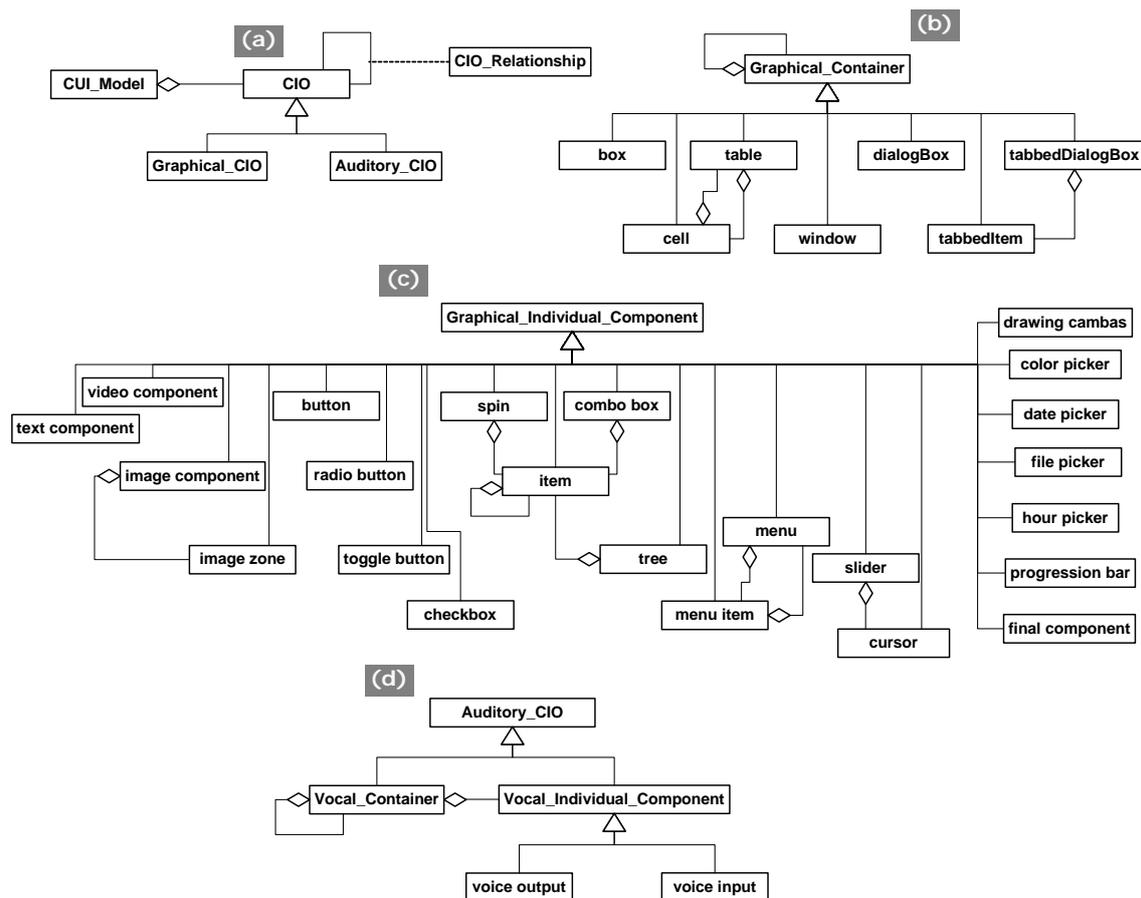


Figura 104. (a) Modelo CUI, (b) CUIs del tipo Graphical Container, (c) CUIs del tipo Graphical Individual Component, (d) CUIs del tipo auditivo

Para cada tipo de CIO se identifican una serie de CIOs como resultado de la abstracción de los elementos de los toolkits más comunes (ver Figura 104):

- Graphical Container (GC): window; tabbed dialog box, tabbed item; table, cell; dialog box; menu bar; box.
- Graphical Individual Component (GIC): text component; video component; image component, image zone; button, radio button, toggle button, check box, radio button; spin, combo box, item, tree; menu, menu item; slider, cursor; drawing canvas, color picker, file picker, date picker, hour picker, progression bar; final component.
- Auditory Container (AC)
- *Auditory Individual Component (AudIC)*: elementos de entrada y de salida.

Una vez diseñada la UI, elegida la modalidad y por lo tanto las CUIs correspondientes, se selecciona la plataforma sobre la que se implementará finalmente la solución. Es entonces cuando la UI es totalmente dependiente del lenguaje, el dispositivo y de la plataforma. Es entonces cuando se seleccionan los elementos físicos, tangibles por el usuario. Se da lugar a la interfaz de usuario final o FUI (Final User Interface).

5.5.2 El Problema de la Falta de Estándares en los Elementos para las Aplicaciones Groupware

En la programación de UIs se emplean multitud de toolkits con elementos más o menos comunes. Generalmente los elementos entre toolkits para plataformas diferentes se asemejan, por lo que se puede abstraer una idea genérica: los CIOs. Toolkits conocidos y muy empleados son Java Swing [Eckstein, 1998], HTML 4.01 [W3C, 1999], Flash DRK6 [Macromedia, 2004], earcons [Crease, 1999] o VoiceXML 2.0 [W3C, 2004].

Ya en 1991, Clarence Ellis, uno de los investigadores más relevantes en torno al groupware, habla acerca de este asunto [Ellis, 1991]. Comenta cómo la tecnología para el desarrollo de interfaces de usuario (single-user) es una tecnología madura que podría generalizarse para su uso en tecnología groupware. Sin embargo, también apunta que los toolkits para interfaces de grupo no deben ser sólo extensiones de toolkits existentes, sino que deben introducir nuevos constructores y ser específicos y originales.

En el desarrollo de aplicaciones groupware el problema radica en la falta de estándares o al menos en la falta de toolkits de uso frecuente con elementos comunes para este tipo de aplicaciones. Algunos elementos como los telepunteros suelen repetirse en el diseño de sistemas CSCW, pero no hay un grupo de toolkits o un conjunto de ideas de las que se pueda abstraer un tipo de CIO o un conjunto de CIOs groupware.

Existen, sin embargo, algunas aproximaciones que proporcionan funcionalidades para el prototipado rápido de aplicaciones groupware. Probablemente el toolkit para el desarrollo de aplicaciones groupware más conocido haya sido Groupkit [Roseman, 1992], elaborado en la Universidad de Calgary por el grupo de Saul Greenberg en Canadá. Es una librería basada en Tcl/Tk que incorpora procesos distribuidos y datos replicados.

Unos años más tarde surge Collabrary [Boyle, 2002] en el seno del mismo grupo. Se considera la sucesión de Groupkit. Está basado en componentes COM para realizar prototipado rápido de aplicaciones groupware multimedia. Posee un diccionario compartido (servidor de notificaciones), permite la manipulación,

análisis y captura de información multimedia y permite transmitir información en tiempo real. Collabrary se puede emplear para desarrollar aplicaciones que trabajen en tiempo real como herramientas de dibujado, editores o herramientas tipo meeting que son compartidas por varios usuarios.

Sin embargo, estos y otros toolkits no están centrados en la interfaz de usuario para el diseño de las aplicaciones groupware. Por ejemplo Collabrary permite el acceso a fuentes de audio, video y codecs, proporciona un medio de transmisión para audio y video, controla accesos de lectura y escritura directos, etc.; e incluso adicionalmente se dice que es compatible con toolkits GUI convencionales como DHTML, Visual Basic, Tcl/Tk, Java, etc.

El desarrollo de aplicaciones groupware es relativamente reciente y su diversidad es tal que hace difícil conocer qué es lo que realmente se necesita, qué generalizar. Por este motivo, aunque algunos toolkits contemplan elementos típicos de estas UIs como un telepuntero, no hay toolkits específicas, genéricas y amplias que contemplan elementos característicos de este tipo de interfaces.

5.5.3 CIOs en los Sistemas Groupware

En la etapa anterior, el diseño, se ha propuesto la extensión del modelo conceptual de AIOs presentado en [Limbourg, 2004] con la inclusión de un nuevo AIO, el *AWAC (Abstract Workspace Awareness Container)*, que facilita el contexto compartido donde el usuario interactúa, así como con la inclusión de tres nuevas facetas que describen los AICs: *embodiment, expressive artifact, visibility*.

En la etapa actual, donde se tienen en cuenta detalles de implementación, se identifican los CIOs independientes de la plataforma.

Los CIOs más importantes en los sistemas CSCW se descubren en aplicaciones síncronas. En aplicaciones asíncronas los CIOs que se emplean normalmente suelen ser comunes a otras aplicaciones.

Para esta identificación se continuará con la extensión del modelo de Limbourg. Concretamente se mantiene la idea de dos tipos de CIO que forman el CUI: los gráficos o Graphical CIO y los auditivos o Auditory CIO. Así mismo se mantiene la idea de que ambos tipos puedan ser contenedores o componentes individuales. Este es el esquema que se muestra en la Figura 105.

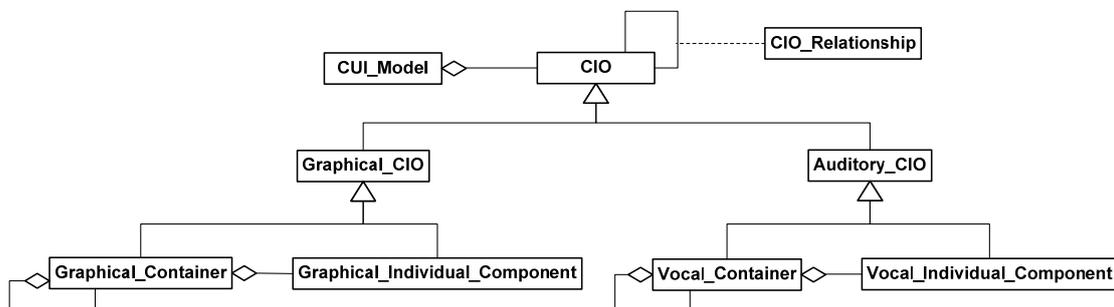


Figura 105. Modelo conceptual de clasificación de CIOs [Limbourg, 2004]

Por otro lado, los CIOs en una aplicación groupware pueden ser elementos *simples* como los vistos hasta ahora para interfaces de aplicaciones HCI (Figura 104), pero

también podría tratarse de *CIOs compuestos*, sin que esto deba confundirse con los *contenedores*.

Un ejemplo para un elemento *simple* sería un *telepuntero*. Se trata de un CIO *componente individual gráfico*. El telepuntero proviene del AIO caracterizado con la faceta *embodiment* puesto que representa al usuario y sus movimientos. Estos elementos simples son, pues, *componentes individuales* o *contenedores*, tanto *gráficos* como *auditivos*. La idea de composición en este caso es diferente.

Un *CIO compuesto* está formado por contenedores, por componentes individuales o incluso por otros elementos compuestos. En aplicaciones groupware se repiten algunos patrones para representar ciertas interfaces. Un ejemplo puede clarificar este concepto: en aplicaciones de chat, meeting, pizarras compartidas, etc. existe un espacio que se repite constantemente. En este espacio compartido se muestran los usuarios del sistema. El patrón que se muestra en la Figura 106 podría ser una generalización de un elemento que podríamos llamar *cuadro de usuarios del sistema*.

El *cuadro de usuarios del sistema* muestra los usuarios que participan en él dando una serie de información complementaria por medio de CIOs tradicionales o de aplicaciones groupware:

- El primer CIO, *Box (1)*, es del tipo *Graphical_Container*. Agrupa una serie de elementos que se repetirán. Forma el cuadro más general de todos y proviene de un AWAC, puesto que es el AIO propuesto en la etapa de diseño de esta metodología para proporcionar espacios compartidos para múltiples usuarios.
- El segundo CIO, *Box (2)*, también es del tipo *Graphical_Container*. En este caso agrupa un conjunto de CIOs relacionados con el concepto de usuario del sistema: avatar, text components y buttons/menus.
- El *avatar* es un CIO propuesto en esta metodología como generalización de elementos en aplicaciones groupware, como se verá posteriormente. Representa al usuario, por lo que proviene de un AIO caracterizado con la faceta *embodiment*. Da información acerca del estado del usuario como: ausente, fuera de línea, no puede comunicarse, está alegre, triste, enfadado, etc.
- El CIO *Text Component (1)* podría ser una etiqueta (como elemento del FUI) con el nombre del usuario.
- El CIO *Text Component (2)* podría ser una etiqueta, un campo de texto, etc. (como elemento del FUI) con información adicional sobre el usuario: página Web, frase del usuario, perfil, etc.
- Por último, un CIO como un *Menú* podría aglutinar un conjunto de acciones típicas sobre el usuario en cuestión: iniciar conversación telefónica, iniciar video-conferencia, iniciar sesión de chat, ver perfil del usuario, etc.

Este *cuadro de usuarios del sistema* del ejemplo sería un típico *CIO compuesto* en una aplicación groupware.

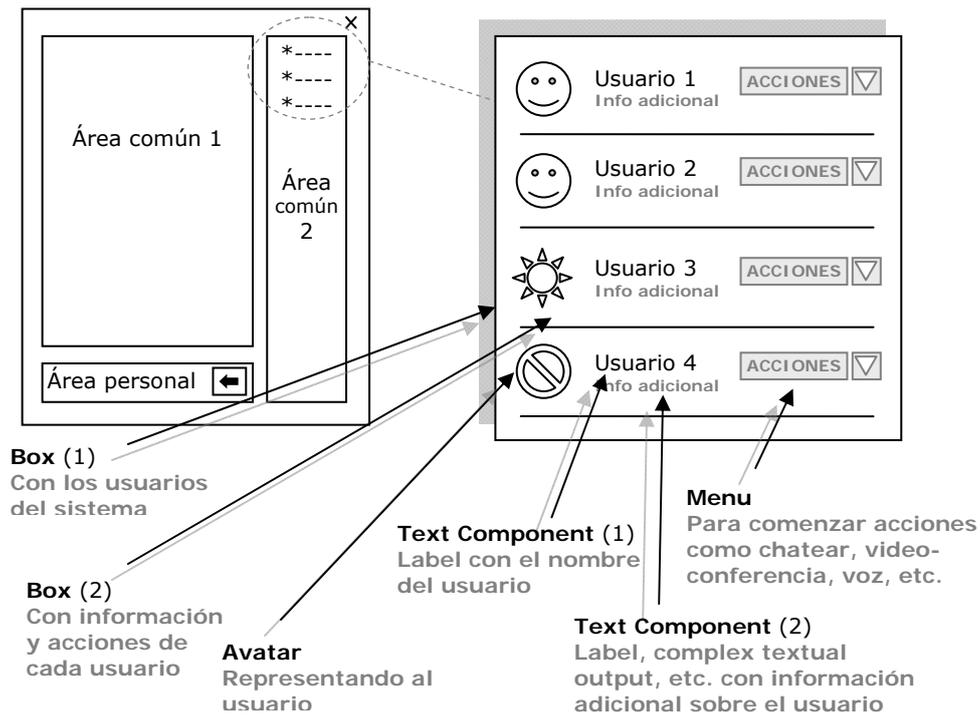


Figura 106. CIO elemento compuesto en un sistema tipo meeting: cuadro de usuarios del sistema

Por este motivo, los CIOs que se propongan para el diseño de interfaces de usuario en sistemas CSCW pueden pertenecer a cualquiera de los tipos propuestos en [Limbourg, 2004] más el tipo elemento compuesto que se ha presentado y que formalmente se ha llamado *Composed Groupware Component (CGC)*. Es decir, los CIOs simples podrían clasificarse como *Graphical Container*, *Graphical Individual Component*, *Vocal Container* o *Vocal Individual Component*. Mientras que un CGC es un tipo nuevo que podría estar formado por CIOs de cualquiera de los tipos anteriores. El modelo conceptual extendido se muestra en la Figura 107. Puesto que un CGC podría contener elementos tanto gráficos como auditivos, el CIO está al mismo nivel que estos dos.

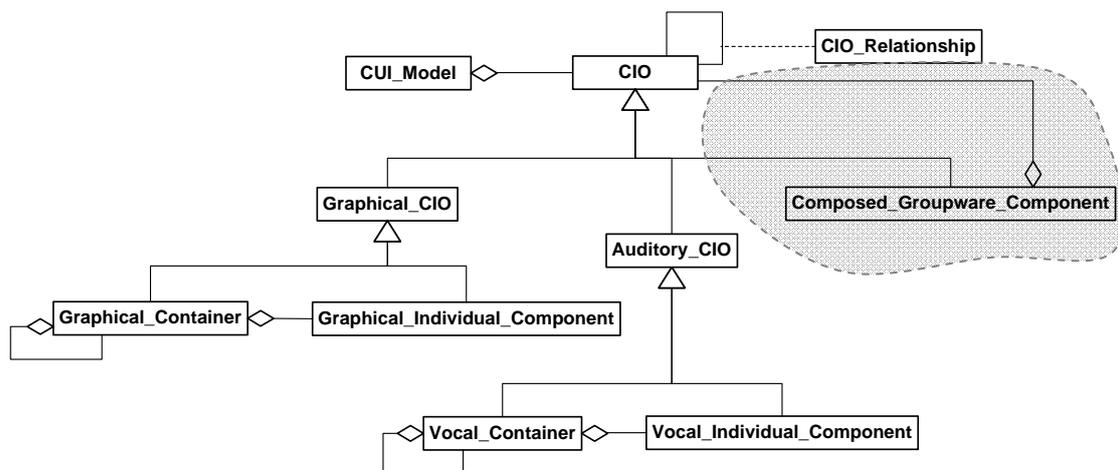


Figura 107. Modelo conceptual de clasificación de CIOs extendido de [Limbourg, 2004] para contemplar CIOs de aplicaciones groupware. Sombreada la propuesta añadida.

A continuación se muestra un conjunto de CIOs clasificados dentro del modelo conceptual anterior. Este conjunto no es único. Es decir, se parte de la generalización de CIOs observados en múltiples aplicaciones groupware y en aplicaciones e investigaciones realizadas por autores relevantes del área, especialmente en la Universidad de Calgary [Gutwin, 2004]; sin embargo, no se trata de un estándar, ni de un grupo de CIOs completo ni cerrado.

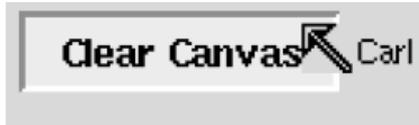
Provenientes de AIOs con faceta *embodiment* se pueden encontrar los siguientes CIOs:

- *Telepuntero* (telepointer). El telepuntero es un cursor que aparece en más de un dispositivo indicando la presencia de varios usuarios. El usuario puede ver su puntero más el puntero de cada uno de los usuarios remotos del sistema. Cuando un usuario mueve el puntero, el movimiento y las acciones se ven reflejadas en las pantallas de los demás usuarios del mismo modo. Los telepunteros pueden incluir el nombre de los usuarios, cambiar el sentido del telepuntero para indicar hacia donde se mueven, etc.
- *Avatar* (avatar). Un avatar es un icono que proporcionan algún tipo de información gestual fácilmente interpretable sobre el usuario al que representa.
- *Video embodiments*. Se trata de una técnica que consiste en mostrar una composición formada por imágenes de los miembros del equipo junto con representaciones del espacio de trabajo. Así, las imágenes de los usuarios se emplazan en el sitio "correcto" relativo al espacio de trabajo. Algunas variantes podrían emplear siluetas, sombras, partes como brazos o manos, etc.

A partir de AIOs caracterizados con la faceta *expressive artifact* (*Process feedthrough* y *Action indicators and animations*) se pueden encontrar los siguientes CIOs:

- *Botones feedthrough*. Si alguien pulsa un botón, en el resto de UIs remotas se enfatizan iluminándolos y haciendo un sonido.
- Para este tipo de acciones o para acciones más rápidas donde es más difícil que el usuario pueda ser consciente de lo que los demás han hecho se pueden emplear las *barras feedthrough*: barras de estado o paneles informativos que van diciendo lo que el resto de usuarios van haciendo en cada momento. Pueden estar enriquecidas mostrando colores, iconos, etc. para las diferentes acciones que se vayan realizando en cada momento, aumentando así el awareness de los usuarios del sistema.
- *Indicadores de acción*. Son señales que aparecen tipo "imagen alternativa" en HTML que proporcionan información sobre lo que el usuario remoto está haciendo. Por ejemplo, si un usuario hace clic con el ratón sobre una zona, junto a su telepuntero puede aparecer una imagen pequeña indicando que ha hecho ese clic. Otro ejemplo sería que pulsara sobre una opción de un menú contextual. En ese caso podría aparecer una pequeña imagen en la pantalla remota mostrando únicamente la opción sobre la que ha hecho clic. Las imágenes de la Figura 108 muestran este tipo de indicadores.

Carl presiona el botón

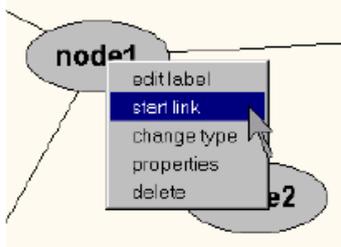


Vista de Saul de la acción enfatizada

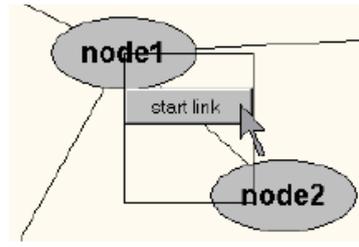


a) Información de acción al presionar un botón

Carl navega por el menú



Vista de Saul de la acción de navegar



b) Un menú popup: vista local (izquierda) y remota (derecha)

Figura 108. Ejemplos de indicadores de acción extraídos de [Gutwin, 2004]

La idea de los AIOs caracterizados con la faceta *visibility* es proporcionar una vista de una parte del espacio de trabajo que no se ve. Es decir, cuando se trabaja sobre un espacio de trabajo virtual, este suele ser más grande de lo que se muestra por pantalla a simple vista. Es muy probable que si hay varios usuarios trabajando en colaboración, algunos de ellos lo han en diferentes zonas del espacio de trabajo y no puedan ser vistas en parte del espacio de trabajo que alcanza a ver cada uno. En [Gutwin, 2004] se identifican tres CIOs que proporcionan una vista del espacio de trabajo que aumenta el awareness y facilita de esta manera las tareas en grupo:

- *Vista Radar*. Proporciona, en un área dedicada, una vista de todo o parte del espacio de trabajo en "miniatura" para poder tener una visión general del mismo. Además se puede apreciar el espacio de trabajo de cada usuario puesto que aparece remarcado y sombreado en el espacio de trabajo global. Ver ejemplo en Figura 109c.
- *Vista Over-the-shoulder*. Proporciona, en un área dedicada, una vista del espacio de trabajo de un usuario remoto concreto. De esta manera se puede apreciar lo que está haciendo en ese momento determinado de la misma manera que podría verlo él. Ver ejemplo en Figura 109d.
- *Vista Cursor's-eye*. Proporciona, en un área dedicada, una vista ampliada de una zona del espacio de trabajo de una manera similar a la Vista Over-the-shoulder, con la diferencia de que permite ver un espacio menor, pero más ampliado del espacio de trabajo del usuario remoto. Así se pueden contemplar con mayor nivel de detalle ciertas acciones. Ver ejemplo en Figura 109e.

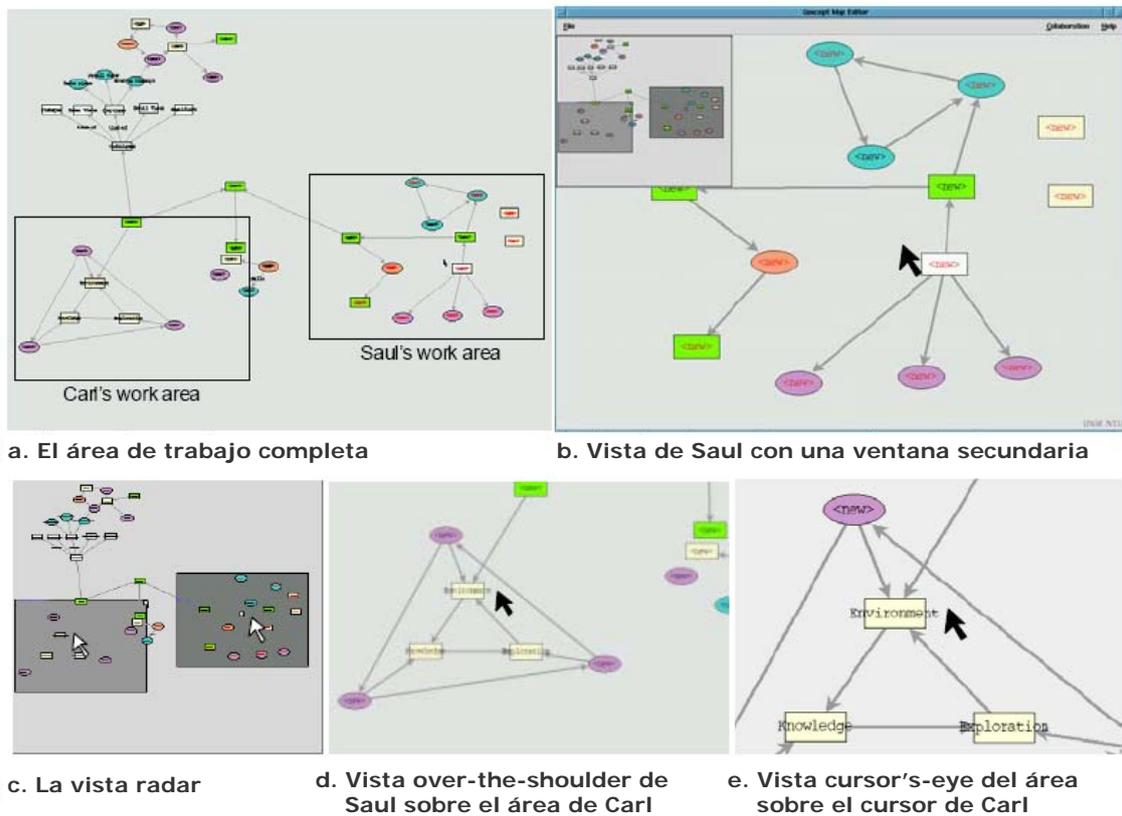


Figura 109. Ejemplos de CIOs como resultado de la reificación de AIOs caracterizados con la faceta *visibility*. Imagen extraída de [Gutwin, 2004]

El número de CIOs para aplicaciones groupware podría ser muy elevado puesto que se trata de elementos que facilitan la colaboración, coordinación, etc. de los usuarios del sistema y, a menudo, lo hacen de una forma muy específica para casos muy concretos. Sin embargo, a continuación se mencionan algunos CIOs del tipo CGCs identificados como generalización de elementos frecuentes en aplicaciones groupware. Este conjunto de elementos es abierto por los motivos considerados anteriormente:

- *Cuadro de Usuarios del Sistema (User Set)*. Este CIO ha sido explicado con anterioridad al definir los CIOs compuestos. Ver Figura 106.
- *Espacio de Trabajo Compartido (Shared Workspace)*. Es el área de trabajo donde los usuarios pueden trabajar y ver el trabajo del resto de usuarios de la aplicación. Es elemento muy genérico que sirve como marco de trabajo para la aplicación. Ejemplos típicos se pueden encontrar en editores colaborativos, pizarras compartidas, etc. Es un elemento compuesto porque suele contar con la presencia de otros elementos como *vistas* o *telepunteros*. En su implementación suelen trabajar sobre espacios de memoria compartida en el servidor central que trata de mantener la coherencia en todo momento por medio de mecanismos típicos como exclusión mutua, semáforos, sincronización, etc. El espacio de trabajo compartido está muy relacionado con tres conceptos descritos por [Ellis, 1991]: contexto compartido (shared context), ventana de grupo (group window) y sesión (session). El *contexto compartido* es un conjunto de objetos donde los objetos las acciones que se pueden realizar sobre los objetos son visibles para un conjunto de usuarios, la *ventana de grupo* es una colección de ventanas cuyas instancias (conectadas)

aparecen en diferentes sitios y la *sesión* es un periodo de tiempo en el que se realizan interacciones síncronas (esto no siempre es así).

Como se ha comentado anteriormente, el conjunto de CIOs descrito sobre estas líneas no es completo ni cerrado. Desde que Clarence Ellis comentara la necesidad de toolkits específicos para el desarrollo de aplicaciones groupware en el 91 [Ellis, 1991], no se ha desarrollado ninguno de manera que se haya convertido en un estándar de facto. Como también se ha mentado en apartados anteriores, sí existen aproximaciones que permiten el desarrollo rápido de prototipos, y es a partir de estos toolkits y de otras generalizaciones desde donde se han propuesto estos CIOs.

Cabe destacar que los CIOs propuestos son especialmente importantes en el desarrollo de interfaces de usuario de aplicaciones síncronas.

5.5.4 Elementos de los FUIs en los Sistemas Groupware

La interfaz de usuario final (FUI) contiene, por fin, los elementos necesarios para la implementación de la interfaz de usuario en la aplicación groupware. Estos elementos son dependientes tanto de la modalidad como de la plataforma elegidas.

Es decir, si en un determinado espacio se ha considerado poner un Text Component como CIO, es ahora el momento de decidir que será un JLabel de Java Swing para una PDA por ejemplo.

Sin embargo, no se identifican elementos de las FUIs por los motivos expresados con anterioridad (apartado 5.5.2) sobre la falta de toolkits genéricos que tengan en cuenta expresamente un conjunto de elementos específicos para las UIs de las aplicaciones groupware.

A título de ejemplo se puede comentar el toolkit *SDGToolkit* desarrollado en el GroupLab de la Universidad de Calgary [Tse, 2004]. Se trata de un framework para diseñar aplicaciones groupware en las que se puedan utilizar diferentes ratones, teclados, cursores y orientaciones desde un mismo dispositivo. Está implementado como paquete .NET.

En la Figura 110 se pueden observar los telepunteros empleados con SDGToolkit resultado de la reificación de los CIOs telepuntero para esta sencilla aplicación de pizarra compartida. En estos telepunteros se puede ver cómo aparece el nombre del usuario junto al telepuntero y cómo la orientación y el estilo del mismo puede variar. En cualquier caso, el objetivo de este toolkit no es proporcionar esos telepunteros, sino la utilización de diferentes recursos (ratones, teclados) desde un mismo dispositivo.

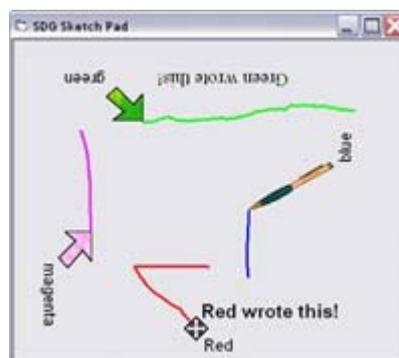


Figura 110. Captura de ejemplo de prototipo rápido creado con SDGT [Tse, 2004]

5.5.5 Trazabilidad

La trazabilidad entre las etapas de diseño e implementación está garantizada puesto que se trata de una *traducción* de los AIOs identificados en la primera a CIOs para la segunda. Recordando que los AIOs son independientes de la modalidad y de la plataforma, y que los CIOs son independientes de la plataforma pero dependientes de la modalidad, se habrá de tener en cuenta el modo de interacción del usuario: interacción gráfica, vocal, reconocimiento y síntesis del habla, interacción basada en video, virtual, realidad mixta o aumentada.

Posteriormente, una nueva *traducción* garantiza la trazabilidad intra-etapa en el proceso de reificación de CIOs a elementos de la FUI. Como dichos elementos en esa interfaz de usuario final son dependientes de la plataforma, ahora sí se ha de tener en cuenta el sistema operativo y el dispositivo finales para poder concluir el desarrollo del a UI de la aplicación.

CAPÍTULO 6. CASO DE ESTUDIO

El caso de estudio proporciona un ejemplo de aplicación de algunas de las aportaciones presentadas en la tesis.

No se trata de un sistema completo, de una especificación de una aplicación groupware en producción. El objetivo de este ejemplo es mostrar aquellos puntos desarrollados en la tesis más interesantes sobre lo que podría ser un caso real.

La especificación de una aplicación como esta requiere un nivel de detalle que queda fuera del alcance y objetivos de este capítulo que, lejos de querer mostrar un documento de especificación del sistema, proporciona una idea más clara de algunos de los puntos de la tesis.

La aplicación groupware presentada es un sistema por medio del cual se podría elaborar un documento de forma cooperativa que posteriormente pasaría un proceso de revisión para determinar si sería publicado o no.

El proceso que se sigue en la especificación del sistema es el propuesto en el capítulo anterior.

6.1 Descripción del problema

La descripción del problema presentada en este apartado es muy breve, simplemente para dar una idea de lo que debería hacer.

La aplicación ficticia ideada se denomina COREA, acrónimo que proviene de tres palabras clave que la describen: Colaboración (y Cooperación), Revisores y Autores.

COREA es una aplicación groupware que permite la colaboración y la cooperación entre diferentes usuarios del sistema para crear y revisar documentos.

La aplicación posibilita la elaboración de un documento entre varios autores a través de la red. Diferentes usuarios conectados pueden elaborar un documento cooperativamente trabajando sobre el mismo de forma síncrona, o bien colaborativamente enviando borradores del mismo que van siendo completados por otros usuarios.

Cuando los autores del documento tienen un documento listo para ser revisado, uno de ellos se encarga de enviar, por medio de la misma aplicación, el documento candidato a ser publicado a los revisores.

Los revisores analizan el documento y dan su opinión acerca de si ha de ser publicado o no. Trabajan de este modo de forma colaborativa.

Un documento que ha sido publicado puede ser leído por todos los usuarios del sistema, aunque no sean autores o revisores, a través de la interfaz Web del sistema.

6.2 Elicitación de requisitos

El modelo de proceso propuesto sugiere una primera etapa de elicitación de requisitos según los pasos desarrollados en el apartado 5.2.7.2.

6.2.1 Adquisición del Conocimiento del Dominio del Problema

En este primer paso de la metodología se debe tomar contacto con el dominio del problema y recabar cuanta información sea posible para comenzar su estudio.

Por medio de la recopilación de documentación, entrevistas, brainstorming, reuniones en grupo, cuestionarios, inmersión en el negocio del cliente, aprendizaje del negocio, modelado del sistema actual, etc. se consigue un conocimiento del sistema que el cliente quiere obtener.

Se obtendría un primer borrador del Documento de Requisitos del Sistema (DRS) con la información recogida hasta el momento. Éste podría contener una primera descripción de los objetivos y requisitos del sistema a partir de las plantillas: *plantilla general para objetivos y requisitos, más la extensión específica correspondiente y la extensión CSCW en su caso.*

En este caso de estudio, este paso se verá reducido a la información obtenida en la descripción del mismo.

6.2.2 Identificar la Estructura Organizativa y los Actores del Sistema

La identificación de los actores del sistema y la descripción de su organización es un paso iterativo que se retroalimenta de los pasos siguientes. Por lo tanto, la información mostrada en las plantillas que recogen esta información es fruto de varias iteraciones, no habiéndose obtenido en una sola pasada.

De la descripción del sistema (del paso de adquisición del conocimiento del dominio del problema) y de las diferentes iteraciones es posible identificar varios actores. Los actores son los participantes del sistema y pueden ser de diferentes tipos: *actores* (si todavía no se ha concretizado qué tipo de actor es, sobre todo en las primeras iteraciones), *grupos*, *individuos*, *usuarios* o *agentes*.

A continuación se muestran las plantillas para la *Estructura Organizativa* y para los diferentes *actores* identificados en el sistema. Las plantillas fueron descritas en el apartado 5.2.6.5.

Considerando que se muestra una última iteración (ya no hay actores ni individuos, son todo grupos, usuarios o agentes concretos), se han identificado los actores que se muestran en la plantilla de la Estructura Organizativa de la Tabla 35.

Tabla 35. Descripción de la Estructura Organizativa de los participantes del sistema del caso de estudio

Estructura Organizativa	
Versión	10 (18/05/2007)
Autores	• Víctor M. R. Penichet (Investigador)
Fuentes	• Descripción textual (caso de estudio ficticio)
Actores	• Ninguno
 -- Grupos	<ul style="list-style-type: none"> • #G-1 (AUTHORS) • #G-2 (REVIEWERS) • #G-3 (INTERNAL) • #G-4 (EXTERNAL) • #G-5 (WHOLE_SYSTEM)
 -- Individuos	• Ninguno
 -- Usuarios	<ul style="list-style-type: none"> • #U-1 (Author) • #U-2 (Chair_author) • #U-3 (Reviewer) • #U-4 (Chair_reviewer) • #U-5 (Reader)
 -- Agentes	<ul style="list-style-type: none"> • #S-1 (Author_notifier) • #S-2 (Reader_notifier)
Descripción	<p>Los participantes están organizados en los siguientes grupos: El grupo #G-1 (AUTHORS) depende jerárquicamente del grupo #G-2 (REVIEWERS) puesto que lógicamente el trabajo de los primeros está supeditado a lo que decidan los segundos. Al grupo de #G-1 (AUTHORS) pertenecen los usuarios #U-1 (Author) y #U-2 (Chair_author). Al grupo #G-2 (REVIEWERS) pertenecen los usuarios #U-3 (Reviewer) y #U-4 (Chair_reviewer). Estos dos grupos conforman el grupo #G-3 (INTERNAL). Existe otro grupo, #G-4 (EXTERNAL), al que pertenecen el usuario #U-5 (Reader). Esta separación es lógica para diferenciar la parte interna de la aplicación de la que pueden ver todos los usuarios incluso sin registrarse en el sistema. Existe un último grupo, #G-5 (WHOLE_SYSTEM), que agrupa a estos dos y a dos agentes del sistema: #S-1 (Author_notifier) y #S-2 (Reader_notifier). Este grupo en realidad es un contenedor al que pertenecen todos los actores del sistema, ya sea directa o indirectamente (por pertenecer a otros grupos).</p>
Comentarios	<p>Los grupos #G-3 (INTERNAL), #G-4 (EXTERNAL) y #G-5 (WHOLE_SYSTEM), así como los agentes #S-1 (Author_notifier) y #S-2 (Reader_notifier) se han considerado en iteraciones posteriores tras analizar el sistema.</p> <p>Los agentes #S-1 (Author_notifier) y #S-2 (Reader_notifier) no pertenecen exactamente a ningún grupo, por eso se han considerado como parte directa del grupo #G-5 (WHOLE_SYSTEM).</p>

Para cada uno de los actores identificados, actores entendidos como término general para referirse a los actores, grupos, individuos, usuarios y agentes, se muestra la información relevante a partir de la plantilla de participantes y la extensión para grupos en su caso (apartado 5.2.6.5).

Para este caso de estudio se mostrarán las especificaciones del usuario Chair_author y del grupo AUTHORS a modo de ejemplo (Tabla 36 y Tabla 37 respectivamente). El resto de actores del sistema se especificarían de forma similar.

Tras este paso se obtendría la información relativa al apartado 2 de DRS. Esta información, como se ha comentado en varias ocasiones, iría enriqueciéndose con cada iteración hasta conseguir una descripción de actores y estructura organizativa estables.

Tabla 36. Descripción del usuario Author a partir de la plantilla para actores

U-2	Chair_author
Versión	3 (18/05/2007)
Autores	<ul style="list-style-type: none"> • Victor M. R. Penichet (Investigador)
Fuentes	<ul style="list-style-type: none"> • Descripción textual (caso de estudio ficticio)
Descripción	La función específica de este usuario es la de enviar a los revisores los documentos candidatos a ser publicados cuando lo considere oportuno.
Supergrupos	Este Actor es parte de los siguientes grupos: <ul style="list-style-type: none"> • #G-1 (AUTHORS)
Jerarquía superior	Este Actor depende jerárquicamente de los siguientes Actores: <ul style="list-style-type: none"> • Ninguna
Jerarquía inferior	Los siguientes actores dependen jerárquicamente de este Actor: <ul style="list-style-type: none"> • Ninguna
Otras asociaciones	Existen estas otras relaciones: <ul style="list-style-type: none"> • #U-1 (Author): los usuarios #U-2 (Chair_author) actúan como representantes de los usuarios Author, enviando a los revisores del documento candidato a ser publicado cuando lo estimen oportuno. • #G-2 (REVIEWERS): los usuarios #U-2 (Chair_author) envían al grupo de revisores REVIEWERS el documento candidato a ser publicado cuando lo estimen oportuno.
Capacidades	Habilidades o responsabilidades del Actor: <ul style="list-style-type: none"> • R-4: Es el responsable último de cuanto se escriba en el documento • R-5: Responsable de seleccionar un documento como candidato para ser revisado • H-3: Tiene visión global • H-4: Es capaz de coordinar de alguna manera a los usuarios #U-1(Author)
Comentarios	Se han de tener en cuenta las capacidades introducidas en el grupo del que forma parte: #G-1 (AUTHORS) *

* En principio no sería necesario realizar este apunte puesto que si un elemento pertenece a un grupo, lógicamente las propiedades del grupo pasan al elemento también. Se ha introducido por aclararlo explícitamente en este ejemplo.

Tabla 37. Descripción del grupo AUTHORS a partir de la plantilla para actores y su extensión para grupos

G-1		AUTHORS
Versión		2 (18/05/2007)
Autores		<ul style="list-style-type: none"> • Víctor M. R. Penichet (Investigador)
Fuentes		<ul style="list-style-type: none"> • Descripción textual (caso de estudio ficticio)
Descripción		<p>Este grupo reúne al conjunto de usuarios cuya función principal en el sistema sea la de elaborar documentos candidatos a ser publicados.</p> <p>Los miembros del grupo son usuarios autenticados del sistema que elaboran estos documentos hasta que los revisores consideran que estos documentos, que de momento son borradores o candidatos a ser publicados, son finalmente publicados o rechazados. La elaboración de los documentos puede ser de forma individual por medio de un procesador de textos. También podría ser de forma colaborativa, es decir, cada miembro elabora parte del documento y se discuten con herramientas de autor como la herramienta de control de cambios de MS Word. En este caso el documento es enviado/recibido por los usuarios del sistema para su elaboración. Por último, el documento lo pueden elaborar de forma cooperativa por medio de la parte síncrona de la herramienta que permite que se realice por parte de varios usuarios en tiempo real.</p>
Supergrupos		<p>Este Actor es parte de los siguientes grupos:</p> <ul style="list-style-type: none"> • #G-3 (INTERNAL). Esta relación de pertenencia se ha introducido por separar de forma lógica los usuarios anónimos (en este caso Reader) de aquellos que se autentican en el sistema. • ...
Jerarquía superior		<p>Este Actor depende jerárquicamente de los siguientes Actores:</p> <ul style="list-style-type: none"> • #G-2 (REVIEWERS): este grupo supervisa el trabajo de los autores (usuarios del grupo #G-1 (AUTHORS) de manera que un documento no será público si estos no lo estiman oportuno. • ...
Jerarquía inferior		<p>Los siguientes actores dependen jerárquicamente de este Actor:</p> <ul style="list-style-type: none"> • Ninguna
Otras asociaciones		<p>Existen estas otras relaciones:</p> <ul style="list-style-type: none"> • No
Capacidades		<p>Habilidades o responsabilidades del Actor:</p> <ul style="list-style-type: none"> • R-1: Es responsable directo de cuanto escriba en el documento • R-2: El contenido ha de ser original • R-3: Puede introducir contenidos • H-1: Capacidad investigadora • H-2: Capacidad de expresión
Comentarios		No
Grupo		
Objetivo común		#OBJ-1 (Elaboración de documentos candidatos a ser publicados)
Pertenencia		<p>Este Grupo está formado por los siguientes participantes:</p> <ul style="list-style-type: none"> • #U-1 (Author) • #U-2 (Chair_Author)
Leyes		<p>Normas impuestas por el grupo:</p> <ul style="list-style-type: none"> • L-1: Para pertenecer al grupo, un actor debe cumplir con las capacidades especificadas. • L-2: Los miembros del grupo #G-2 (REVIEWERS) no pueden ser miembros de este grupo para un mismo documento a elaborar/revisar.

6.2.3 Definir los Objetivos del Sistema

La identificación de los objetivos del sistema permite dividirlo en subsistemas que hagan más sencilla su especificación y, como consecuencia, su desarrollo final.

Dejando a un lado objetivos como los de autenticación en el sistema y otros detalles, en el estudio de este ejemplo se muestran los objetivos principales: elaboración síncrona de documentos candidatos a ser publicados y revisión de documentos candidatos. Para la descripción de estos objetivos del sistema se emplean las plantillas para objetivos (y extensión específica) del apartado 5.2.6.4, tal y como se muestra en la Tabla 38 y en la Tabla 39.

Tabla 38. Descripción del objetivo "elaboración síncrona de documentos candidatos a ser publicados" a partir de la plantilla para objetivos

OBJ-3	Elaboración síncrona de documentos candidatos a ser publicados
Versión	1 (21/05/2007)
Autores	• Victor M. R. Penichet (Investigador)
Fuentes	• Descripción textual (caso de estudio ficticio)
Objetivos asociados	• #OBJ-2 (Revisión de documentos candidatos)
Requisitos asociados	• # <i>{RI-<id>, RF-<id>, RNF-<id>}</i> (<nombre del requisito>) • ...
Importancia	Muy importante
Urgencia	Alta
Estado	Especificado; Por implementar
Estabilidad	Puede sufrir cambios, pero actualmente estable
Necesidad de percepción	<p>De este objetivo deberían estar informados los siguientes actores:</p> <ul style="list-style-type: none"> • #G-1 (AUTHORS): <ul style="list-style-type: none"> - Qué: ha de ser informado de los cambios que realicen otros en el documento - Cómo: gráficamente, con áreas dedicadas tipo sección crítica - Cuándo: en tiempo real - Dónde: en la misma pantalla en la que trabaja (en el mismo documento) - Por qué: para conocer los cambios realizados en el documento y saber por dónde va su proceso de elaboración • #U-2 (Chair_author): <ul style="list-style-type: none"> - Qué: el documento está listo para ser enviado a revisión - Cómo: los miembros del grupo #G-1 (AUTHORS) pueden comunicárselo explícitamente por medio de los mecanismos de comunicación de la herramienta (chat) u otros medios - Cuándo: cuando todos los implicados consideren que está listo para ser enviado a revisión - Dónde: --- - Por qué: necesita saber cuándo enviarlo al proceso de revisión. Este es el punto clave para poder enviarlo, es decir, que los propios autores consideren que ya está listo • #G-2 (REVIEWERS): <ul style="list-style-type: none"> - Qué: han de ser informados de documentos candidatos listos para ser revisados - Cómo: notificación - Cuándo: el proceso lo inicia el usuario #U-2 (Chair_author) cuando considera que el documento está listo - Dónde: por medio de la intranet del sistema y por e-mail - Por qué: necesita saber qué documentos están listos para poder comenzar el proceso de revisión y saber cuáles serán finalmente públicos
Participantes	<p>Los actores que participan en la consecución del objetivo son:</p> <ul style="list-style-type: none"> • #G-1 (AUTHORS): elaboración síncrona del documento • #U-2 (Chair_author): envió a revisión cuando documento candidato esté listo
Comentarios	Ninguno
Descripción	El sistema deberá dar soporte a la elaboración síncrona de documentos candidatos a ser publicados. Es decir, los documentos, antes de ser públicos tras su proceso de revisión, pueden ser elaborados por diferentes usuarios a la vez a través del sistema.
Dependencias-S	• #OBJ-1 (Autenticación)
Dependencias-I	• # <i>{OBJ-<id>, RI-<id>, RF-<id>, RNF-<id>}</i> (<nombre del objetivo o requisito>) • ...
Descripción CSCW	<p>Por la naturaleza colaborativa del objetivo se debería tener en cuenta lo siguiente:</p> <ul style="list-style-type: none"> • Los usuarios interactúan en tiempo real sobre un mismo documento • Los usuarios se comunican por medio de chat
Descripción del entorno	<p>El entorno de ejecución del sistema será:</p> <ul style="list-style-type: none"> • Los usuarios podrán interactuar desde cualquier máquina con un navegador y acceso a internet • No hay otra necesidad específica adicional
Coordinación	No
Cooperación	Sí. Han de poder escribir sobre un mismo documento en tiempo real
Colaboración	No
Comunicación	Sí. Al menos por medio de un chat

Espacio	Diferente
Tiempo	Síncrono
Nivel de exigencias	La exigencia más importante es la necesidad de ejecución en tiempo real. Es necesario que funcione con agilidad para no entorpecer el desarrollo del documento, sin embargo, puesto que realmente trabajan sobre zonas diferentes, no es absolutamente vital. Se podría permitir un cierto retraso ocasional, no así en la parte de comunicación.

Tabla 39. Descripción del objetivo “revisión de documentos candidatos” a partir de la plantilla para objetivos

OBJ-2	Revisión de documentos candidatos
Versión	1 (21/05/2007)
Autores	• Victor M. R. Penichet (Investigador)
Fuentes	• Descripción textual (caso de estudio ficticio)
Objetivos asociados	• #OBJ-3 (Elaboración síncrona de documentos candidatos a ser publicados)
Requisitos asociados	• #{RI-<id>, RF-<id>, RNF-<id>} (<nombre del requisito>) • ...
Importancia	Muy importante
Urgencia	Alta
Estado	Especificado; Por implementar
Estabilidad	Puede sufrir cambios, pero actualmente estable
Necesidad de percepción	<p>De este objetivo deberían estar informados los siguientes actores:</p> <ul style="list-style-type: none"> • #G-2 (REVIEWERS): <ul style="list-style-type: none"> - <i>Qué</i>: documento listo para revisar - <i>Cómo</i>: notificación - <i>Cuándo</i>: cuando el usuario #U-2 (Chair_author) determine que está listo para ser revisado - <i>Dónde</i>: a través de la intranet y del e-mail - <i>Por qué</i>: comenzar proceso revisión • #G-1 (AUTHORS): <ul style="list-style-type: none"> - <i>Qué</i>: estado proceso revisión - <i>Cómo</i>: notificación - <i>Cuándo</i>: en cada cambio de estado - <i>Dónde</i>: a través de la intranet y del e-mail - <i>Por qué</i>: para dar seguridad y confianza sobre el proceso de revisión • #U-4 (Chair_reviewer): <ul style="list-style-type: none"> - <i>Qué</i>: opiniones de todos los revisores - <i>Cómo</i>: notificaciones - <i>Cuándo</i>: cuando los revisores (miembros del grupo #G-2 (REVIEWERS)) finalicen su revisión - <i>Dónde</i>: intranet - <i>Por qué</i>: para poder hacer valoración y decidir si el documento revisado por varios revisores es publicado o no. • #G-3 (INTERNAL): <ul style="list-style-type: none"> - <i>Qué</i>: Notificaciones de ko, es decir, documentos rechazados para su publicación - <i>Cómo</i>: notificación - <i>Cuándo</i>: tras decisión de usuario #U-4 (Chair_reviewer) - <i>Dónde</i>: intranet y e-mail - <i>Por qué</i>: todos los miembros de este grupo han de saber el resultado del proceso si ha sido rechazado para su publicación. • #G-5 (WHOLE_SYSTEM): <ul style="list-style-type: none"> - <i>Qué</i>: Notificaciones de ok, es decir, documentos aceptados para su publicación - <i>Cómo</i>: notificación - <i>Cuándo</i>: tras decisión de usuario #U-4 (Chair_reviewer) - <i>Dónde</i>: extranet, intranet y e-mail (a quien corresponda) - <i>Por qué</i>: todos los usuarios de la aplicación han de poder conocer el resultado del proceso si ha sido aceptado y han de poder acceder a documentos publicados. • #S-1 (Author_notifier): <ul style="list-style-type: none"> - <i>Qué</i>: estados del proceso, resultados del proceso - <i>Cómo</i>: internamente. Es parte del workflow - <i>Cuándo</i>: en cada cambio de estado y tras decisión de usuario #U-4 (Chair_reviewer) - <i>Dónde</i>: --- - <i>Por qué</i>: es un agente de notificación • #S-2 (Reader_notifier): <ul style="list-style-type: none"> - <i>Qué</i>: resultado del proceso en caso de que el documento sea admitido para

	publicación - <i>Cómo</i> : internamente. Es parte del workflow - <i>Cuándo</i> : tras decisión de usuario #U-4 (Chair_reviewer) - <i>Dónde</i> : --- - <i>Por qué</i> : es un agente de notificación
Participantes	Los actores que participan en la consecución del objetivo son: • #G-2 (REVIEWERS): realizan la revisión del documento y dan el resultado • #U-4 (Chair_reviewer): decide si publicar o no el documento y da pie al envío de las notificaciones correspondientes
Comentarios	Ninguno
Descripción	El sistema deberá proporcionar soporte a la revisión de documentos candidatos a ser republicados. Los revisores deciden si el documento puede ser público o por el contrario rechazan su publicación. En cualquier caso se informa del resultado.
Dependencias-S	• #OBJ-1 (Autenticación)
Dependencias-I	• #{OBJ-<id>, RI-<id>, RF-<id>, RNF- <id>} (<nombre del objetivo o requisito>) • ...
Descripción CSCW	Por la naturaleza colaborativa del objetivo se debería tener en cuenta lo siguiente: • Revisión asíncrona de documentos • Notificación de revisión al responsable (#U-4 (Chair_reviewer)) • Notificación de resultados • Notificación de estados
Descripción del entorno	El entorno de ejecución del sistema será: • Los usuarios podrán interactuar desde cualquier máquina con un navegador y acceso a internet • No hay otra necesidad específica adicional
Coordinación	Sí. El proceso que se va siguiendo deja algunas notificaciones sobre el estado o el resultado de la revisión. De todas maneras no es una particularidad especial del sistema
Cooperación	No
Colaboración	Sí. Los revisores dan su resultado asíncronamente. El responsable decide en base a esos resultados
Comunicación	Sí. Por medio de e-mail o sistema intranet con herramienta tipo foro
Espacio	Diferente
Tiempo	Asíncrono
Nivel de exigencias	No demasiado altas

La identificación y descripción de los objetivos es "posterior" a la identificación y descripción de los actores del sistema. Sin embargo, hay un campo en la plantilla para grupos que relaciona al grupo con un posible objetivo común de los miembros del grupo. Como el proceso es iterativo, si del paso 1 de esta primera etapa se ha podido deducir un objetivo del sistema, se podría haber apuntado en esta plantilla como primera versión, o simplemente se podría haber dejado en blanco inicialmente el objetivo común en la descripción del grupo y haberlo identificado en una iteración posterior.

Algo similar ocurre con los requisitos en la descripción de estos objetivos. Los requisitos se identifican y describen en el paso siguiente, sin embargo, la plantilla para la descripción de los objetivos los pone en relación con requisitos que no han sido identificados todavía. Una vez más, las iteraciones irán haciendo el proceso de especificación más completo, describiendo en cada caso lo que sea más natural.

En el metadato *necesidad de percepción*, se introduce la información de la que han de tener constancia los diferentes actores para el objetivo dado. Esto se hace a un alto nivel, puesto que en la descripción de requisitos se especificará con mayor detalle. Ahora interesa poder describir el cada objetivo del sistema y saber qué hace quién, cómo afecta, etc. sin entrar en demasiado detalle.

Se trata de objetivos del sistema, de una división del sistema en subsistemas. Como se trata de una aplicación en la que se dará colaboración y/o cooperación entre usuarios, es inevitable tener que introducir en la descripción de ambos

objetivos la extensión CSCW, en la que se expresará, de nuevo en un lenguaje de alto nivel y a grandes rasgos, cómo es esa colaboración.

Otros objetivos del sistema serían los relativos a la gestión de usuarios, elaboración asíncrona de documentos candidatos a ser publicados o el manejo público de documentos y se describirían de forma análoga.

6.2.4 Elicitar los Requisitos

En este paso se identifican y describen los requisitos de información, funcionales y no funcionales del sistema a partir de la información que se tiene.

En este apartado se muestra la descripción de dos requisitos de este caso de estudio: un requisito funcional (Tabla 40) para ver la relación de éste con los casos de estudio, tareas, etc. y un requisito no funcional (Tabla 41) puesto que estos requisitos son especialmente interesantes en los sistemas CSCW. No se muestra ningún ejemplo de especificación de requisito de información porque son los menos interesantes en este trabajo según lo mencionado hasta ahora, pero se harían de forma análoga.

Para la descripción de estos requisitos se emplean las plantillas para requisitos (y extensiones específicas) del apartado 5.2.6.4.

El primer requisito mostrado es un requisito funcional del sistema. La Figura 111 muestra el diagrama de casos de uso correspondiente al objetivo del sistema *OBJ-3: Elaboración síncrona de documentos candidatos a ser publicados*. Los requisitos más importantes se expresan como caso de uso y se describen de acuerdo a las plantillas mencionadas. En este ejemplo se muestra la descripción del requisito *Editar documento*.

El diagrama de casos de uso correspondiente al objetivo mencionado muestra un conjunto de casos de uso relacionados. Son requisitos identificados en el sistema de alguna manera a partir de la información obtenida en los pasos anteriores. Son requisitos como acceder al sistema, registrarse en él, crear, iniciar unirse o eliminar una sesión para trabajar sobre un mismo documento; crear un nuevo documento, editarlo o eliminarlo; crear una nueva versión del mismo; los autores pueden considerarlo listo para revisión (considerar para revisión); pero es el usuario Chair_author el que debe decidir si realmente lo está para enviarlo.



Figura 111. Diagrama de Casos de Uso del objetivo del sistema *OBJ-3: Elaboración sincrónica de documentos candidatos a ser publicados*

Estos requisitos son los necesarios para la elaboración de documentos candidatos a ser publicados. Los únicos actores que participan de estos requisitos son el grupo **AUTHORS** y el usuario **Chair_author**, usuario que, como se puede apreciar por la relación de herencia que se muestra en el diagrama, puede realizar las mismas acciones que los usuarios pertenecientes al grupo **AUTHORS** puesto que en la descripción de los mismos se comentó que de hecho este usuario pertenece a este grupo. Adicionalmente puede considerar definitivamente listo para enviar a revisión un documento candidato.

Tabla 40. Descripción del requisito funcional "Editar documento" a partir de la plantilla para requisitos

RF-8	Editar documento	
Versión	1 (04/06/2007)	
Autores	• Víctor M. R. Penichet (Investigador)	
Fuentes	• Descripción textual (caso de estudio ficticio)	
Objetivos asociados	• #OBJ-3: Elaboración síncrona de documentos candidatos a ser publicados	
Requisitos asociados	• #RI-<id>, RF-<id>, RNF- <id> (<nombre del requisito>) • ...	
Importancia	Muy importante	
Urgencia	Alta	
Estado	Especificado; Por implementar	
Estabilidad	Puede sufrir cambios, pero actualmente estable	
Necesidad de percepción	De este requisito deberían estar informados los siguientes actores: • #G-1 (AUTHORS): - <i>Qué</i> : un actor está modificando una parte del documento - <i>Cómo</i> : muestra gráficamente dónde se está realizando la modificación - <i>Cuándo</i> : en tiempo real - <i>Dónde</i> : en la misma pantalla en la que trabaja (en el mismo documento) - <i>Por qué</i> : para saber quién está modificando qué parte y no interceder en esa parte - <i>Qué</i> : un actor modificó un documento - <i>Cómo</i> : notificación por medio de mail con las partes modificadas - <i>Cuándo</i> : de forma asíncrona una vez salvada la nueva versión del documento - <i>Dónde</i> : por medio del correo electrónico y en la intranet del actor - <i>Por qué</i> : para saber quién modificó qué parte	
Participantes	Los actores que participan en la consecución del requisito son: • #G-1 (AUTHORS)	
Comentarios	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del grupo #G-1 (AUTHORS) edite un documento.	
Precondición	El documento al que se hace referencia existe, el usuario es un usuario registrado del sistema perteneciente al grupo #G-1 (AUTHORS), se ha iniciado una sesión de trabajo	
Secuencia Normal	Paso Acción	
	1	El usuario selecciona el documento que pretende editar
	2	Se realiza el caso de uso RF-7 (Validar sesión)
	3	Se realiza el caso de uso RF-9 (Validar documento)
	4	El usuario selecciona la herramienta de edición que usará
	5	El usuario marca dónde realizará la edición del documento
	6	El usuario realiza las modificaciones oportunas
	7	El sistema envía información al resto de usuarios sobre las modificaciones que se están realizando
	8	El usuario salva las modificaciones
9	El sistema manda notificación por correo electrónico y en la intranet sobre los cambios realizados al documento	
Postcondición	Salvar o cancelar cambios	
Excepciones	Paso Acción	
	-	-
	-	-
Rendimiento	Paso Cota de tiempo	
	7	Tiempo real
	9	De forma asíncrona
Frecuencia esperada	Varias veces por sesión y por usuario	
Descripción CSCW	Por la naturaleza colaborativa del requisito se debería tener en cuenta lo siguiente: • Las notificaciones son necesarias para mantener la percepción de los usuarios • La inserción, modificación y eliminación de nuevos elementos en el documento es un paso de ligado puesto que se debe mantener la percepción de los usuarios en tiempo real. Son acciones que pueden ocurrir demasiado deprisa (p.e. el borrado de una imagen del documento) y el resto de usuarios han de ser conscientes. • La sensación de tiempo real para la edición del documento es importante aunque no crucial. Se puede permitir una ligera latencia no demasiado elevada.	
Descripción del entorno	El entorno de ejecución del sistema será: • --	
Coordinación	No	

Cooperación	Sí. Están escribiendo un documento en tiempo real, luego habrá zonas que se traten como secciones críticas, es decir, si un usuario está modificando un párrafo, otro usuario no puede modificar ese mismo párrafo: protección/desprotección.
Colaboración	No
Comunicación	No
Espacio	Diferente
Tiempo	Síncrono cuando se trata de editar el documento y percibir a los demás usuarios de esta modificación, pero además tiene algo de asincronía, puesto que cuando se salva un documento también se envía, y ya no en tiempo real, una notificación con esos cambios realizados.
Nivel de exigencias	No demasiado altas. Un poco más en la edición síncrona.

La descripción de los requisitos se completa con las iteraciones. Por ejemplo, en las primeras iteraciones probablemente no se puede relacionar bien con qué otros requisitos está relacionado el que se está describiendo; sin embargo, poco a poco esa información se agrega de manera que la descripción sea más completa.

Tabla 41. Descripción del requisito no funcional "Percepción en la edición" a partir de la plantilla para requisitos

RNF- 1	<Percepción en la edición >
Versión	1 (05/06/2007)
Autores	• Victor M. R. Penichet (Investigador)
Fuentes	• Descripción textual (caso de estudio ficticio)
Objetivos asociados	• #OBJ-3: Elaboración síncrona de documentos candidatos a ser publicados
Requisitos asociados	• # {RI-<id>, RF-<id>, RNF- <id>} (<nombre del requisito>) • ...
Importancia	Muy importante
Urgencia	Alta
Estado	Especificado; Por implementar
Estabilidad	Puede sufrir cambios
Necesidad de percepción	De este requisito deberían estar informados los siguientes actores: • En este caso no es un requisito del que tenga que estar informado nadie. Ese es el requisito en sí. Habrá que ver las necesidades de percepción en el resto de requisitos.
Participantes	Los actores que participan en la consecución del requisito son: • #G-1 (AUTHORS)
Comentarios	Ninguno
Descripción	El sistema deberá mantener la percepción de los usuarios en la edición de los documentos
Descripción CSCW	Por la naturaleza colaborativa del requisito, se debería tener en cuenta lo siguiente: • probablemente sean necesario elementos de personificación que muestren qué usuarios están en línea • elementos que muestren qué está haciendo quién • elementos que muestren quién hace qué • elementos que ralenticen o muestren mejor acciones que suceden demasiado deprisa (borrado de elementos...)
Descripción del entorno	El entorno de ejecución del sistema será: • --
Coordinación	No
Cooperación	No
Colaboración	No
Comunicación	Sí. Es el sistema el que comunica al usuario qué está haciendo otro actor del sistema.
Espacio	Diferente
Tiempo	Habrán elementos de percepción síncronos cuando se trate de la elaboración conjunta (cooperación) del documento por parte de varios actores. Sin embargo, se consideran algunos elementos asíncronos cuando se trate de notificaciones por medio de correo electrónico al finalizar una edición, por ejemplo.
Nivel de exigencias	No demasiado altas. Un poco más en la edición síncrona

6.2.5 Priorizar Objetivos y Requisitos

En este paso se habrían de organizar los objetivos y los requisitos por medio de una lista ordenada por prioridad de ejecución. En este caso, como se trata de un ejemplo didáctico donde no se han descrito todos los objetivos y requisitos que formarían parte del sistema, no se muestra esa lista. Ese es el orden en el que se podrían mostrar los requisitos y los objetivos en el DRS (Documento de Requisitos del Sistema).

6.3 Análisis

Como se comentó con anterioridad, la etapa de análisis versa sobre el estudio del dominio del problema: descubrir el *qué* describiendo los requisitos del sistema sin describir asuntos de implementación.

En el caso de los sistemas CSCW se centrará el estudio en aspectos relevantes como los roles que desempeñan los actores previamente identificados y las tareas que realizan cada uno de ellos a partir de los requisitos (especialmente las tareas de grupo).

Tras la identificación y descripción de roles y tareas, se emplean una serie de diagramas para describir la estructura y el comportamiento del sistema. En esta metodología se emplean concretamente los diagramas de clases y OSD y los diagramas de tareas y CDs respectivamente.

Como siempre, es un proceso altamente iterativo en el que cada paso retroalimenta a los demás, lo cual completa la descripción del sistema.

6.3.1 Identificación y Descripción de Roles

El punto de partida es el conjunto de actores identificados en la etapa anterior. A partir de los interrogantes propuestos en esta etapa y de una generalización de los tipos de actores identificados se propone una *lista de roles candidatos*.

En este caso de estudio se han identificado los actores que muestra la Tabla 35:

- #G-1 (AUTHORS)
- #G-2 (REVIEWERS)
- #G-3 (INTERNAL)
- #G-4 (EXTERNAL)
- #G-5 (WHOLE_SYSTEM)
- #U-1 (Author)
- #U-2 (Chair_author)
- #U-3 (Reviewer)
- #U-4 (Chair_reviewer)
- #U-5 (Reader)
- #S-1 (Author_notifier)
- #S-2 (Reader_notifier)

De los actores anteriores:

- Los autores (AUTHORS, Author, Chair_author) pueden escribir documentos candidatos a ser publicados.
- Hay autores responsables de enviar un documento como candidato a ser revisado para su publicación (Chair_author).
- Los revisores (REVIEWERS, Reviewer, Chair_reviewer) se encargan de la revisión de un documento candidato a ser revisado para su publicación.
- Hay revisores (Chair_reviewer) que determinan si el documento candidato se publica o se rechaza.
- Algunos agentes (Author_notifier, Reader_notifier) se encargan de realizar las notificaciones correspondientes sobre los documentos.
- Todos los actores pueden leer documentos publicados.

El resto de actores no mencionados se han empleado por cuestiones de organización logística y probablemente no tienen una función específica. Por lo tanto la lista de roles candidatos podría ser la siguiente:

1. AUTHORS que pueden escribir un documento.
2. Author que puede escribir un documento.
3. Chair_author que puede escribir un documento.
4. Chair_author que puede enviar a revisión un documento candidato a publicar que considere listo para ello.
5. REVIEWERS que pueden revisar un documento.
6. Reviewer que puede revisar un documento.
7. Chair_reviewer que puede revisar un documento.
8. Chair_reviewer que toma la decisión de publicar o rechazar un documento candidato.
9. Author_notifier que envía notificaciones a los autores de los documentos informándoles del estado.
10. Reader_notifier que envía notificaciones a los usuarios que pueden leer documentos públicos indicándoles que hay uno nuevo.
11. AUTHORS que pueden leer un documento publicado.
12. Author que pueden leer un documento publicado.
13. Chair_author que pueden leer un documento publicado.
14. REVIEWERS que pueden leer un documento publicado.
15. Chair_reviewer que pueden leer un documento publicado.
16. Reader que pueden leer un documento publicado.

En un segundo paso, el listado de roles candidatos ha de pasar a un listado de roles definitivo. Es un proceso de refinamiento que da lugar al Listado de Roles del Sistema. En este paso, los roles identificados se deben agrupar por roles similares con los objetivos de evitar tener roles demasiado parecidos que realmente pudieran ser uno solo, evitar duplicados, simplificar y generalizar el modelo, etc.

La lista de roles del sistema que se obtiene es la siguiente:

- Writer: a partir de los puntos 1-3.
- Chair_writer: a partir del punto 4.
- Reviewer: a partir de los puntos 5-7.
- Chair_reviewer: a partir del punto 8.
- Notifier: a partir de los puntos 9 y 10.
- Reader: a partir de los puntos 11-16.

Por último se realiza una descripción de los roles a partir de las plantillas propuestas. Sirva de ejemplo la descripción del rol Writer que se muestra en la Tabla 42.

Tabla 42. Descripción del rol Writer a partir de la plantilla para la descripción de los Roles del sistema

Rol-1	Writer
Versión	1 (05/06/2007)
Autores	<ul style="list-style-type: none"> • Víctor M. R. Penichet (Investigador)
Fuentes	<ul style="list-style-type: none"> • Descripción textual (caso de estudio ficticio)
Descripción	Un usuario del sistema que desempeñe este rol puede elaborar documentos. <ul style="list-style-type: none"> • Las responsabilidades que se requieren de un actor para que pueda desempeñar este rol son las siguientes:
Responsabilidades	<ul style="list-style-type: none"> • R1: Es el responsable directo de cuanto se escriba en el documento • R2: El contenido ha de ser original • R3: Debe introducir contenidos • Las habilidades que se requieren de un actor para que pueda desempeñar este rol son las siguientes:
Habilidades	<ul style="list-style-type: none"> • H1: Capacidad investigadora • H2: Capacidad de expresión
Permisos	Puede escribir en los documentos, crearlos, modificarlos y destruirlos. Los siguientes actores desempeñan este rol:
Actores	<ul style="list-style-type: none"> • #G-1 (AUTHORS) • #U-1 (Author) • #U-2 (Chair_author)
Comentarios	--

Nota: como se comenta en la descripción de esta etapa, se podría ampliar la descripción de los roles por medio de los metadatos propuestos para el paso 4, pero no se lleva a cabo para no hacer demasiado tediosa esta etapa.

Puesto que ya están identificados los roles y los actores del sistema, ya se está en condiciones de poder representar la estructura organizativa de los usuarios del sistema. Esta estructura se representa por medio del diagrama OSD del apartado de estructura del sistema.

6.3.2 Identificación y Descripción de Tareas

A partir de los requisitos identificados y descritos en la etapa anterior se identifican y describen las tareas en esta etapa. Tareas que serán modeladas posteriormente por medio de los diagramas adoptados, CTT, para su representación.

Las tareas se describen por medio de la plantilla presentada en la descripción de esta etapa.

Cada requisito funcional identificado en la etapa anterior dará lugar a una tarea en esta etapa. Sin embargo, cada tarea en esta etapa podría desdoblarse en otras

tareas más sencillas hasta llegar a tareas atómicas que no se pueden descomponer en más tareas.

Por ejemplo, el requisito *RF-8 (Editar documento)* descrito en este ejemplo da lugar a la tarea *Edicion_doc* representada en el diagrama de tareas CTT de la Figura 113. Esta tarea, como puede apreciarse, se descompone en otras tareas más sencillas (metadato *Tareas dependientes*). A partir de la plantilla para la descripción de las tareas del sistema, cada metadato propuesto describe la tarea *Edicion_doc* como puede apreciarse en la Tabla 43.

Tabla 43. Descripción de la tarea *Editar_doc* a partir de la plantilla para la descripción de las tareas del sistema

Tarea-18	Edicion_doc
Versión	1 (06/06/2007)
Autores	<ul style="list-style-type: none"> • Victor M. R. Penichet (Investigador)
Fuentes	<ul style="list-style-type: none"> • Descripción textual (caso de estudio ficticio)
Objetivo	<ul style="list-style-type: none"> • La tarea está asociada al objetivo #OBJ-3 (Elaboración síncrona de documentos candidatos a ser publicados)
Requisito	La tarea está asociada al requisito #RF-8 (Editar documento)
Descripción	Se realiza la edición de un documento en tiempo real teniendo en cuenta que puede haber otros usuarios que estén trabajando sobre el mismo documento.
Tarea a la que pertenece	<ul style="list-style-type: none"> • La tarea no es parte de ninguna otra
Objetos	<p>La tarea está relacionada o afecta a los siguientes objetos del sistema:</p> <ul style="list-style-type: none"> • <nombre del objeto> • ...
Modo de Ejecución	Según su modo de ejecución la tarea es del tipo Abstracta
Composición	Según su composición, la tarea es Compuesta
Tarea Compuesta	
Tareas dependientes	<p>La tarea está compuesta por las siguientes:</p> <ul style="list-style-type: none"> • #Tarea-19 (Seleccionar_doc) • #Tarea-20 (Validacion_de_sesion) • #Tarea-21 (Validacion_doc) • #Tarea-22 (Trabajo_en_doc) • #Tarea-23 (Interrumpir)
Descripción como Tarea Compuesta	Es una tarea abstracta directamente relacionada con un requisito que necesita descomponerse en otras tareas para llevarse a cabo. Además, al tratarse de una tarea de muy alto nivel, la mayoría de las tareas en las que se descompone también son compuestas
Tarea de Grupo	
CSCW	<p>Según sus características CSCW, la tarea es de:</p> <ul style="list-style-type: none"> • Coordinación: No • Comunicación: No • Cooperación: Sí • Colaboración: No
Tiempo	<ul style="list-style-type: none"> • Según sus características temporales, la tarea es principalmente Síncrona
Lugar	<ul style="list-style-type: none"> • Según sus características espaciales, la tarea se lleva a cabo en lugares diferentes
Descripción como Tarea de Grupo	<ul style="list-style-type: none"> • Como tarea de grupo, se habrá de tener en cuenta: • Cuando el actor edite un documento debe haber una sección crítica que se bloquee para que ningún otro usuario pueda editar esa sección • Se debe reflejar en tiempo real los cambios en todas las pantallas del resto de actores • Se debe desbloquear la sección cuando no se edite esa zona • Cuando el usuario guarde el documento tras las modificaciones que haya realizado, se manda asincrónicamente una notificación al resto de actores

6.3.3 Estructura

El análisis de la estructura de los sistemas se aborda en esta etapa por medio de los diagramas de clases y el diagrama de la estructura organizativa de los actores del sistema (Organizational Structure Diagram, OSD). El primero de ellos muestra los objetos del dominio y las relaciones entre ellos. Puesto que se trata de un diagrama tradicional de UML no se expresa el ejemplo del caso de estudio en términos de este diagrama.

El segundo de los diagramas permite modelar la distribución de los actores del sistema y cómo se relacionan (en términos estructurales) para constituir los grupos y jerarquías, qué roles desempeñan, etc.

Tanto los actores y sus relaciones como los roles que desempeñan han sido identificados previamente, si bien, la relación entre los roles y los actores se muestra en este diagrama y se tiene en cuenta al considerar la trazabilidad entre las etapas, como se verá posteriormente.

El diagrama OSD de la Figura 112 muestra gráficamente la estructura del ejemplo por medio de los elementos y relaciones organizativas descritas en la etapa.

Se puede apreciar cómo el rol Writer se asocia directamente al grupo AUTHORS, por lo que todos los actores pertenecientes a ese grupo podrán desempeñar las tareas asociadas a ese rol: los usuarios Author y Chair_author. El rol Chair_writer está asociado sólo al usuario Chair_author, que es lo que permite distinguirlo del usuario Author y hacerlo necesario en el sistema porque tiene otras tareas diferentes adicionales que llevar a cabo.

El rol Reader está asociado al grupo que representa a todos los usuario del sistema, puesto que todos ellos pueden leer documentos publicados.

Se aprecia cómo se han organizado los actores creados para facilitar la estructura lógica del sistema: INTERNAL, EXTERNAL y WHOLE_SYSTEM.

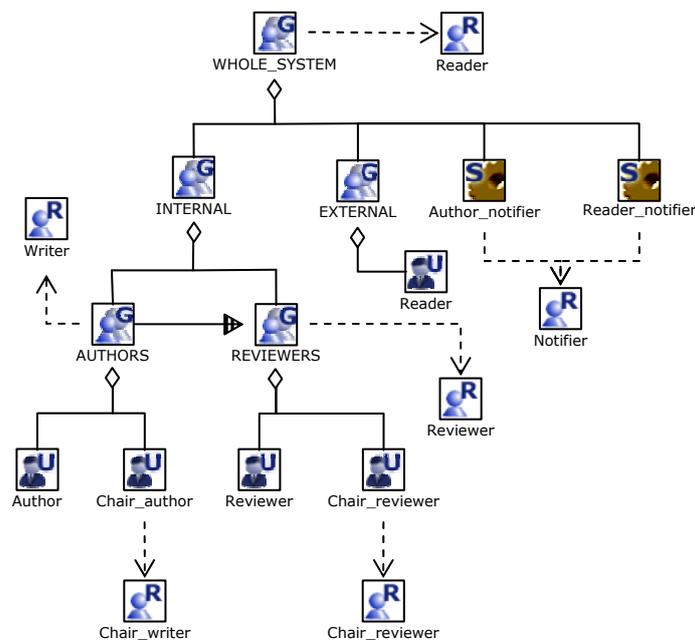


Figura 112. Diagrama de la Estructura Organizativa del sistema del caso de estudio

De un golpe de vista se comprende qué actores pertenecen a qué grupos o qué roles desempeñan cada uno de los actores y permite realizar un análisis más sencillo que simplifique esta estructura.

También puede apreciarse que el grupo AUTHORS depende jerárquicamente del grupo REVIEWERS, puesto que realmente los que tienen la última palabra para la publicación o el rechazo de un documento son los miembros de este último grupo.

6.3.4 Comportamiento

El análisis del comportamiento de los sistemas se aborda en esta etapa por medio del Diagrama de Tareas (Task Diagram, TD) y el Diagrama de Co-Interacciones (Co-Interactions Diagram, CD). El primero de ellos, concretamente CTT que es la notación empleada, permite modelar el trabajo realizado por los actores del sistema en base a las tareas que realizan según los roles que desempeñan.

La Figura 113 muestra las tareas que desempeñaría un actor que juegue el rol *Writer*. Por falta de espacio y por centrar la atención en la tarea *Edición_doc*, no se muestran todas las tareas asociadas a este rol (de ahí los puntos suspensivos entre las tareas abstractas *Autenticación* y *Edición_doc*) y las que se ven no se muestran desplegadas (son en su mayoría tareas compuestas).

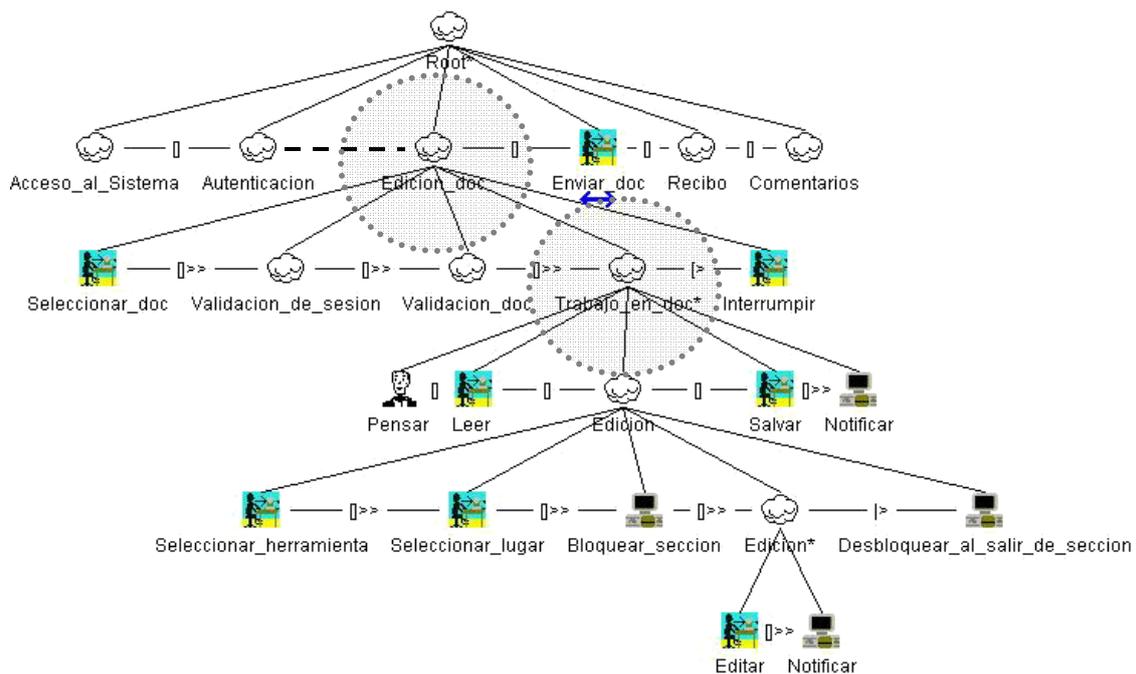


Figura 113. Tareas llevadas a cabo por los actores que juegan el rol *Writer* representadas por medio de un TD

Se puede apreciar cómo aparecen en primer nivel una serie de tareas, la mayoría abstractas, que coinciden con los casos de uso del diagrama de casos de uso anterior: *Acceso_al_sistema*, *Autenticación*, *Edición_doc*, etc. Estas tareas se descomponen en otras hasta que se descomponen en tareas que son atómicas. Las tareas abstractas que no se han desplegado también se descomponen, pero sólo se ha mostrado descompuesta en subtareas la tarea *Edición_doc* por claridad.

Siguiendo este ejemplo, a partir de un requisito se ha identificado directamente una tarea. Es la que se corresponde con la edición del documento. Sin embargo, en el análisis de tareas, ésta no es una tarea atómica sino que para su ejecución es necesario llevar a cabo otras tareas más sencillas como seleccionar el documento que se editará, después el sistema comprueba que se está en una sesión válida y que el documento también es válido, para iniciar la edición real del documento hasta que el usuario termine de editarlo. Esa edición real del documento, representada en la figura por medio de la tarea *Trabajo_en_doc*, conlleva a su vez otras subtareas en las que participa el sistema y el usuario para que se lleve a cabo. En resumen, por medio de los diagramas de tareas se puede especificar cómo será la actividad que realiza el actor que desempeña este rol.

El segundo de los diagramas permite modelar las colaboraciones establecidas entre los actores (individuos, usuarios, agentes, grupos) de un sistema: tareas de grupo. Por lo tanto, por medio de este diagrama se especifican las interacciones entre usuarios a través del sistema, por lo que de un "golpe de vista" se puede apreciar entre quiénes se está llevando a cabo una tarea de grupo.

La Figura 114 muestra un CD del caso de estudio. El actor AUTHORS tiene tres co-interacciones que representan las tareas de grupo *Compartir_borrador*, *Comentar_borrador* y *Edicion_doc*. Para realizar cada una de estas tareas de grupo, es necesario que se realicen tareas individuales por parte de cada actor, tareas que aparecieron en el diagrama anterior. Por ejemplo, en el caso la edición del documento de forma cooperativa, para que tenga lugar la tarea de grupo *Edicion_doc* es necesario que diferentes actores realicen de forma "individual" la tarea *Trabajo_en_doc*, una tarea abstracta que conecta con el diagrama de tareas. Esta tarea de grupo, como se definió en la Tabla 43, es una tarea cooperativa que se realiza de forma síncrona entre actores separados en el espacio. Esto queda reflejado visualmente en el diagrama por medio de la leyenda "--, --, op, S, D" bajo el nombre de la tarea, tal y como se explicó en el apartado 4.2.1.

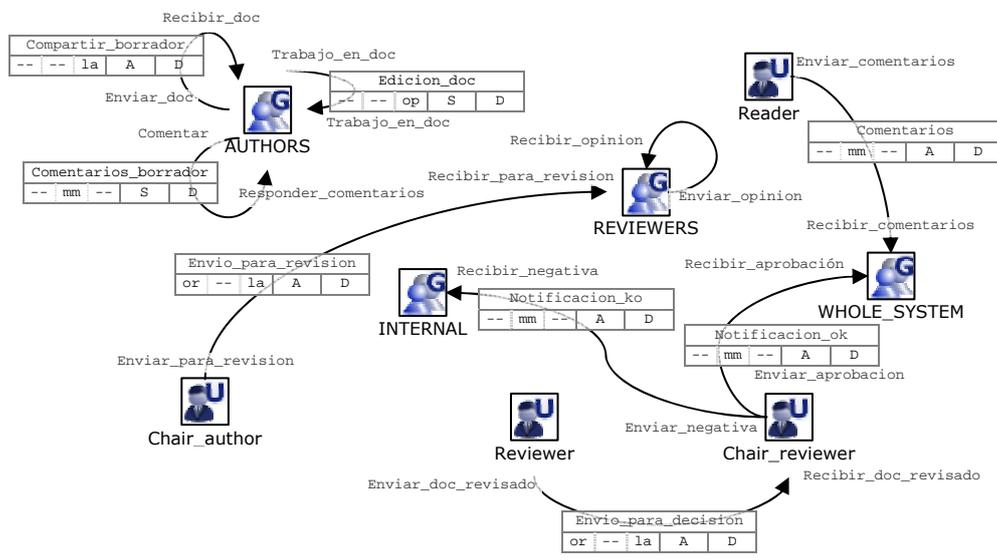


Figura 114. Diagrama de co-interacciones del sistema del caso de estudio

El resto de tareas entre usuarios funciona de un modo similar. Por ejemplo, en el proceso de envío a revisión del documento candidato a ser publicado se ejecuta la tarea de grupo *Envio_para_revision*. Para ello, un usuario *Chair_author* ejecuta la tarea *Enviar_para_revision* y la tarea de grupo se completa cuando los miembros del grupo *Reviewers* ejecutan la tarea *Recibir_para_revision*.

6.3.5 Trazabilidad

El paso de requisitos a tareas y de actores a roles es el hilo conductor que permite la trazabilidad entre las etapas de elicitación de requisitos y análisis. De la abstracción de los usuarios del sistema se han definido los roles y la matriz de trazabilidad de la Tabla 44 pone de manifiesto la relación de los roles con los actores según la asignación realizada en la plantilla de la Tabla 27 para la descripción de los roles.

Tabla 44. Trazabilidad entre actores-roles

	Rol-1 (Writer)	Rol-2 (Chair_writer)	Rol-3 (Reviewer)	Rol-4 (Chair_reviewer)	Rol-5 (Notifier)	Rol-6 (Reader)
#G-1 (AUTHORS)	•					•
#G-2 (REVIEWERS)			•			•
#G-3 (INTERNAL)						•
#G-4 (EXTERNAL)						•
#G-5 (WHOLE_SYSTEM)						•
#U-1 (Author)	•					•
#U-2 (Chair_author)	•	•				•
#U-3 (Reviewer)			•			•
#U-4 (Chair_reviewer)			•	•		•
#U-5 (Reader)						•
#S-1 (Author_notifier)					•	•
#S-2 (Reader_notifier)					•	•

Tabla 45. Trazabilidad entre tareas-requisitos para el objetivo *OBJ-3: Elaboración sincrónica de documentos candidatos a ser publicados*

	RF-1 (Acceso al Sistema)	RF-2 (Autenticación)	...	RF-8 (Editar Documento)	...
...					
Tarea-18 (Edicion_doc)				•	
Tarea-22 (Seleccionar_doc)				•	
Tarea-23 (Validacion_de_sesion)				•	
Tarea-24 (Validacion_doc)				•	
Tarea-25 (Trabajo_en_doc)				•	
Tarea-26 (Interrumpir)				•	
Tarea-30 (Pensar)				•	
Tarea-31 (Leer)				•	
Tarea-32 (Edicion)				•	
...		

Algo similar ocurre con respecto a la relación entre requisitos y tareas. El paso de un requisito en la etapa de elicitación de requisitos a tareas en la etapa de análisis

es una relación 1 a 1. Por lo tanto, en análisis siempre habrá una tarea, probablemente abstracta y compuesta, que lleve a cabo ese requisito. Esa asociación es clara en la Tabla 28 donde se describen las tareas y se muestra explícitamente. Sin embargo, al realizar el análisis de tareas éstas se desglosan en otras cada vez más sencillas hasta llegar a otras que son atómicas. Todas esas tareas no tienen explícitamente un requisito asociado, sino que en su conjunto solucionan el requisito de la tarea padre. La Tabla 45 mostraría esa relación de forma clara y resumida.

Por último, la matriz de trazabilidad mostrada en la Tabla 46 pone en relación las tareas y los roles, mostrando explícitamente la trazabilidad intraetapa: los roles son un conjunto de tareas que pueden desempeñar los actores del sistema. Aquí se muestra qué tareas puede llevar a cabo cada rol. El diagrama de tareas muestra las tareas que desempeña cada rol, pero de esta manera se puede apreciar, adicionalmente, si una tarea la desempeña más de un rol.

Tabla 46. Trazabilidad entre tareas-roles

	Rol-1 (Writer)	Rol-2 (Chair_writer)	Rol-3 (Reviewer)	Rol-4 (Chair_reviewer)	Rol-5 (Notifier)	Rol-6 (Reader)
...						
Tarea-5 (Introducir_nombre)	•	•	•	•		
Tarea-6 (introducir_clave)	•	•	•	•		
...						
Tarea-18 (Edicion_doc)	•					
Tarea-22 (Seleccionar_doc)	•					
Tarea-23 (Validacion_de_sesion)	•					
Tarea-24 (Validacion_doc)	•					
Tarea-25 (Trabajo_en_doc)	•					
Tarea-26 (Interrumpir)	•					
Tarea-30 (Pensar)	•					
Tarea-31 (Leer)	•					
Tarea-32 (Edicion)	•					
...						

6.4 Diseño

En esta metodología el diseño se centra en la navegación por las diferentes interfaces y en la presentación de las mismas, lo que se consigue por medio del

Diagrama de Interacción de Contenedores Abstractos (ACID) y el Diagrama de Interfaces de Usuario Abstractas (AUID) respectivamente.

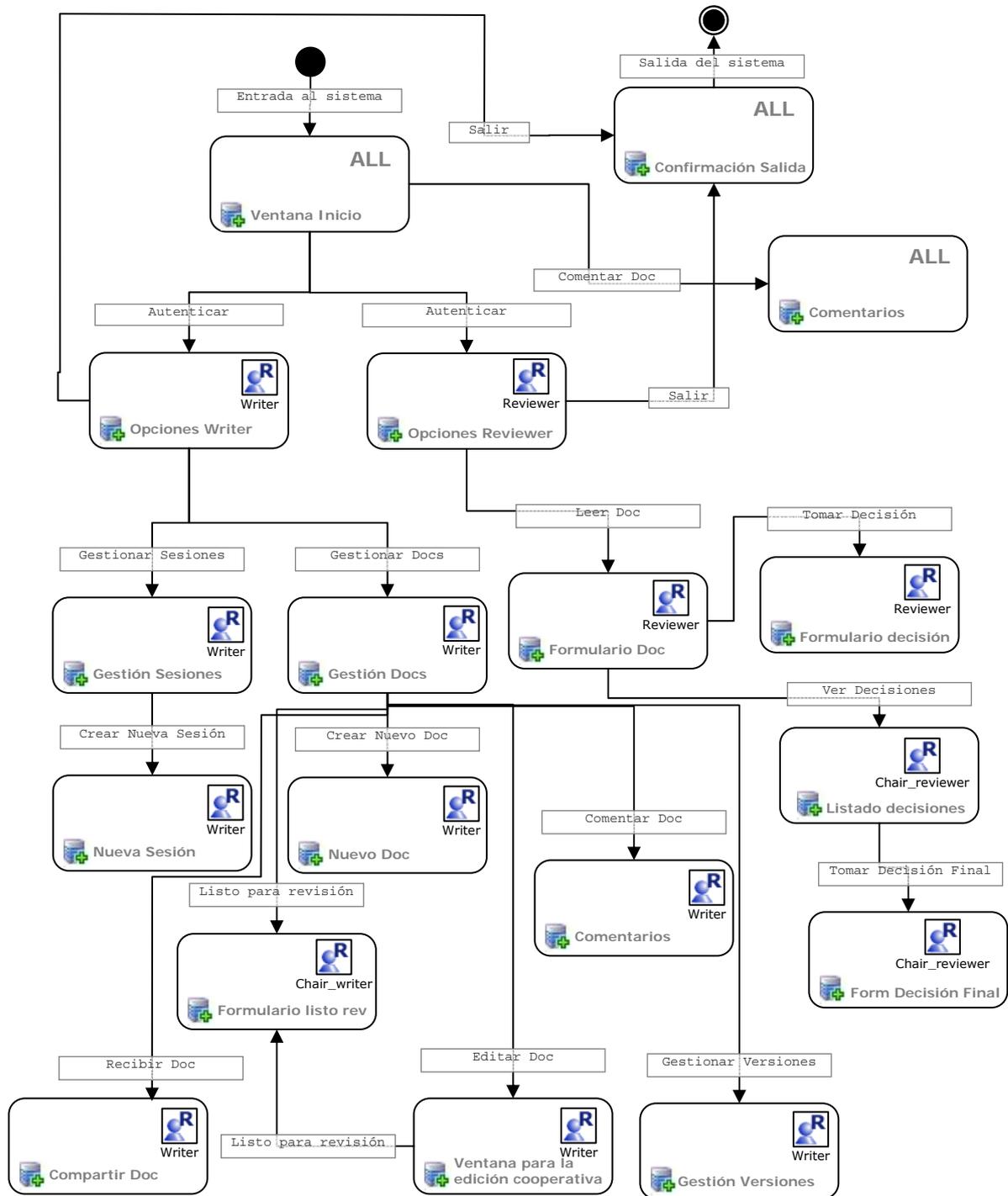


Figura 115. Diagrama de Interacción de Contenedores Abstractos (ACID) del caso de estudio que representa la navegación por las distintas interfaces

La Figura 115 muestra un ejemplo del primero de los diagramas para el caso de estudio presentado. En él se puede apreciar cómo de cada una de las interfaces se puede pasar a otra a través de la ejecución de una tarea determinada.

Lógicamente, no todas las tareas que un usuario puede realizar en un sistema llevan asociado un cambio de "ventana". Para aclarar un poco más este diagrama se explica brevemente la interfaz *Gestión Docs*. A esta interfaz se llega desde la interfaz *Opciones Writer* y sólo es alcanzable por los actores que desempeñen el rol *Writer*, es decir, un usuario del grupo *AUTHORS* (pues son los que desempeñan el susodicho rol) selecciona alguna opción en esa interfaz que ejecuta la tarea *Gestionar Docs*. La ejecución de esa tarea le llevará a una nueva ventana (*Gestión Docs*) que contiene toda la información y opciones relativas a la gestión documental que puede visualizar este tipo de actores. El diseño de esta interfaz se puede ver en la Figura 116, donde se muestran una serie de AIOs algunos de los cuales podrán hacer que se cambie a otra interfaz.

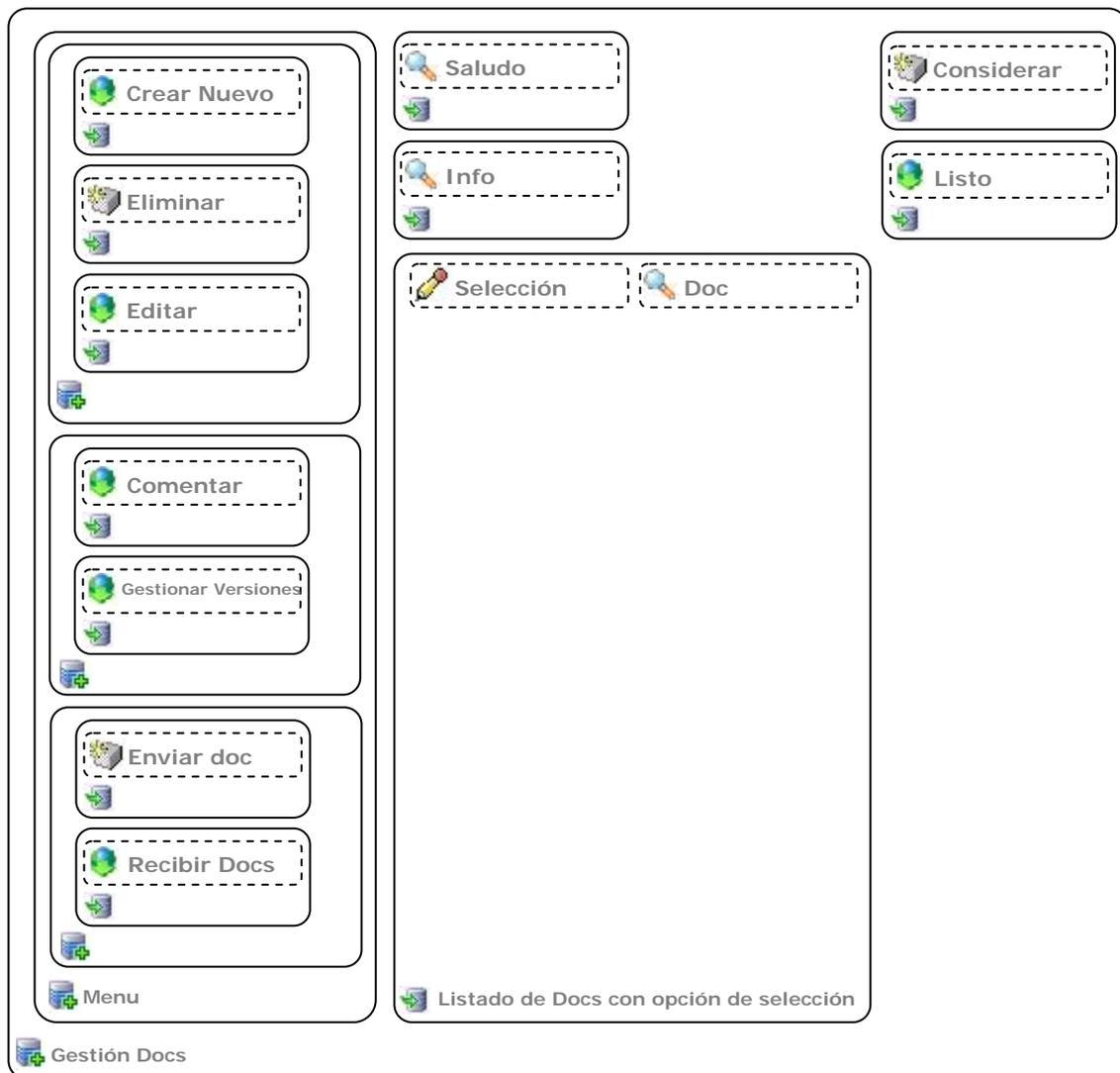


Figura 116. Diseño de la interfaz *Gestión Docs* del caso de estudio por medio del Diagrama de Interfaces de Usuario Abstractas (AUI)

En esta interfaz, la mayoría de los AIOs están caracterizados con facetas de navegación  puesto que se trata de una ventana desde donde el usuario podrá acceder a otras para realizar funciones específicas. Estos AIOs están asociados a tareas que permiten navegar de un sitio a otro como también se puede apreciar en la figura anterior. Otros AIOs están caracterizados con facetas de control  puesto

que están asociados a tareas que realizarán una función que además no cambiará de ventana al actor. Por ejemplo, el AIC *Enviar doc* estará asociado a la tarea (del mismo nombre, lógicamente) que aparece en la Figura 113 y en la Figura 114, y que es parte de la tarea de grupo *Compartir_borrador*. Esta tarea permite enviar los documentos que se hayan seleccionado en el AIC central (*Listado de Docs con opción de selección*) a los actores que participen de la elaboración del documento. Este último AIC está caracterizado con dos facetas puesto que además de mostrar una lista de los documentos permite realizar la selección de algunos de ellos para poder ejecutar acciones sobre dicha selección.

Como se puede apreciar, hay un AIC cuyo nombre es *Listo* (para revisión) que se corresponde con una acción que sólo un actor que desempeñe el rol *Chair_writer* podría ejecutar, sin embargo, según el diagrama de la figura anterior esta interfaz la visualizan los actores que desempeñan el rol *Writer*. Esta tarea por tanto la visualizarían todos pero sólo podrían ejecutarla los actores que desempeñen el rol *Chair_writer*. Un actor de estas características sí podría acceder a la siguiente interfaz por medio de esta tarea asociada al AIC.

La Figura 117 muestra el diseño de la interfaz *Ventana para la edición cooperativa* del caso de estudio por medio del Diagrama de Interfaces de Usuario Abstractas (AUID). Esta interfaz es la que permite a los actores que desempeñan el rol *Writer* editar un documento de forma cooperativa.

Esta interfaz dispone de un chat para mantener comunicados a los usuarios del sistema. Básicamente el chat dispone de dos AICs, uno caracterizado por medio de una faceta de tipo input  para la introducción del texto que un usuario quiera escribir y otro caracterizado por una faceta de tipo control  para poder enviar el texto escrito al resto de usuarios.

Este texto se visualiza en una zona común, puesto que todos los usuarios que participan del chat la visualizan por igual, por lo que se representa por medio de un AWAC . Para poder tener una idea de lo que se realiza en el resto de usuarios del sistema existe otro AWAC: el radar. Este elemento posee una faceta del tipo visibility  y permitirá mostrar una vista de todo o parte del espacio de trabajo en "miniatura" para poder tener una visión general del mismo.

A la derecha de la interfaz *Ventana para la edición cooperativa* hay otro AWAC que muestra los usuarios del sistema, algo similar al cuadro de usuarios del sistema mostrado en el apartado 5.5.3.

El editor en sí ocupa la parte central de la interfaz y está representado por un AWAC. Este elemento tiene una serie de controles por medio de los cuales los usuarios podrían editar el documento (poner la fuente en negrita, cambiar el tipo, insertar imagen, etc.), visualizar y trabajar sobre la pantalla de edición y visualizar el telepuntero de cada uno de los usuarios que estén trabajando. Puesto que tanto los controles como la pantalla de edición están dentro del AWAC (además del telepuntero), esto significa que ambos AIOS forman parte del entorno cooperativo, es decir, puede parecer obvio que en la pantalla de edición se puede apreciar el trabajo de cada uno de los usuarios, pero además se ha considerado que también los botones del editor están dentro de esa zona compartida. Esto se debe a que se ha introducido la posibilidad de que si se hace uso de esos controles, el resto de usuarios puedan "ser conscientes" de ello. Esto se representa además caracterizando el AIO correspondiente por medio de la faceta del tipo expressive artifact . Esta faceta también caracteriza el AIO de la pantalla puesto que se informa a los usuarios de algunas acciones del resto de usuarios, como podría ser que hay una sección crítica establecida en un párrafo que está editando o que ha

eliminado una imagen. El AIO que representa los telepunteros se caracteriza por medio de la faceta de tipo embodiment .

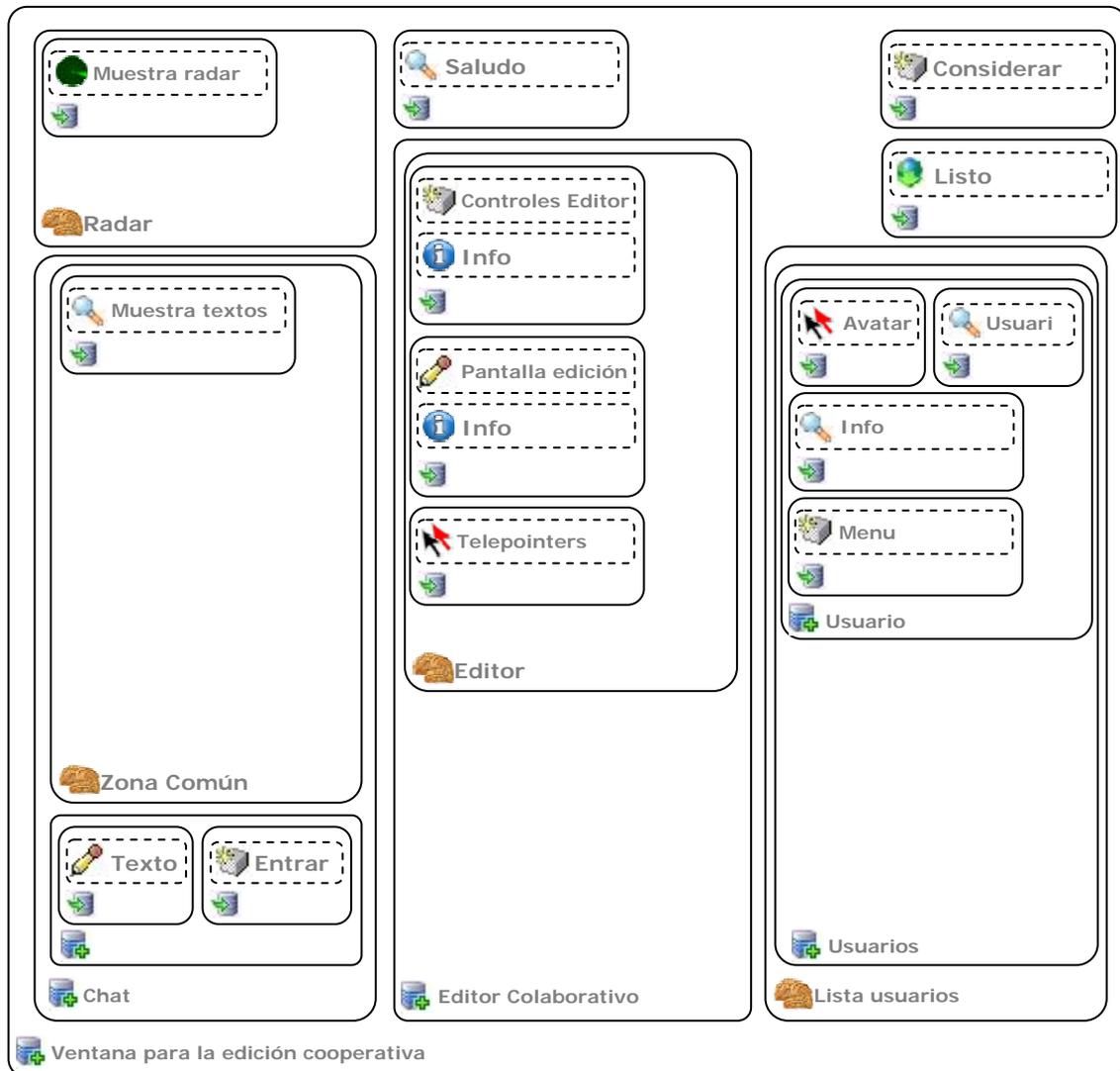


Figura 117. Diseño de la interfaz *Ventana para la edición cooperativa* del caso de estudio por medio del Diagrama de Interfaces de Usuario Abstractas (AUID)

Tanto el ACID como el AUID contienen elementos provenientes de la etapa de análisis anterior. En el primero de ellos, el paso de una ventana, de un contenedor a otro se hace por medio de alguna tarea; además, cada contenedor muestra el rol que han de tener asociados los actores que pueden entrar en el mismo. Por otro lado, en el AUID, los componentes individuales abstractos descritos por medio de facetas (los AICs) están relacionados con tareas. Por ejemplo, el AIC caracterizado con la faceta *input* "Texto" tendrá asociada una tarea de *interacción* que permite al usuario introducir el comentario que quiere hacer llegar al resto de actores.

6.5 Implementación

La etapa de implementación del sistema contempla la reificación de los AIOs de la etapa de diseño en CIOs para formar la interfaz de usuario concreta o CUI en primera instancia y terminar generando la FUI o interfaz de usuario final.

Puesto que el trabajo presentado en esta tesis es el desarrollo de interfaces de usuario para entornos de trabajo en grupo y no está centrada en que se puedan generar de manera independiente a la plataforma, en principio no se ha considerado una representación de la CUI (que sería independiente de la plataforma) y se realiza el proceso en un mismo paso. Otros trabajos basados en Cameleon y que sí están centrados en la posibilidad de generar interfaces de usuario independientes de la plataforma utilizan herramientas como GrafiXML, empleada en UsiXML para generar este tipo de interfaces independientes de la plataforma. El resultado obtenido como CUI es muy similar a las FUI, por este motivo, la generación de la interfaz de usuario final en esta tesis se realiza desde los AIOs de la etapa de diseño directamente, pero si se quisiera hacer independiente de la plataforma, debería tener en cuenta este paso del mismo modo.

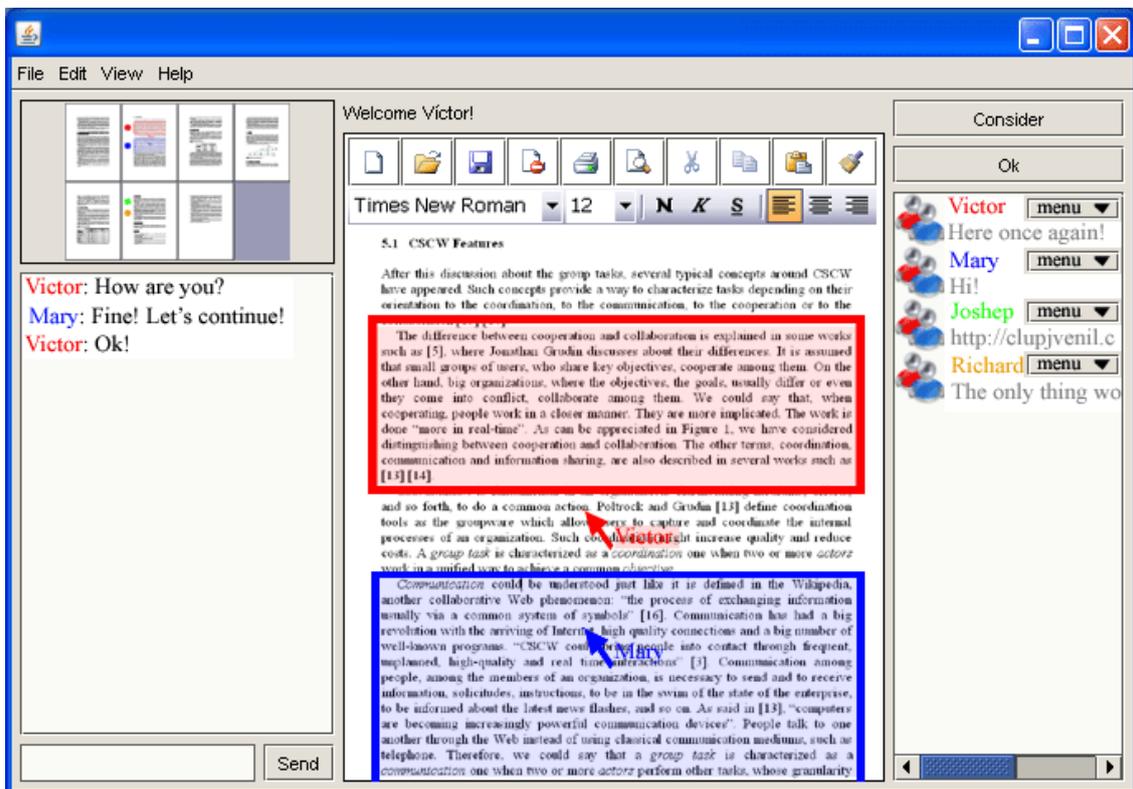


Figura 118. Interfaz de usuario final para de la edición cooperativa de un documento. FUI a partir de la AUI *Ventana para la edición cooperativa*

La Figura 118 muestra cómo podría ser la interfaz de usuario final del editor cooperativo de documentos del sistema según la interfaz abstracta presentada en la etapa anterior. Merece la pena destacar cuatro puntos principales:

- El radar, que muestra como se decía en la etapa anterior esa visión del espacio de trabajo en su conjunto para dar una idea del "todo".
- El chat como sistema de comunicación tradicional en tiempo real.

- El cuadro de usuarios del sistema ya discutido con anterioridad. Se pueden observar elementos como la imagen representativa del usuario (avatar), su nombre, información relativa a él como una frase que haya escrito, y un menú con las opciones que se pueden realizar.
- El editor cooperativo muestra los punteros de los usuarios que trabajan en él, sombrea de colores diferenciados y con marcas de agua las zonas donde están trabajando otros usuarios (secciones críticas), realiza marcas de agua temporales sobre elementos que otros usuarios han eliminado (como imágenes), muestra durante varios segundos un botón sombreado de otro color semitransparente si ha sido pulsado por otro usuario, etc. Todos los colores empleados siempre son los mismos para cada usuario y sus acciones.

CAPÍTULO 7.

TOUCHE CASE TOOL: UNA HERRAMIENTA CASE DE SOPORTE A LA METODOLOGÍA

Los modelos de proceso y metodologías propuestos tradicionalmente en la Ingeniería del Software muestran su potencia real cuando hay una herramienta CASE que les da el soporte necesario para llevar a cabo los proyectos desde el inicio, automatizando ciertas tareas que hacen más sencilla la especificación del sistema.

Ayuda a mantener la coherencia del sistema a lo largo de todo el desarrollo, pero sobre todo realiza de forma automática un conjunto de acciones que de modo manual serían repetitivas, muy laboriosas y podrían llevar a confusión.

Esta automatización disminuye el trabajo de los analistas y demás participantes en el proceso de desarrollo de la aplicación CSCW, generando código desde la misma etapa de elicitación de requisitos.

7.1 Introducción

En Ingeniería del Software el término *herramienta CASE* es bien conocido. Una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) es una aplicación software que ayuda a llevar a cabo el desarrollo de sistemas informáticos de forma ordenada, siguiendo una serie de pasos que mantienen la coherencia de la especificación en su globalidad.

Las aplicaciones actuales requieren una gran cantidad de información que a menudo es difícil de manejar. Documentación, especificaciones de requisitos, diagramas, tablas, elementos multimedia, en algunos casos, que complementan la labor de recolección de la información necesaria para especificar el sistema, análisis de datos en las distintas etapas del ciclo de vida del software, etc. y lo que es más importante, la relación de toda esa información.

Muchas veces, la información que se ha de mostrar parece redundante porque de alguna manera aparece en otro lado de la especificación, sin embargo, resulta de vital importancia que se muestre explícitamente para evitar que el analista, el desarrollador o la persona encargada en ese momento tengan que volver a calcularla. Hacer esto de forma manual es realmente costoso en tiempo y dinero pero, lo que es más peligroso, la posibilidad de cometer errores que comprometan el sistema en su totalidad es muy alta. Las herramientas CASE facilitan al equipo de trabajo esta labor automatizando algunos procesos, mostrando información desde diferentes perspectivas, relacionando información fácilmente y sobre todo manteniendo la coherencia global del sistema.

Todo ello redundando en un aumento considerable de la calidad del software desarrollado disminuyendo tiempos y costes. Al tratarse de un proceso automatizado, lógicamente se facilita tanto el trabajo de diferentes personas sobre un mismo proyecto, como el mantenimiento actual y posterior de la aplicación.

En este capítulo se presenta una herramienta CASE, TOUCHE CASE Tool, que asiste al desarrollo de interfaces de usuario de aplicaciones groupware de acuerdo con las ideas planteadas en la propuesta metodológica de esta tesis. De esta manera se pone de manifiesto que las ideas planteadas y el modelo de proceso en su totalidad es aplicable y válido para el desarrollo de estos interfaces.

7.2 Herramientas CASE Actuales

Actualmente existen en el mercado numerosas herramientas CASE muy conocidas y muy utilizadas. Algunas de las más empleadas son *Rational Rose* [Rational, Web], *MS Visio* [MSVisio, Web], *EclipseUML* [EclipseUML, Web] o *Enterprise Architect for UML* [Enterprise Architect, Web].

Todas ellas comparten una serie de características comunes como el soporte de los nueve diagramas especificados por *UML 2.0* y algunos otros tipos de diagramas adicionales, la generación de código en diversos lenguajes o la ingeniería inversa; la optimización de código la generación de modelos y documentación en formato *HTML* y *XMI* y la compatibilidad e integración con otras herramientas CASE: sistemas de control de versiones, *IDEs* (*Eclipse*, *MS Visual Studio*, etc.).

La versión *Enterprise Edition* de *Rational Rose* incorpora algunas características expresamente creadas para equipos de desarrollo:

- Permite el desarrollo paralelo de un modelo soportando la descomposición del modelo en unidades versionables, llamadas unidades de control.
- Permite que los artefactos sean movidos o copiados entre los espacios de trabajo usando el mecanismo de mapeo de camino virtual.
- Permite a equipos gestionar sus modelos en relación con otros artefactos de proyectos, mediante la integración de sistemas de control de versiones estándar. Rational Rose proporciona una alta integración con ClearCase y Microsoft Visual SourceSafe.
- Proporciona una herramienta separada, llamada Model Integrator, para comparar modelos y fusionar modelos y sus unidades de control.

MS Visio incorpora algunos métodos para permitir la colaboración entre miembros de un mismo equipo. Básicamente, el soporte a la colaboración se realiza mediante la integración con *Microsoft SharePoint Portal Server*, lo que permite publicar diagramas en el espacio de trabajo de esta herramienta. También incorpora un sistema que permite realizar marcas personales dentro de los diagramas (comentarios, anotaciones, tinta digital) y realizar un seguimiento sobre las marcas que otros usuarios van añadiendo a los diagramas.

Enterprise Architect for UML ofrece soporte para el trabajo en grupo mediante un sistema de proyectos compartidos a través de archivos replicados que permite a los usuarios trabajar en un mismo modelo.

Otros editores más recientes como *Poseidon UML* [Poseidon, Web], *Kones* [Hurwitz, 2001], *CO2DE* [Meire, 2003] y *COAST* [Schümmer, 2001] [Mehra, 2004] han sido diseñados para facilitar a grupos de usuarios el desarrollo de proyectos de forma cooperativa. *Poseidon UML* es un ejemplo de editor comercial que facilita un alto grado de colaboración. *Kones* proporciona una plataforma para que equipos de desarrollo puedan trabajar de forma conjunta en las fases de planificación, análisis y diseño del proceso de desarrollo. *CO2DE* y *COAST* son prototipos desarrollados por investigadores de las universidades Federal de Río de Janeiro y La Plata respectivamente.

En el mercado existen otras muchas herramientas CASE para el desarrollo de aplicaciones software, lo que pone de manifiesto la relevancia que tienen para los desarrolladores: AllFusion ERWin, ArgoUML, Blue Ink, BPWin, CASE Studio 2, CASEwise, Database Designer for MySQL, DBDesigner 4, DMS Software Reengineering Toolkit, EasyCase, Embarcadero ER/Studio, eREQUIREMENTS – eRequirements, GeneXus, GNU Ferret, INNOVATOR, iRise, IRqA, MagicDraw, Modelistic, Obsydian / Plex, Oracle Designer, Rational ClearCASE, SILVERRUN, swREUSER, Sybase PowerDesigner, System Architect, Together, Topcased, Umbrello, Visible Enterprise Products, Visual Paradigm for UML, Xcase Database Design Software.

7.3 Una Aproximación al Desarrollo mediante Herramientas Cooperativas: CE4Web

La mayoría de los editores UML del mercado no permiten el desarrollo de diagramas de forma cooperativa en tiempo real. En este apartado se muestra la propuesta desarrollada como una aproximación al desarrollo de sistemas mediante herramientas CSCW, que plasma la experiencia que se ha adquirido en este campo

y que ha permitido comprender mejor las necesidades de colaboración necesarias en estos entornos. Concretamente, el prototipo que se ha implementado permite realizar diagramas de clases UML de forma cooperativa. Diferentes usuarios se pueden conectar al mismo proyecto para realizarlo en tiempo real a la vez, manteniendo un cierto awareness de lo que hacen el resto de usuarios en el sistema en cada momento.

CE4Web (Cooperative Editor for the Web) [Martínez, 2006; Penichet, 2007c], la herramienta presentada en este apartado, es un editor UML cooperativo síncrono de diagramas, con una interfaz WYSIWIS. Se trata pues de una aplicación multiusuario para generar diagramas en tiempo real, lo que conlleva una serie de requisitos como: respuesta inmediata, tiempos de notificación cortos y la implementación de un sistema de soporte a la colaboración que proporcione los servicios de comunicación, coordinación, control de concurrencia y consistencia de datos básicos para permitir el proceso de cooperación síncrona. El proceso comunicativo de los usuarios se facilita por medio de un sistema de mensajería interna tipo Chat, para dar flexibilidad a la comunicación dentro del grupo.

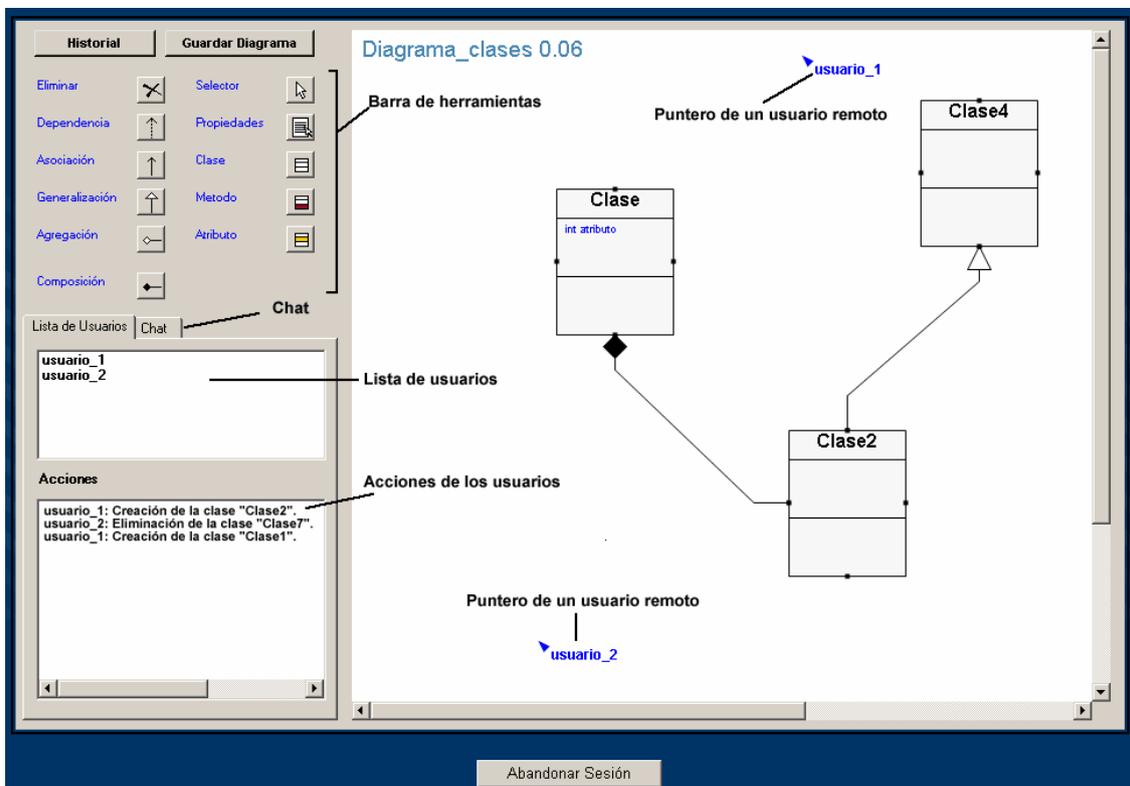


Figura 119. Captura de una sesión donde participan varios usuarios en CE4Web para elaborar un diagrama de clases

Otro de los aspectos a tener en cuenta es el mantenimiento de un repositorio o espacio de memoria de grupo, donde almacenar no sólo los artefactos generados (diagramas UML en este caso), sino toda la información generada en cada una de las sesiones de modelado. Así mismo, un sistema de versionado permite almacenar todos los estados de un diagrama, desde su creación hasta su publicación, sin perder las modificaciones realizadas en cada instante. También se almacenan las conversaciones mantenidas entre los autores durante la sesión de modelado.

El editor propuesto se trata de una herramienta cooperativa basada en Web que permite la creación, modificación y mantenimiento de diagramas de clase UML de forma síncrona y distribuida en el espacio. Se ha adoptado la Web como el entorno

desde el cual el usuario se comunicará con el sistema. Algunos autores consideran que la industria de la Ingeniería Software debe empezar a cambiar su estilo de trabajo y aprovechar las ventajas de la Internet para pasar de un modelo centralizado de trabajo a uno descentralizado basado en Web [Wang, 2001].

Un entorno Web proporciona, entre otras ventajas, accesibilidad y la portabilidad a la herramienta. Además, ninguno de los editores UML colaborativos estudiados son aplicaciones basadas en Web, sino herramientas de escritorio, lo que hace a este sistema más novedoso. La interfaz Web es pues el punto de acceso a la herramienta, a los componentes o subsistemas asociados:

- Editor UML Cooperativo que permitirá la conexión de múltiples usuarios a una misma sesión en la que se realizará la creación o modificación de un diagrama de forma cooperativa.
- Sistema de control de versiones que daría soporte a los procesos de mantenimiento y gestión de los diagramas. De esta manera se facilita el mantenimiento de la memoria del trabajo del grupo, no solamente entendida como el conjunto de documentos generados, sino como todo el conocimiento generado a lo largo de la vida de un proyecto
- Sistema de gestión de usuarios que permitirá el registro de nuevos usuarios en la aplicación y la creación de grupos de trabajo y proyectos.
- Sistema base para los procesos de cooperación síncronos del editor. Se trata del subsistema encargado de dar soporte a los procesos de comunicación, coordinación y cooperación que permiten a los usuarios trabajar de forma síncrona en una sesión de modelado con el Editor UML manteniendo la coherencia e integridad de los datos.

7.3.1 Arquitectura de CE4Web

Según algunas aproximaciones aportadas por investigadores del área [Meire, 2003] [Lee, 2000], se podría adoptar una arquitectura distribuida, en la que no existiera un repositorio común, sino que cada usuario trabajara con una copia local de la aplicación y de los datos, manteniendo de alguna forma sincronizados a todos los usuarios de una misma sesión. Sin embargo, se pretende no limitar la aplicación a la creación del editor, sino que se trata de crear un entorno de trabajo compartido, en el que diferentes usuarios puedan cooperar y trabajar en grupo para la creación, gestión y mantenimiento de diagramas UML.

Podría parecer entonces que una aproximación cercana a una arquitectura centralizada (cliente/servidor) sería idónea para este sistema. Las arquitecturas centralizadas se suelen utilizar para implementar el acceso y manipulación simultáneos de diversos usuarios a un conjunto de datos o documentos, como es el caso de las herramientas de control de versiones o gestión documental. De esta forma se consigue una total portabilidad y disponibilidad de la herramienta.

No obstante, la elección de la Web como plataforma conlleva algunos problemas que se han de tener en cuenta. Sí es cierto que se trata de un buen medio para proporcionar mecanismos de comunicación y colaboración asíncronos, pero no es tan sencillo implementar cooperación síncrona y menos cuando los procesos requieren de una gran interacción por parte de los usuarios.

Finalmente, la opción adoptada es el uso de un applet, una aplicación de escritorio embebida en una página Web y que se ejecuta en la máquina del cliente. De esta forma, se aprovechan las ventajas de ambos medios. Por un lado, el editor no tiene limitada su funcionalidad, ya que aunque utilice la Web para llegar al cliente, no

deja de ser una aplicación de escritorio. También se aprovechan los recursos de la máquina del cliente para reducir tráfico y procesamiento en el servidor. Por otro lado, la disponibilidad y portabilidad de las aplicaciones Web hacen que el editor esté disponible en cualquier máquina que cuente con conexión a Internet, sin necesidad de tener instalado ninguna copia de la aplicación. Al tener una sola instancia física de la aplicación, cualquier modificación o cambio es transparente al usuario.

7.3.2 Interfaz de Usuario

El diseño apropiado de aplicaciones cooperativas debe considerar los mecanismos y esquemas necesarios para ofrecer un alto grado de contexto y conocimiento compartido para permitir la cooperación de los usuarios [Antunes, 1994]. La interfaz de la aplicación debe proporcionar recursos de percepción de las acciones de los demás participantes, coordinación de sus acciones e identificación de su disponibilidad [Greenberg, 2000]. En la propuesta planteada en este trabajo, los usuarios trabajan sobre el mismo conjunto de datos, con lo que en todo momento visualizan la misma información y su interfaz presenta los mismos elementos.

A la funcionalidad y apariencia básica de los editores UML convencionales que se presupone en toda aplicación desarrollada con estas características se la han añadido algunos recursos de percepción del trabajo del propio usuario y de las acciones del resto de participantes:

- Un historial de las acciones que se han realizado durante la sesión.
- Un Chat con el que comunicarse con los demás.
- La posición de los punteros de los demás usuarios.

Cuando un usuario modifica un elemento del diagrama, éste es bloqueado por el sistema y no se permite ninguna operación sobre él hasta que el usuario termine de realizar su acción. El resto de usuarios son alertados de esta situación.

7.4 Implementación de TOUCHE CASE Tool

El sistema está implementado en *Java*, y está dividido en dos subsistemas principales que son el modelo de dominio y la vista / controlador. En este caso, tanto la vista como el controlador forman un solo subsistema ya que no es necesario especificar controladores especiales en el sistema (ver diagrama de paquetes del sistema en la Figura 120).

La vista está basada en *SWING* y tanto los controladores como los elementos gráficos utilizan este framework. El paquete *View* depende de *Swing* y a su vez es la base de los elementos gráficos de la interfaz. En este paquete se encuentran los formularios, las ventanas y diálogos de la aplicación. En paquetes separados se encuentran los elementos gráficos pertenecientes a los diagramas del sistema y a las matrices que el sistema utiliza para especificar la trazabilidad del modelo.

El modelo de dominio tiene una estructura similar a la vista y contiene las clases relacionadas con el modelo utilizado para representar la metodología. De él dependen los paquetes que agrupan los modelos de los diagramas y las posibles representaciones del modelo del documento de requisitos del sistema. En este caso se ha optado por una implementación en *HTML*, sin embargo se puede, de manera sencilla, extender el sistema para que genere archivos *LaTeX* que pueden ser las fuentes de posibles archivos *PDF*.

Para establecer los contratos y desacoplar entre el modelo y la interfaz se han utilizado clases que se encuentran en el paquete *App* que auspician de "Adapters" entre el modelo y la interfaz.

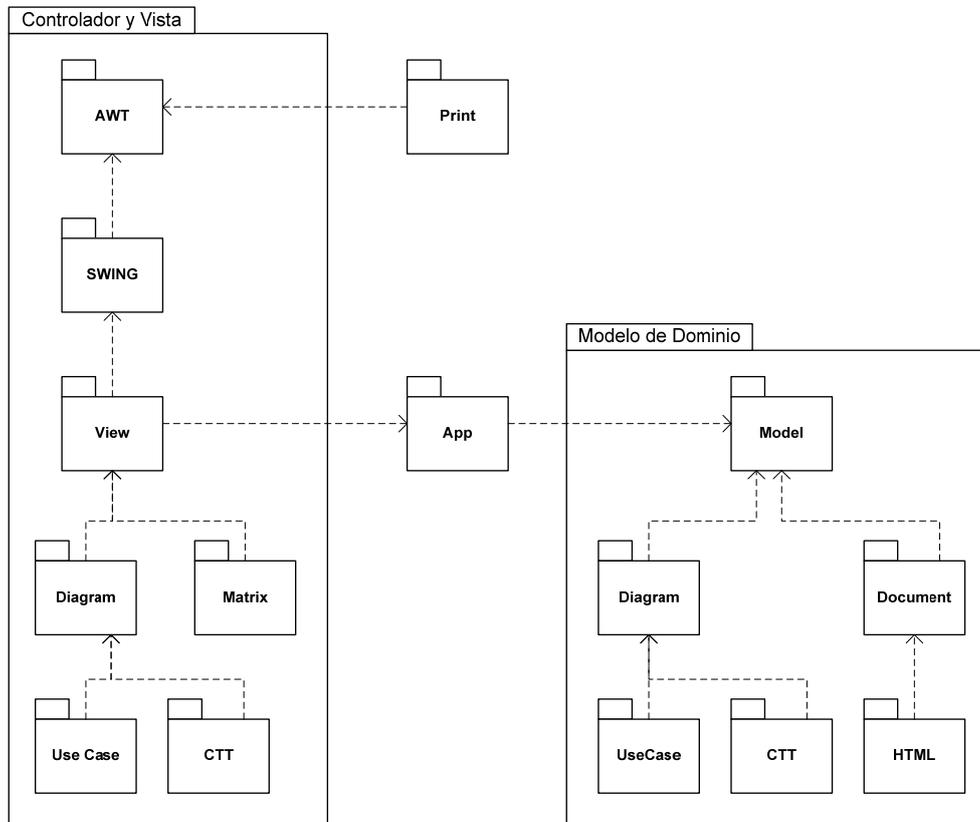


Figura 120. Diagrama de paquetes del sistema

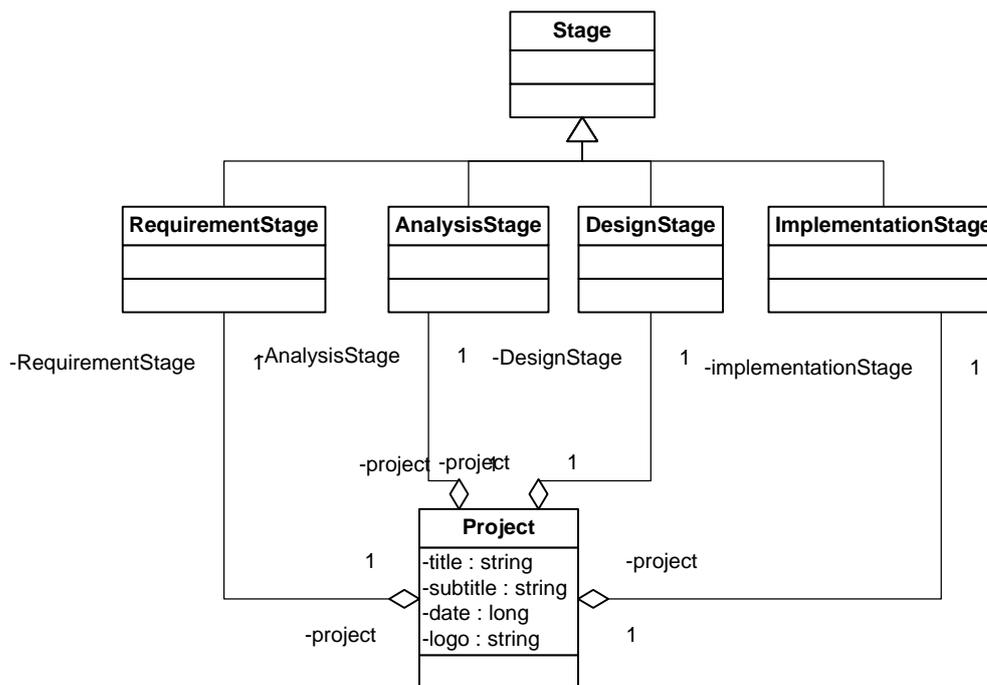


Figura 121. Diagrama del modelo de dominio de la aplicación

7.4.2 Elicitación de Requisitos

La Figura 122 ilustra la estructura y relación con los diferentes elementos que componen la etapa de elicitación de requisitos (RequirementStage).

Esta etapa está compuesta por cuatro entidades básicas que son el *conocimiento del dominio* (*DomainKnowledgeStageStep*), los *requisitos* (*Requirements*), los *objetivos* (*Objectives*) y la *estructura organizativa* (*OrganizationStructure*):

- El *conocimiento del dominio* se representa como una colección de URLs que contienen la información recopilada sobre el dominio de aplicación del sistema. Cada URL se representa mediante un *recurso* (*Resource*) que contiene un *nombre de identificación*, una *descripción* y la *URL* en cuestión.
- Los *requisitos* (*Requirements*) están organizados según su tipo; de *información* (*InformationRequirement*), *funcionales* (*FunctionalRequirement*) y *no funcionales* (*NonFunctionalRequirement*). Todos los objetivos son subclase de la clase *Requirement* que implementa la interfaz *GeneralTemplateModel*. Más adelante en el documento describiremos más en detalle los requisitos.

GeneralTemplateModel es una clase abstracta/interfaz que permite la reutilización de las interfaces gráficas comunes a los objetivos y a los requisitos. Esta entidad contiene la información relacionada con las características CSCW y permite asociar requisitos tanto a objetivos como a otros requisitos.

- Los *objetivos* (*Objectives*) son parte de la elicitación de requisitos y comparten en gran parte la estructura de los requisitos. Sin embargo, éstos permiten la definición de objetivos superiores e inferiores generando una jerarquía de objetivos.
- Finalmente, la elicitación de requisitos permite definir los actores del sistema y la relación que existe entre ellos. Estos actores están estructurados por sus tipos y se encuentran organizados dentro de la *estructura organizativa* que está representada por la clase *OrganizationStructure*, la cual detallaremos más adelante.

El siguiente diagrama (Figura 123) nos muestra la estructura de clases de un *requisito funcional* (*FunctionalRequirement*). Como se vió anteriormente, este requisito comparte la mayoría de la estructura de todos los requisitos, sin embargo posee información adicional que nos permitirá relacionar este tipo de requisitos con los diagramas de casos de uso.

Un *requisito funcional* además de contener todas las características de un requisito, permite especificar otras características como la *frecuencia* en que se necesita este requisito (*Frequency*) y las *restricciones de desempeño* del mismo (*Performance*). También, permite especificar tanto las *precondiciones* como las *post-condiciones* del mismo. Sin embargo, una de las características más salientes de este tipo de requisito es la especificación tanto de la *secuencia normal de ejecución* (*SequenceStep*) como una *secuencia excepcional* de ejecución del mismo (*ExceptionStep*). En ambos tipos de secuencia se pueden especificar *acciones llevadas a cabo por los actores* (*ActorException*, *ActorStep*) o *ejecuciones de casos de uso* (*FunctionalRequirementStep*, *FunctionalRequirementException*). En el caso de los *SequenceSteps* se ha implementado la condición mediante la utilización del patrón de diseño *NullObject* ya que este tipo de secuencia es la que se utiliza para la confección del *diagrama de caso de uso* (se verá más adelante).

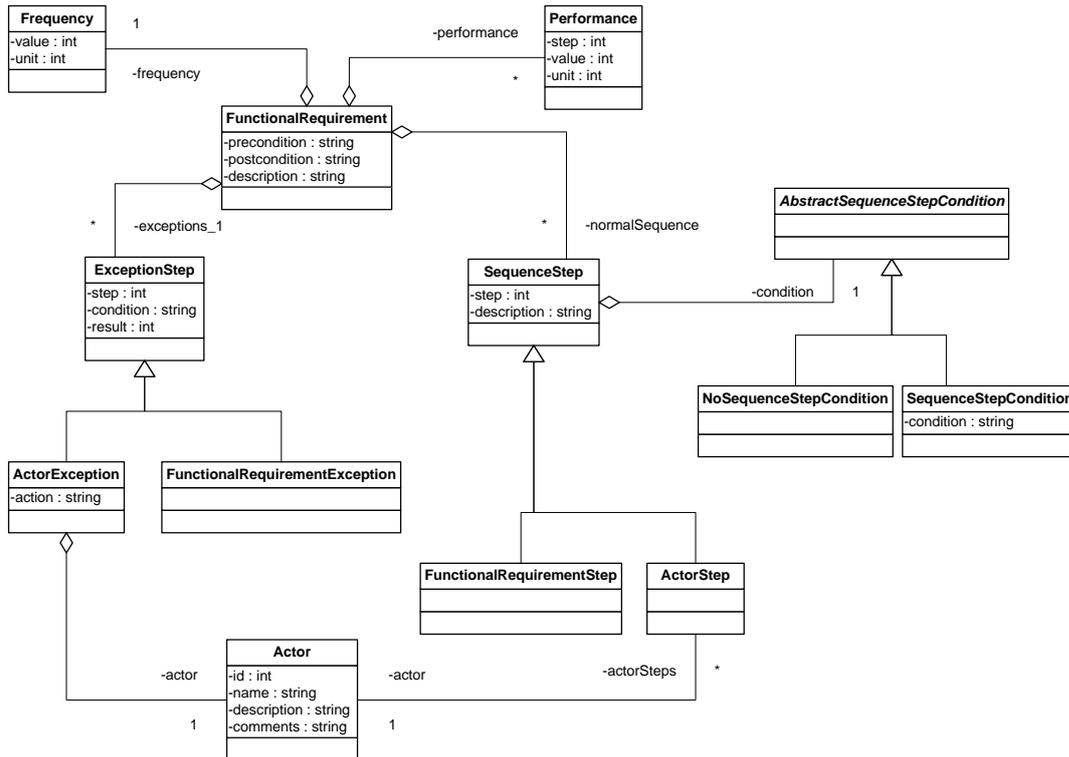


Figura 123. Estructura de clases del requisito funcional

A continuación se muestra el diagrama de clases que representa el modelo de la estructura organizativa de la aplicación (Figura 124). La *estructura de la organización* se basa en la definición y relación que existe entre los diferentes *actores* que participan en la misma. Por lo tanto, la entidad clave en este diagrama es *Actor*. *Actor* define una jerarquía que fue descrita en este documento. Tanto *Actor* como *OrganizationStructure* realizan/implementan la interfaz/clase abstracta *Versionable* que permite la identificación tanto de los *autores* como de las *fuentes* de los datos introducidos en el sistema.

Actor es la clase base de todos los actores del sistema: *grupos* (*Group*), *individuos* (*Individual*), *usuarios* (*User*) y *agentes* (*Agent*). Todos los actores definen relaciones binarias entre ellos. Estas *relaciones binarias* están representadas por las subclases de *BinaryRelationship*. *BinaryRelationship* relaciona dos actores (*actor1* y *actor2*) de acuerdo a una semántica representada por la clase concreta que la implementa.

Los actores también pueden tener asociado un *rol* (*Role*) esta relación se establece en la etapa de Análisis del sistema. Un actor también define *capacidades* (*Capabilities*) que pueden ser tanto *Habilidades* (*Abilities*) como *responsabilidades* (*Responsibilities*). Finalmente se tiene un *actor* que representa un *grupo* de actores (*Group*). Esta entidad también define las *leyes* por las cuales se rige el *grupo* y permite definir un *objetivo* compartido por el *grupo* de actores que representa.

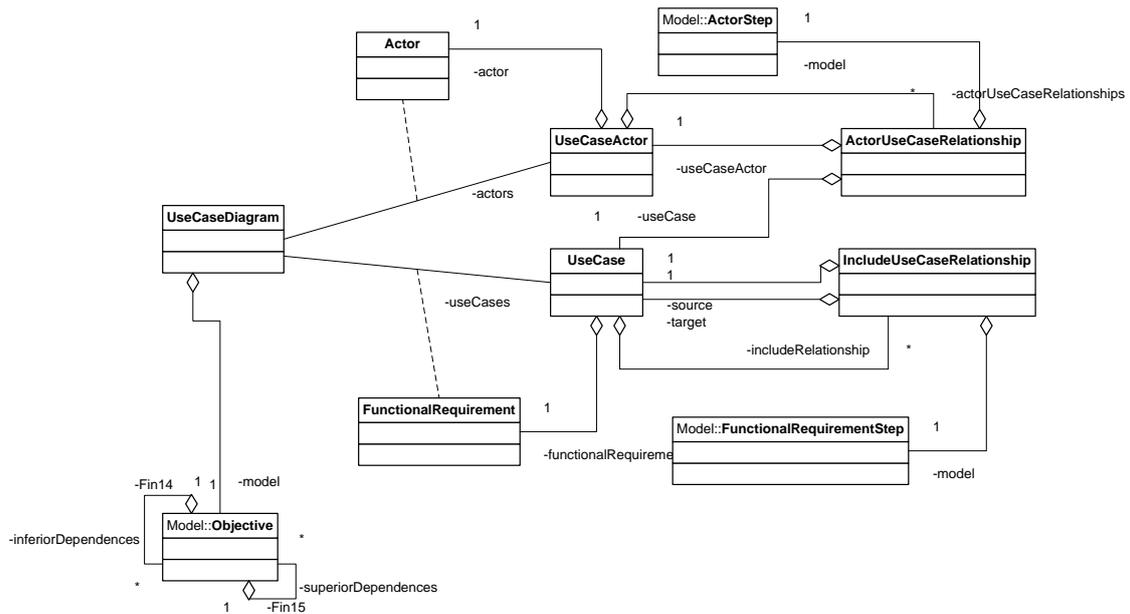


Figura 125. Modelo que representa un *diagrama de casos de uso* y su estructura

Para derivar el *diagrama de casos de uso* se transforma un *ActorStep* en una relación *ActorUseCaseRelationship* que une un *UseCaseActor* (representación de un *Actor* en este diagrama) con un *UseCase* (representación de un *FunctionalRequirement* en este tipo de diagrama). De la misma forma se deriva una relación *IncludeUseCaseRelationship* de un *FunctionalRequirementStep* que relaciona dos *UseCases* (*FunctionalRequirements*).

Como se ha dicho anteriormente, un diagrama de casos de uso (*UseCaseDiagram*) representa un *Objetivo* del modelo.

7.4.2.2 Documento de Requisitos del Sistema

El modelo del *documento de requisitos del sistema* plantea una solución al problema de la generación de documentos basado en un modelo y un dibujador. En este caso las clases definidas dentro del paquete *document* permiten la creación de un modelo abstracto de documento el cual describe la estructura del mismo. Esta estructura luego es dibujada mediante el dibujador (paquete *HTML* en este caso).

El diagrama de clases de la Figura 126 muestra la estructura abstracta del documento.

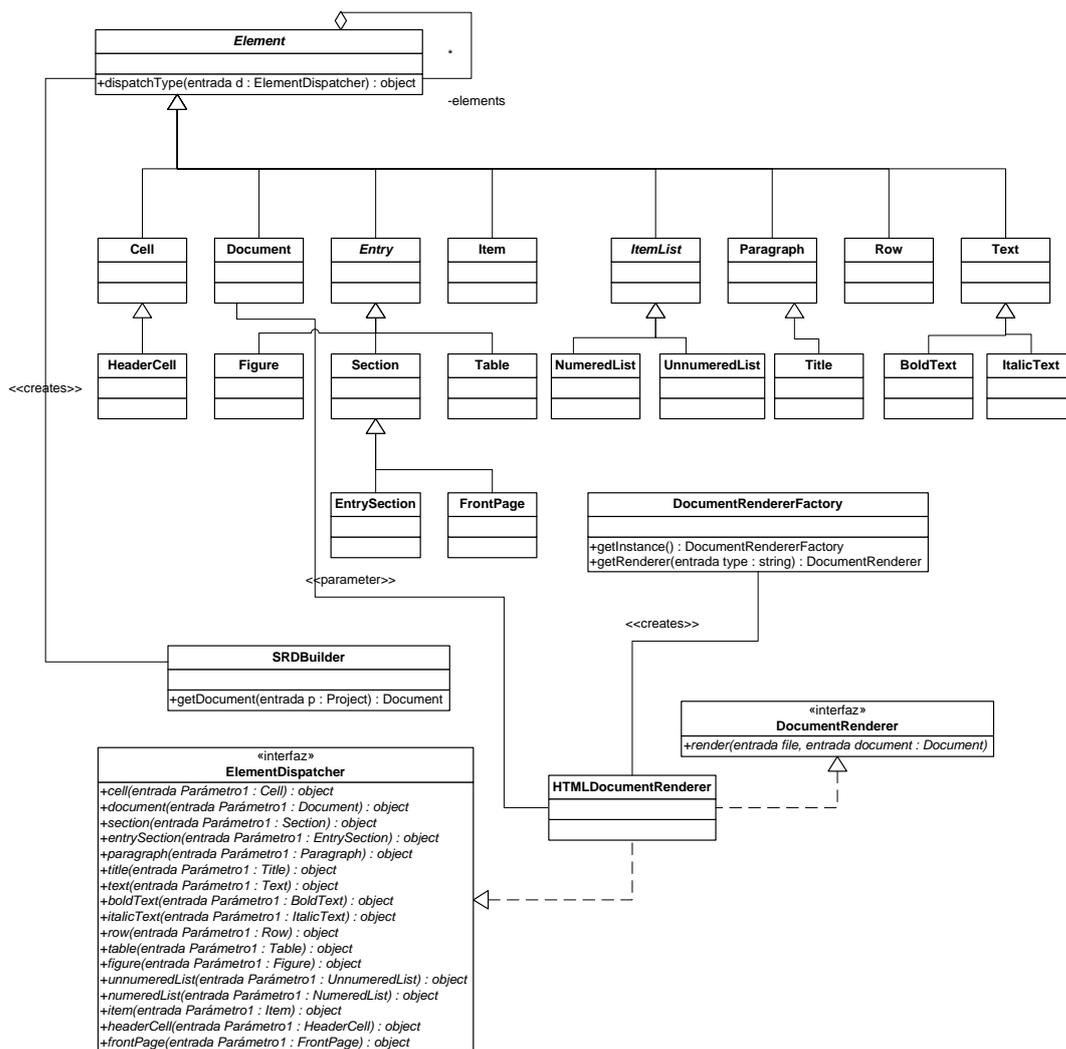


Figura 126. Estructura abstracta del documento de requisitos del sistema

Cada elemento del documento abstracto está representado por una subclase de *Element*. Para construir un documento se utiliza la clase *SDRBuilder* que a partir de un proyecto *Projecto* construye un documento (*Document*).

Una vez construido el documento, se procede a la generación del archivo, para ello se utiliza un dibujador (*DocumentRenderer*) que permite dibujar el contenido de un documento sobre un archivo.

HTMLDocumentRenderer es la implementación específica para *HTML* implementa la interfaz *ElementDispatcher* mediante la cual identifica los elementos que deben ser dibujados de acuerdo a la implementación del dibujador seleccionado.

La selección del dibujador se realiza mediante la clase *DocumentRendererFactory*, este case crea a los diferentes dibujadores y los configura, permitiendo su elección por parte del usuario.

7.4.3 Análisis

Para esta etapa se muestran los diagramas relacionados con los *roles* y las *tareas*. Los *roles* están modelados de acuerdo con el modelo de la Figura 127. La estructura de un rol (Role) relaciona un actor (Actor) con su rol.

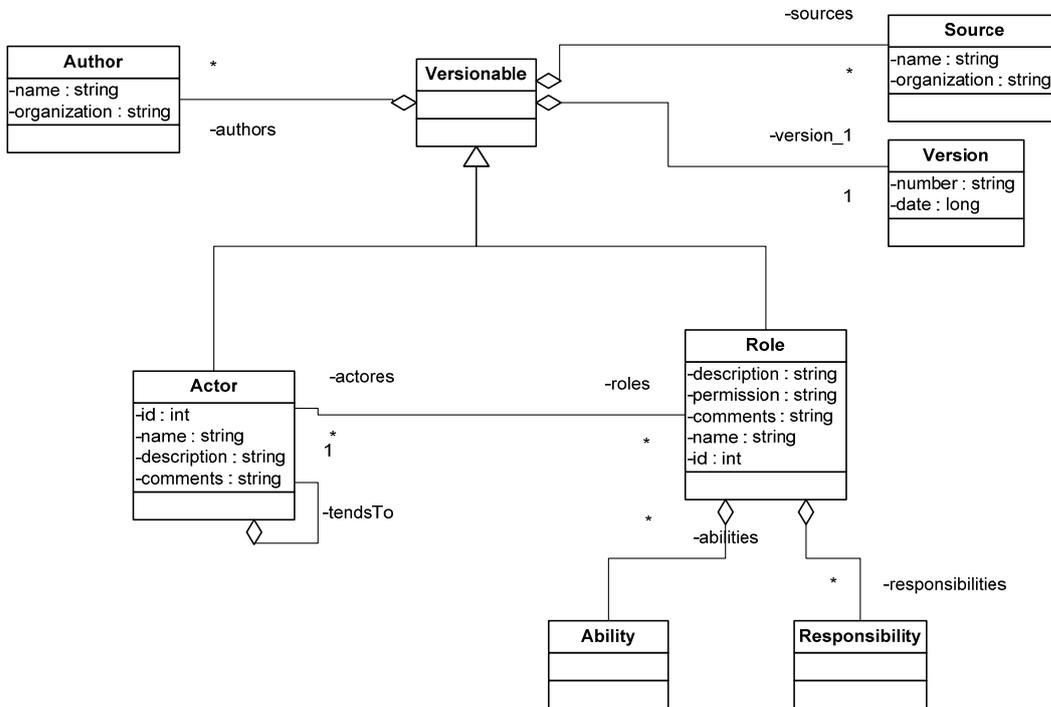


Figura 127. Modelo que representa los roles

7.4.4 La Vista

En esta sección describiremos la relación que existe entre las clases del modelo del dominio y la interfaz gráfica.

Como se puede observar, la interfaz se divide en dos partes principales. La parte izquierda muestra un *árbol* con la estructura de las diferentes etapas de la metodología que permite la navegación rápida y organizada a través de la información del modelo. Por otro lado, la parte derecha muestra en detalle *el nodo seleccionado en el árbol*. De esta forma la interfaz representa una vista previa y de detalle al mismo tiempo que permite una utilización ágil y comprensiva del sistema. El diagrama de la Figura 128 muestra los componentes principales de la interfaz gráfica.

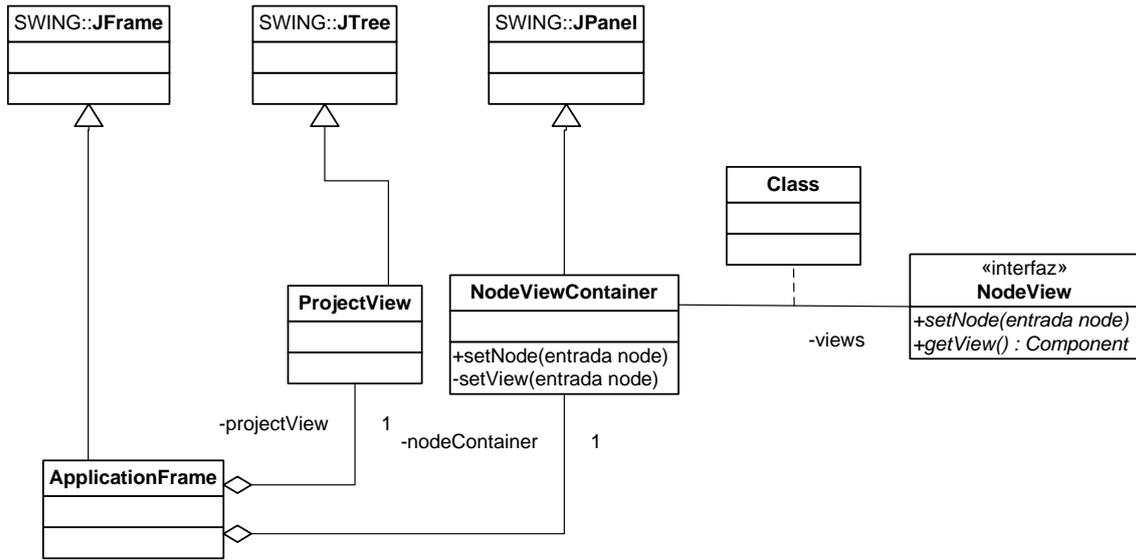


Figura 128. Modelo de los componentes principales de la interfaz gráfica

Cada nodo del árbol (*DefaultMutableTreeNode*) tiene asociado un objeto de usuario (*getUserObject*). Cuando se selecciona un nodo se notifica a la clase *NodeViewContainer* y se pasa el nodo como parámetro. El objeto que contiene este nodo define la vista que se mostrará. La vista se calcula por medio de la clase del objeto mencionado y la asociación se realiza mediante la colección *views* que tiene registradas las vistas para cada clase del modelo.

En general el sistema funciona bastante bien, sin embargo existe un conjunto de situaciones en las que no es tan trivial realizar dicha asociación. El caso típico se da cuando la misma clase define diferentes vistas. El caso más común es el de las colecciones.

Para ello se utilizan objetos al estilo "*adaptadores de aspectos*" que son clases que adaptan un aspecto del modelo al tipo necesario para corresponderse con la vista. Estos artefactos se sitúan usualmente en el paquete *App*. El diagrama de la Figura 129 muestra ejemplos del empleo del artefacto.

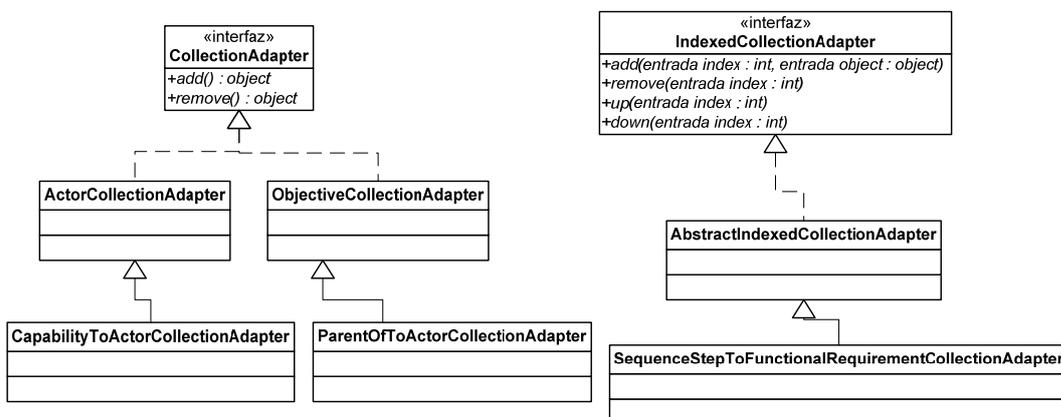


Figura 129. Ejemplos del empleo del artefacto

La principal diferencia entre *CollectionAdapter* e *IndexedCollectionAdapter* yace en la capacidad de la segunda de mantener el orden de los objetos que contiene.

7.4.4.1 Diagramas

La vista de los diagramas se basa en cuatro entidades básicas: *EntityView*, *EntityModel*, *DiagramView*, *RelationshipView*. El diagrama de clases se muestra en la Figura 130.

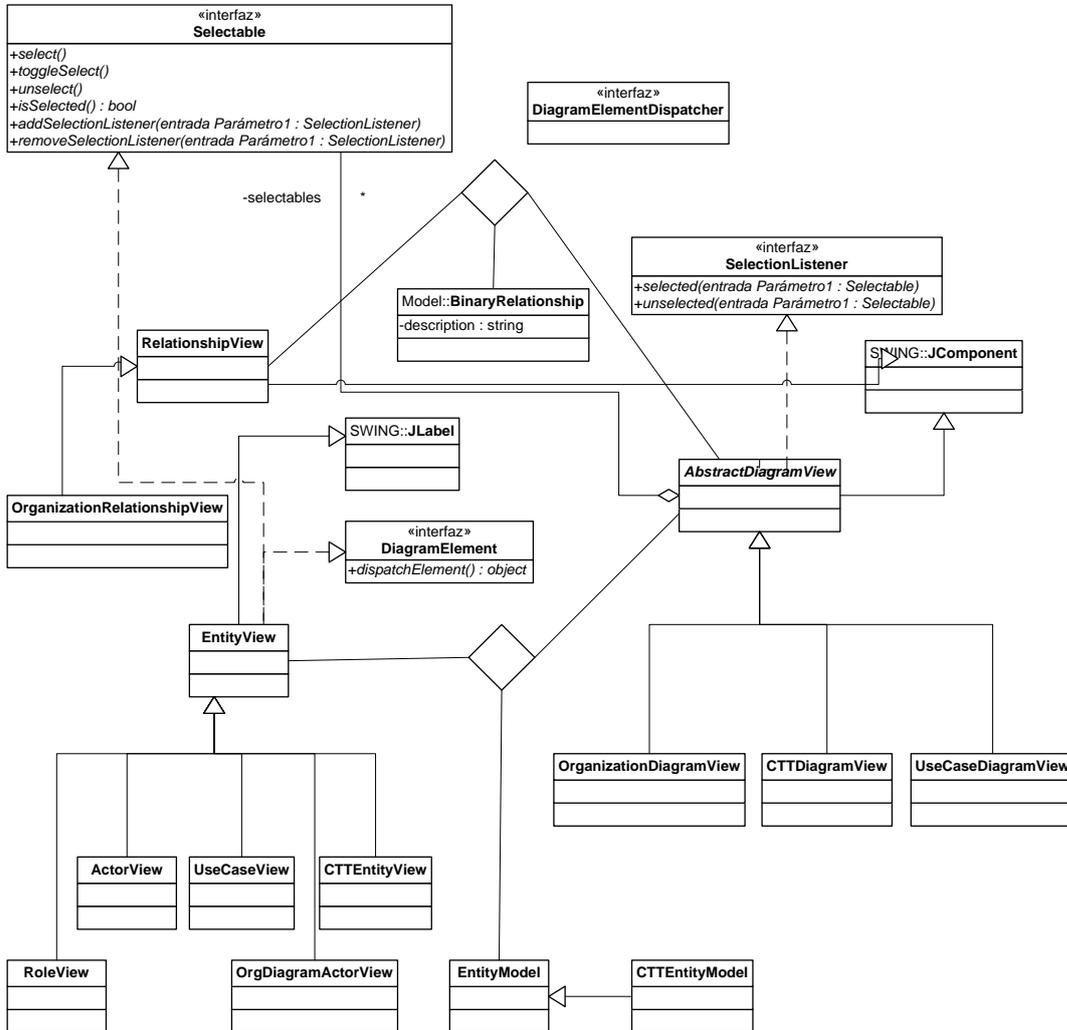


Figura 130. Representación para los diagramas de la aplicación

EntityView es una clase que agrupa la funcionalidad de cualquier entidad del diagrama (por ej. Ser seleccionado, arrastrar, etc.) *RelationshipView* es una clase que dibuja una relación binaria entre 2 entidades. Esta relación no tiene posición ya que es calculada de las entidades origen y destino. Por cada tipo de entidad y relación existe una subclase de *EntityView* y *RelationshipView* que la representa, respectivamente. La clase *AbstractDiagramView* abarca la funcionalidad relacionada con la selección de los objetos y la correspondencia entre los objetos del modelo y la vista (relaciones ternarias). El modelo de las relaciones es el mismo que el modelo de dominio, sin embargo para el caso de las entidades es distinto ya que las mismas deben "guardar" información relacionada con la disposición gráfica de la pieza en el diagrama, por lo que la misma es almacenada en un *Wrapper* a la clase de dominio. La clase encargada de llevar a cabo esto es la clase *EntityModel*. De esta forma se hacen persistentes datos como la posición de los componentes en la pizarra.

7.4.5 Conclusiones

El modelo del sistema se basa en una *arquitectura MVC*, aunque *View* y *Controller* estén estrechamente relacionadas por la forma de trabajo de *SWING*.

El modelo es lo suficientemente genérico para soportar diferentes modos de visualización y diferentes formas de generación de informes.

También permite el almacenamiento y recuperación de archivos en formato *XML*

Finalmente, el sistema es capaz de mantener la coherencia entre las distintas etapas mediante la utilización de correspondencias de modelo (*Requisitos-Tareas*).

7.5 Funcionamiento de TOUCHE CASE Tool

TOUCHE CASE Tool –Task-Oriented and User-Centred Process Model for Developing Interfaces

for Human-Computer-Human Environments Computer-Aided Software Engineering Tool– es la herramienta CASE presentada en la tesis como aplicación que pone en práctica el modelo de proceso y la metodología propuestos en la tesis. Esta herramienta asiste al usuario en el desarrollo de interfaces de usuario centrado en el usuario y dirigido por tareas para sistemas CSCW.

Por su naturaleza CASE, *TOUCHE CASE Tool*, facilita la automatización de algunos procesos en el ciclo de vida del software, relaciona y muestra la información desde diferentes prespectivas facilitando al desarrollador esta ardua tarea y mantiene la coherencia del desarrollo al garantizar la trazabilidad inter- e intraetapa entre los diferentes modelos.

7.5.1 Creación de un Proyecto

La herramienta presenta las cuatro etapas del proceso de desarrollo de aplicaciones tradicional en Ingeniería del Software: Elicitación de Requisitos, Análisis, Diseño e Implementación. El modelo de proceso –TOUCHE– es además un modelo iterativo y considera el prototipado y la evaluación por parte de los usuarios para llevar a cabo un proceso más completo y de mayor calidad. Esto es algo que la herramienta facilita considerablemente.

En la creación de cualquier proyecto, estas etapas están representadas en un árbol a la izquierda de la pantalla tal y como se muestra en la Figura 131.

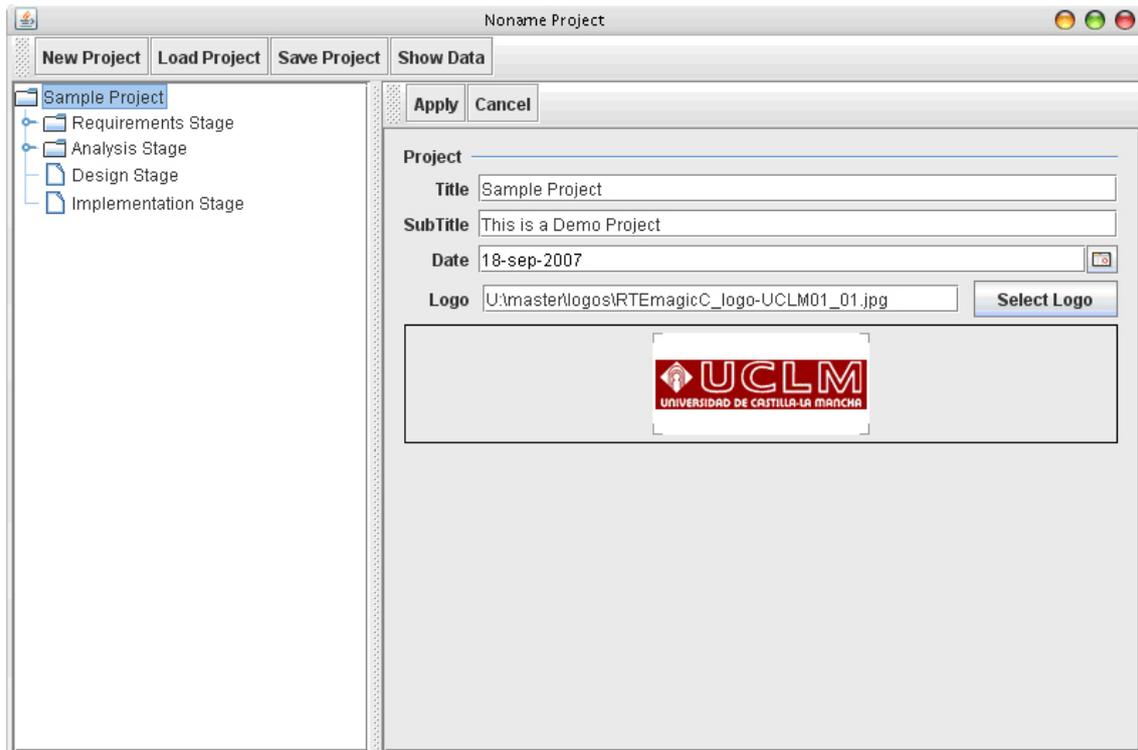


Figura 131. Captura de la creación de un proyecto donde se pueden apreciar, en el menú de la izquierda, las cuatro etapas del modelo de proceso

El nodo principal muestra los datos relacionados con el conjunto del proyecto. Estos datos son el *título*, el *subtítulo*, la *fecha de creación* del documento y se puede asignar un *logotipo* al trabajo. Esta imagen se muestra en la portada del *Documento de Requisitos del Sistema (DRS)* que genera la herramienta.

7.5.2 Etapa de Elicitación de Requisitos

Los datos y plantillas relacionadas con esta etapa se encuentran bajo el primer nodo del proyecto, como se muestra en la Figura 132. Seleccionando el nodo de la etapa de requisitos se pueden insertar los datos relacionados con la *introducción* y la *descripción* del sistema a desarrollar. Bajo este nodo también se introducen los datos relacionados a la información de conocimiento del dominio en forma de *recursos*, como se muestra en la Figura 133.

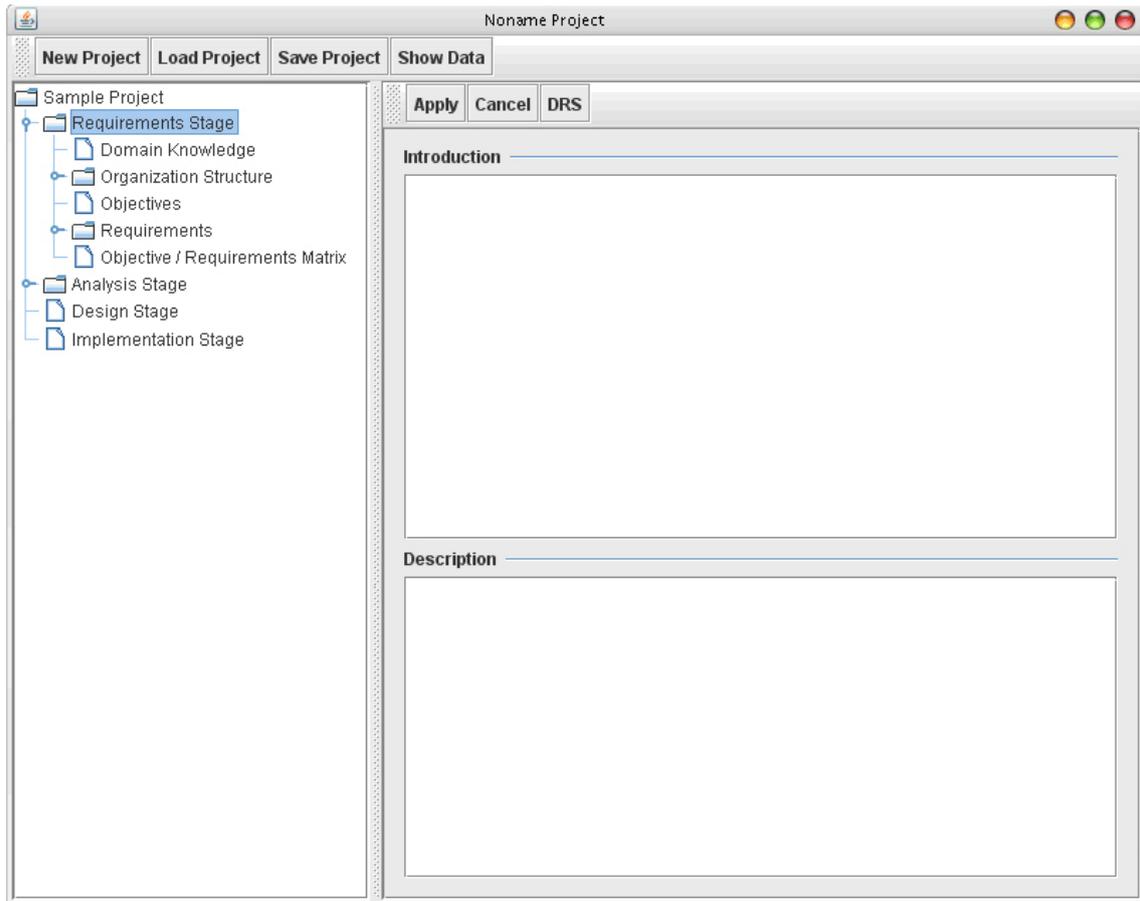


Figura 132. Captura de la pantalla principal en la *Elicitación de Requisitos*

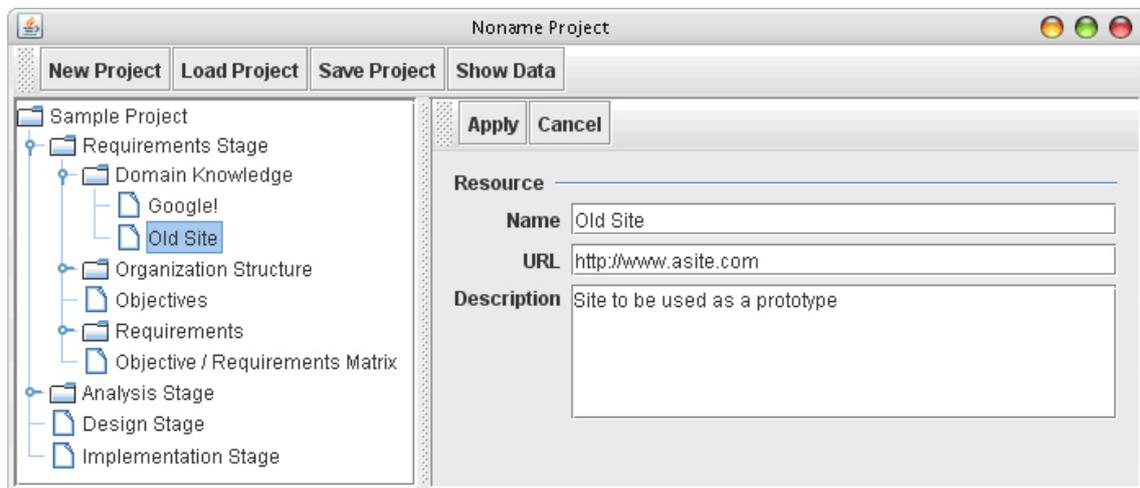


Figura 133. Insercción de recursos: información relacionada con el dominio del problema

El nodo relacionado con la *estructura organizativa* genera las plantillas relacionadas con los *actores* del sistema y está ordenado como se muestra en el panel del árbol de la Figura 134. Cada uno de los *actores* presenta una plantilla de resumen que permite introducir datos y ver el contenido de los nodos a modo de vista previa del mismo. El árbol tiene un nodo que agrupa cada tipo de *actor* (*Actor*, *Grupo*, *Individuo*, *Usuario* o *Agente*), además de las *fuentes* y *autores* de la elicitación de requisitos.

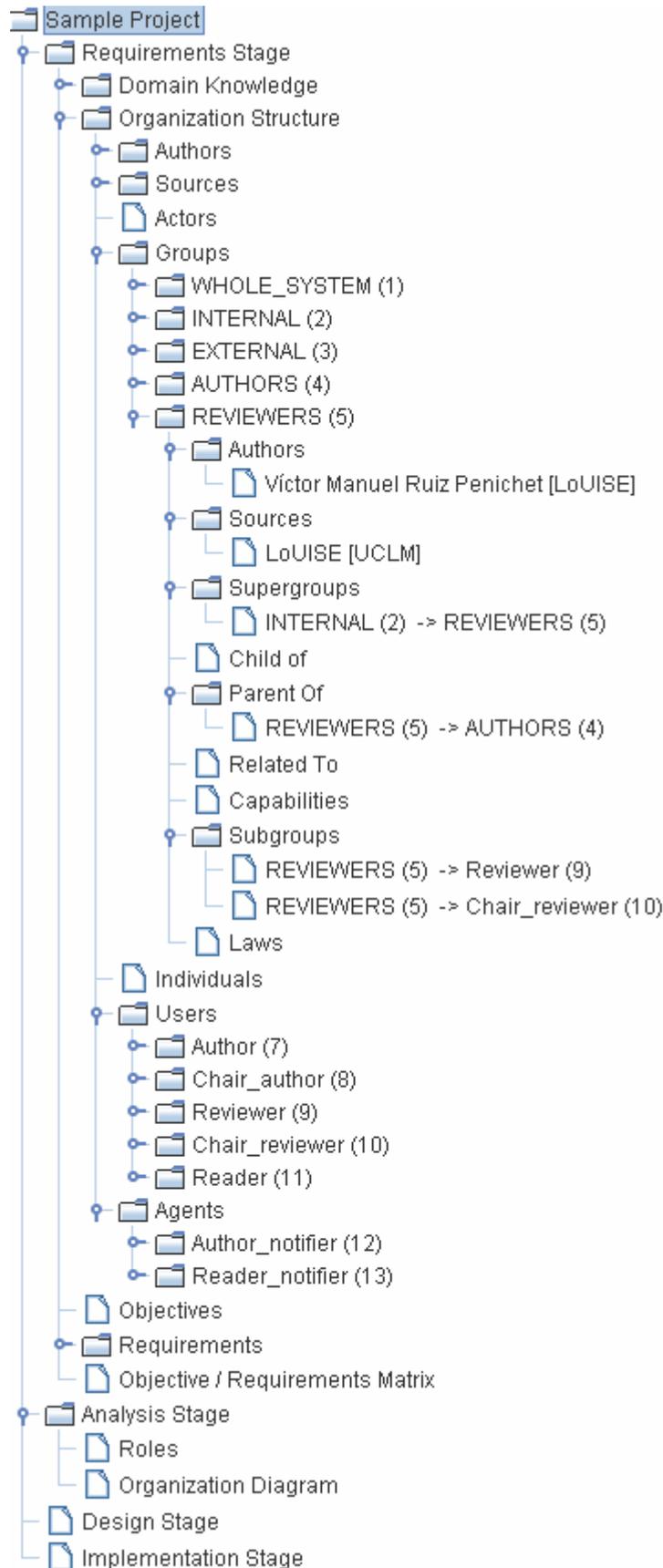


Figura 134. Despliegue en el árbol de la información relativa a la *estructura organizativa* del sistema

Un ejemplo de plantilla se puede ver en la Figura 135. Dicha plantilla incluye información como la descripción del actor (*Actor, Grupo, Individuo, Usuario o Agente*), los *autores*, las *fuentes*, los *súper-grupos* de los que es parte, las *relaciones de jerarquía* del grupo (padre o hijo), los *actores* con los que tiene relación, las *capacidades* (en este caso, *habilidades y responsabilidades asignadas*), los *comentarios*. En este caso es un *grupo*, por lo tanto se especifican, el *objetivo del grupo*, los *grupos* que son parte de él y las *leyes* que lo gobiernan.

Group

Id 5

Name REVIEWERS

Version

Number 20070918170138

Date 18-sep-2007

Description

Description of the Reviewer Group

Authors

Name	Organization
Victor Manuel Ruiz Penichet	LoUISE

Sources

Name	Organization
LoUISE	UCLM

Supergroups

Id	Name	Description
2	INTERNAL	Enter description

Child of

Id	Name	Description
----	------	-------------

Parent of

Id	Name	Description
4	AUTHORS	Enter description

Related to

Id	Name	Description
----	------	-------------

Capabilities

Id	Name	Description
1	Ability	Description of the Abil...
1	Responsibility	Description of Respo...

Comments

Comments of the Group

Shared Objective

Undefined

Laws

Id	Description
1	First Law

Subgroups

Id	Name	Description
9	Reviewer	Enter description
10	Chair_reviewer	Enter description

Figura 135. Ejemplo de *plantilla* para un *actor* de tipo *grupo*. Nota: La figura está partida en dos para mejorar su visualización completa: la parte de la derecha es continuación de la de la izquierda

De la misma manera que en el caso de la *estructura organizativa*, el árbol de la izquierda muestra la organización de los *objetivos* del sistema (Figura 136). Un objetivo tiene asociados *autores, fuentes, objetivos y requisitos asociados, las percepciones, los participantes* que llevan a cabo el objetivo, los *objetivos* de los que depende y los *objetivos* que tiene dependencias en él.

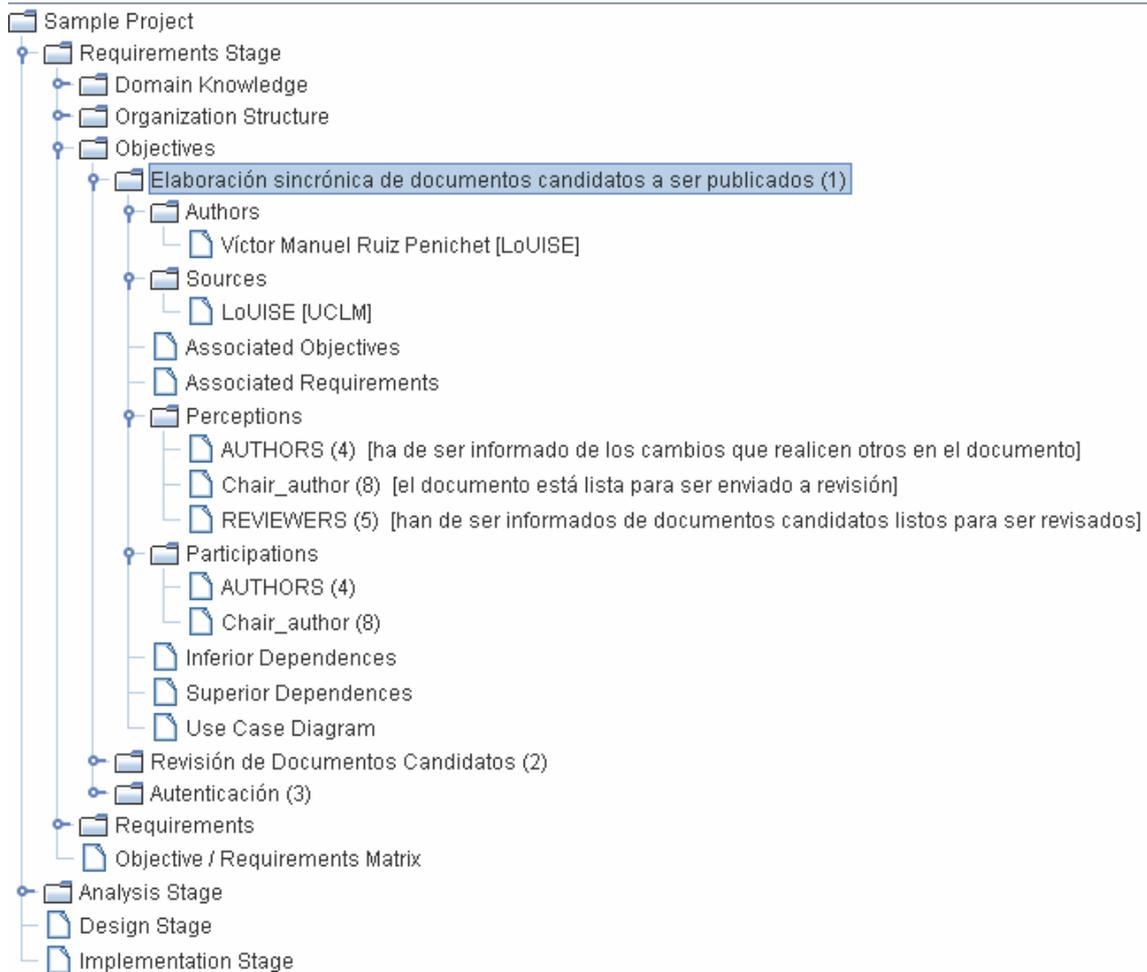


Figura 136. Captura donde se muestra desplegada la información relativa a los objetivos del sistema. Concretamente está señalado el primero de ellos.

Posicionándose en un nodo de un objetivo se puede ver un resumen de la información de la plantilla que tiene asociado (Figura 137).

Objective			
Id 1			
Name: [Elaboración sincrónica de documentos candidatos a ser publicados]			
Version: []			
Number: [20070918174559]			
Date: [18 sep 2007]			
Authors			
Name		Organization	
Victor Manuel Ruiz Penichet		LOUISE	
Sources			
Name		Organization	
LOUISE		UCLM	
Associated Objectives			
Id		Name	
2		Revisión de Documentos Candidatos	
Associated Requirements			
Id		Type	Name
1		FR	Acceso al sistema
2		FR	Autenticación
3		FR	Crear Sesión
4		FR	Iniciar Sesión
5		FR	Unirse a Sesión
Importance			
Muy Importante			
Urgency			
Alta			
State			
Especificado. Por implementar			
Stability			
Puede sufrir cambios, pero actualmente estable			
Perception			
Type	Actor	Description	
AUTHORS (4)		AUTHORS (4) [ha de ser informado de los cambios que realicen otros en el documento]	
Chair_author (8)		Chair_author (8) [el documento está lista para ser enviado a revisor]	
REVIEWERS (5)		REVIEWERS (5) [han de ser informados de documentos candidatos listos para ser revisados]	
Comments			
Ninguno			
Description			
El sistema deberá dar soporte a la elaboración sincrónica de documentos candidatos a ser publicados. Es decir, los documentos, antes de ser públicos tras su proceso de revisión, pueden ser elaborados por diferentes usuarios a la vez a través del sistema.			
Inferior Dependences			
Id		Name	
Superior Dependences			
Id		Name	
3		Autenticación	
CSCW Description			
Por la naturaleza colaborativa del objetivos se debería tener en cuenta lo siguiente: - Los usuarios interactúan en tiempo real sobre un mismo documento - Los usuarios se comunican por medio de Chat			
Environmental Information			
El entorno de ejecución del sistema será: - Los usuarios podrán interactuar desde cualquier máquina con un navegador y acceso a internet - No hay otra necesidad específica adicional			
CSCW Features			
Collaboration	Cooperation	Coordination	Communication
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Space / Time Matrix			
Space		Same	Different
Time		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Level of Requirements			
La exigencia más importante es la ejecución en tiempo real. Es necesario que funcione con agilidad para no entorpecer el desarrollo del documento, sin embargo puesto que trabajan en zonas diferentes, no es absolutamente vital. Se podría permitir un retraso ocasional, no así en la parte de comunicació			
Participants			
Author		Description	
AUTHORS (4)		Elaboración sincrónica del documento	
Chair_author (8)		Envío a revisión cuando el documento candidato está listo	

Figura 137. Captura de la información relativa a un objetivo, concretamente el primero de ellos

Una característica importante de la herramienta es la posibilidad de generar un *diagrama de casos de uso* del *objetivo*. Este diagrama utiliza los datos especificados en los *requisitos funcionales* que se asocian con la *secuencia normal de ejecución* del mismo y la asociación de los *requisitos funcionales* asociados con el *objetivo* (Figura 138).

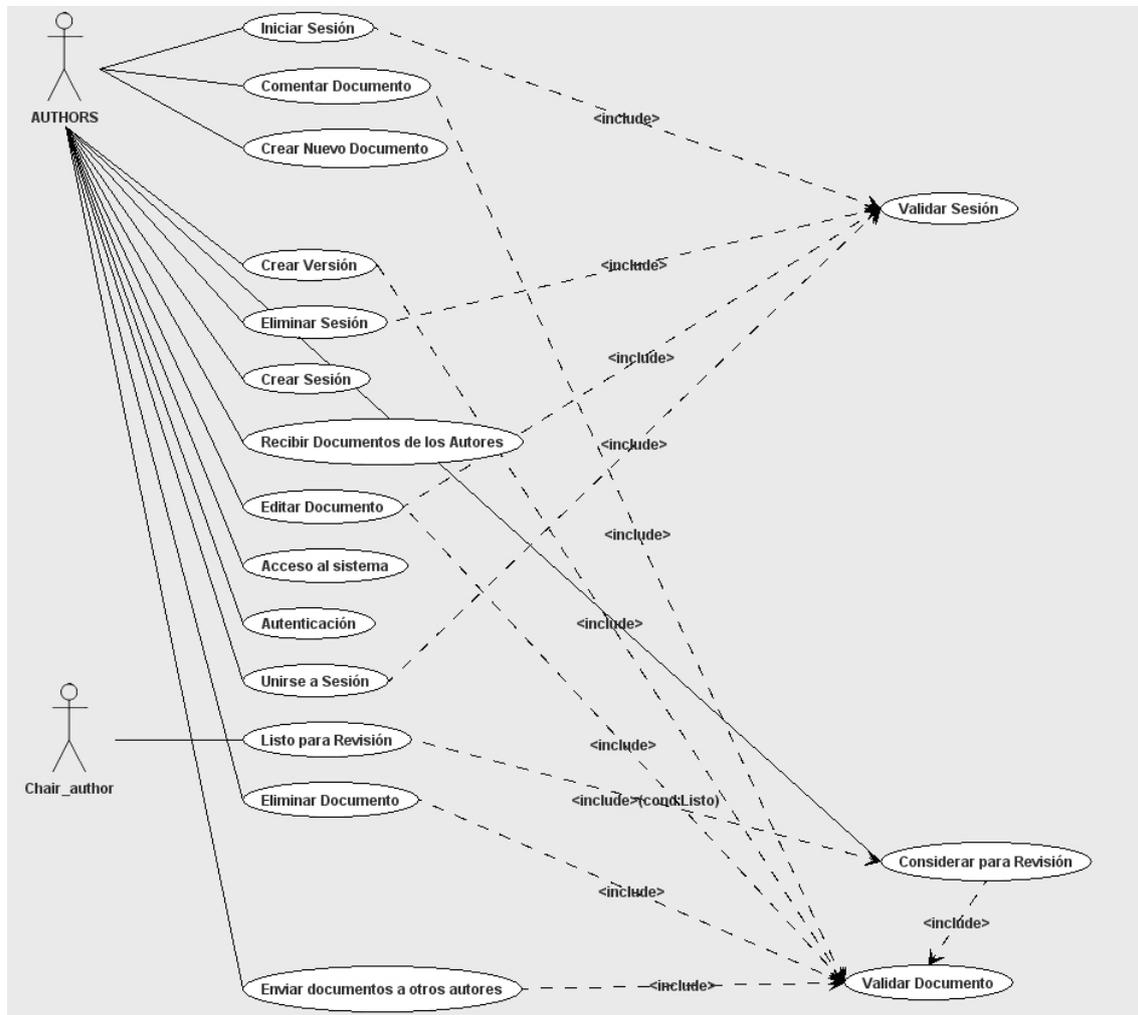


Figura 138. Diagrama de casos de uso para el objetivo especificado generado de forma automática a partir de la información facilitada en los *requisitos funcionales* que se asocian con la *secuencia normal de ejecución* del mismo y la asociación de los *requisitos funcionales* asociados con el *objetivo*

De la misma forma que en los casos anteriores la especificación de los *requisitos* se puede ver en el árbol de la izquierda (Figura 139). Este árbol agrupa los *requisitos* en función del tipo de los mismos, es decir, en *requisitos funcionales*, *no funcionales* y *de información*. Cada nodo de grupo de *requisitos* presenta un resumen de los *requisitos* que contiene. Los *requisitos* permiten la introducción de información según las *plantillas* definidas anteriormente. Tomaremos como ejemplo un *requisito funcional* "Editar Documento".

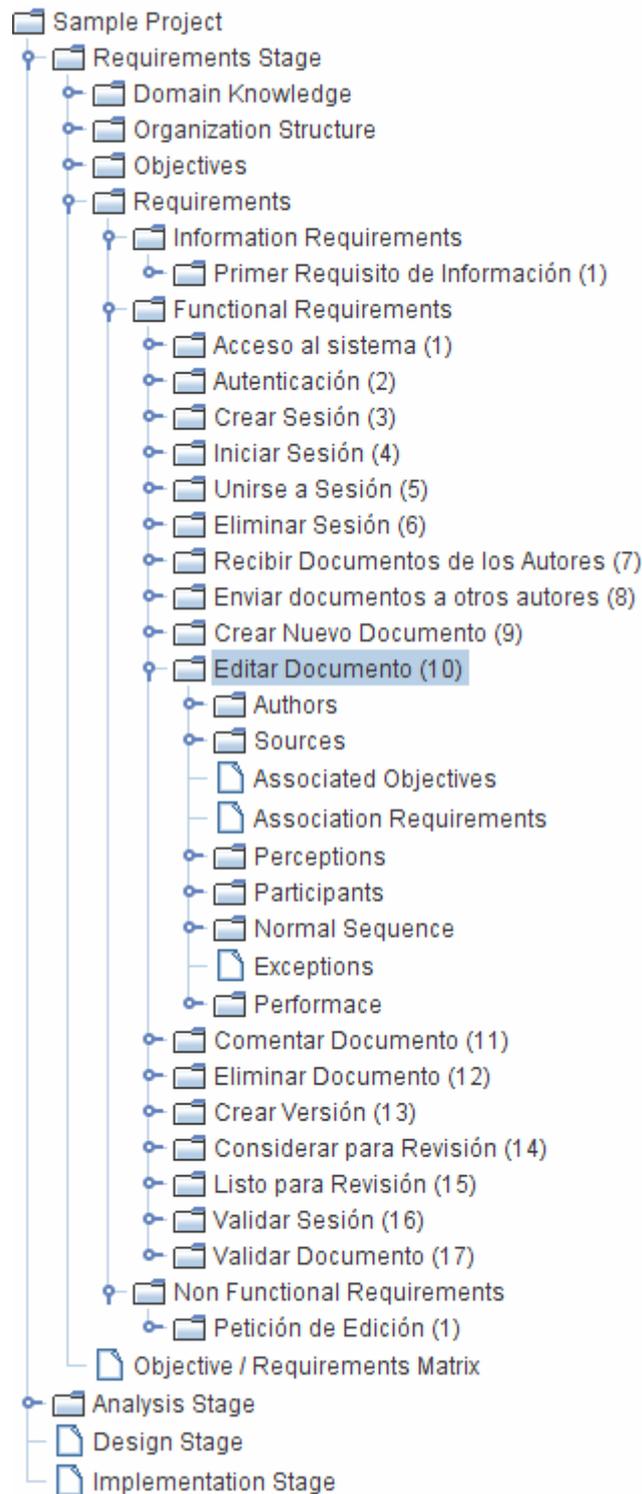


Figura 139. Árbol con los datos específicos para la *elicitación de requisitos*

Cada *requisito funcional* tiene asociado *autores*, *fuentes*, *objetivos asociados* (que se sincronizan automáticamente si se agrega a un *objetivo* un *requisito asociado*), se añaden también las *percepciones*, los *participantes*, una *secuencia de acciones normales* y una de *excepción* y finalmente los *requisitos de desempeño o rendimiento*. Un ejemplo del resumen de este tipo de requisito se ve en la Figura 140.

Functional Requirements			
Id 10			
Name <input type="text" value="Editar Documento"/>			
Version <input type="text"/>			
Number <input type="text" value="20070918181651"/>			
Date <input type="text" value="18-sep-2007"/> <input type="button" value="CA"/>			
Authors			
<input type="text" value="Name"/>	<input type="text" value="Organization"/>		
<input type="text" value="Víctor Manuel Ruiz Penichet"/>	<input type="text" value="LoUISE"/>		
Sources			
<input type="text" value="Name"/>	<input type="text" value="Organization"/>		
<input type="text" value="LoUISE"/>	<input type="text" value="UCLM"/>		
Associated Objectives			
<input type="text" value="Id"/>	<input type="text" value="Name"/>		
<input type="text" value="1"/>	<input type="text" value="Elaboración sincrónica de docum..."/>		
Associated Requirements			
<input type="text" value="Id"/>	<input type="text" value="Type"/>	<input type="text" value="Name"/>	
Importance			
<input type="text" value="Muy importante"/>			
Urgency			
<input type="text" value="Alta"/>			
State			
<input type="text" value="Especificado. Por implementar"/>			
Stability			
<input type="text" value="Puede sufrir cambios, pero actualmente estable"/>			
Perception			
<input type="text" value="Type"/>	<input type="text" value="Actor"/>	<input type="text" value="Description"/>	
<input type="text" value="AUTHORS (4)"/>	<input type="text" value="AUTHORS (4)"/>	<input type="text" value="[un aut..."/>	
<input type="text" value="AUTHORS (4)"/>	<input type="text" value="AUTHORS (4)"/>	<input type="text" value="[un act..."/>	
Comments			
<input type="text" value="Ninguno"/>			
Description			
<input type="text" value="El sistema deberá comportarse tal como se describe en el siguiente caso"/>			
<input type="text" value="o al que se hace referencia existe, el usuario es un usuario registrado en"/>			
Precondition			
<input type="text"/>			
Normal Sequence			
<input type="text" value="Step"/>	<input type="text" value="Description"/>		
<input type="text" value="0"/>	<input type="text" value="<AUTHORS (4) >"/>		
<input type="text" value="1"/>	<input type="text" value="<Validar Sesión (16)>"/>		
<input type="text" value="2"/>	<input type="text" value="<Validar Documento (17)>"/>		
Postcondition			
<input type="text" value="Salvar o cancelar cambios"/>			
Exceptions			
<input type="text" value="Step"/>	<input type="text" value="Description"/>		
Performance			
<input type="text" value="Step"/>	<input type="text" value="Value"/>	<input type="text" value="Unit"/>	
<input type="text" value="1"/>	<input type="text" value="10"/>	<input type="text" value="milliseconds"/>	
<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="days"/>	
Expected Frequency			
<input type="text" value="30"/>	<input type="text" value="seconds"/>	<input type="text"/>	
CSCW Description			
<input type="text" value="Por la naturaleza colaborativa del requisito...."/>			
Environmental Information			
<input type="text" value="El entorno de ejecución será: ..."/>			
CSCW Features			
<input type="checkbox" value="Collaboration"/>	<input checked="" type="checkbox" value="Cooperation"/>	<input type="checkbox" value="Coordination"/>	<input type="checkbox" value="Communication"/>
Space / Time Matrix			
	<input type="text" value="Same"/>	<input type="text" value="Different"/>	
<input type="text" value="Space"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="text" value="Time"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Level of Requirements			
<input type="text" value="No demasiado alto. Un poco más en la sincrónica"/>			
Participants			
<input type="text" value="Author"/>	<input type="text" value="Description"/>		
<input type="text" value="AUTHORS (4)"/>	<input type="text"/>		

Figura 140. Captura de ejemplo de la descripción de un requisito funcional. Nota: La figura está partida en dos para mejorar su visualización completa: la parte de la derecha es continuación de la de la izquierda

El sistema también muestra la matriz de rastreabilidad en el último nodo de la sección de requisitos. La Figura 141 muestra esta tabla.

	OBJ-1	OBJ-2	OBJ-3
FR-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-6	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR-17	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NFR-1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
IR-1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 141. Matriz de rastreabilidad Objetivos-Requisitos

7.5.3 Análisis, Diseño e Implementación

La etapa de elicitación de requisitos se ha mostrado con mayor detalle que las siguientes puesto que el funcionamiento general de la aplicación es similar en todas las etapas. En este apartado se muestran las capturas más representativas correspondientes a las etapas de análisis y diseño del modelo de proceso. La etapa de implementación, a pesar de ser una traducción de la de diseño, es un trabajo futuro que debe contemplar también la creación de toolkits específicos para los sistemas CSCW.

La sección de *análisis* está compuesta básicamente por dos elementos principales, las *tareas* y los *roles*. En el árbol de la izquierda, como se puede ver en la Figura 142, se observan los datos organizados de la siguiente manera: *roles*, dónde cada *rol* muestra los *autores*, las *fuentes*, *responsabilidades*, *capacidades* y el *diagrama CTT* de la asociación de las *tareas* con los *roles*. La Figura 143 muestra el resumen de datos presentado por un *rol*.

La Figura 144 muestra el *diagrama de tareas en notación CTT* asociado a un *rol*, en el que todavía no se han introducido las relaciones entre *tareas*.

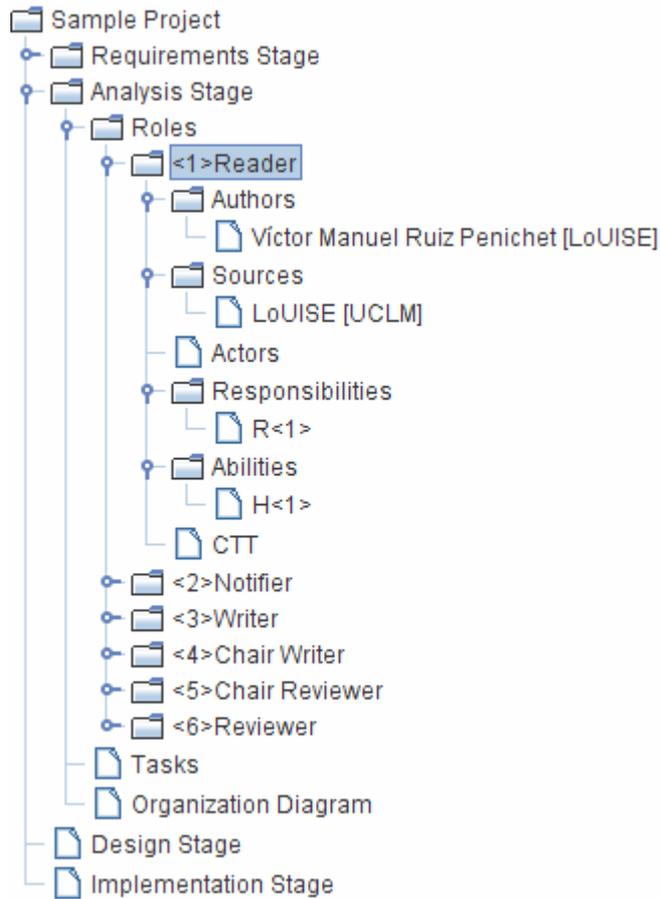


Figura 142. Captura de la etapa de *análisis* en el árbol. Desplegado el rol "Reader"

The screenshot displays a detailed view of a role named 'Reader'. The interface is split into two main sections. The left section contains the following fields and tables:

- Role**: Id 1
- Name**: Reader
- Version**: (empty)
- Number**: 20070919031036
- Date**: 19-sep-2007
- Authors**: A table with columns 'Name' and 'Organization'. It contains one entry: Víctor Manuel Ruiz Penichet (LoUISE).
- Sources**: A table with columns 'Name' and 'Organization'. It contains one entry: LoUISE (UCLM).
- Associated Actors**: A table with columns 'Id', 'Name', 'Tends To', and 'Because'. It is currently empty.

The right section contains the following fields and tables:

- Description**: A large empty text area.
- Responsibilities**: A table with columns 'Id' and 'Description'. It is currently empty.
- Abilities**: A table with columns 'Id' and 'Description'. It is currently empty.
- Permissions**: A large empty text area.
- Comments**: A large empty text area.

Figura 143. Captura con la información relativa a un rol. Nota: La figura está partida en dos para mejorar su visualización completa: la parte de la derecha es continuación de la de la izquierda

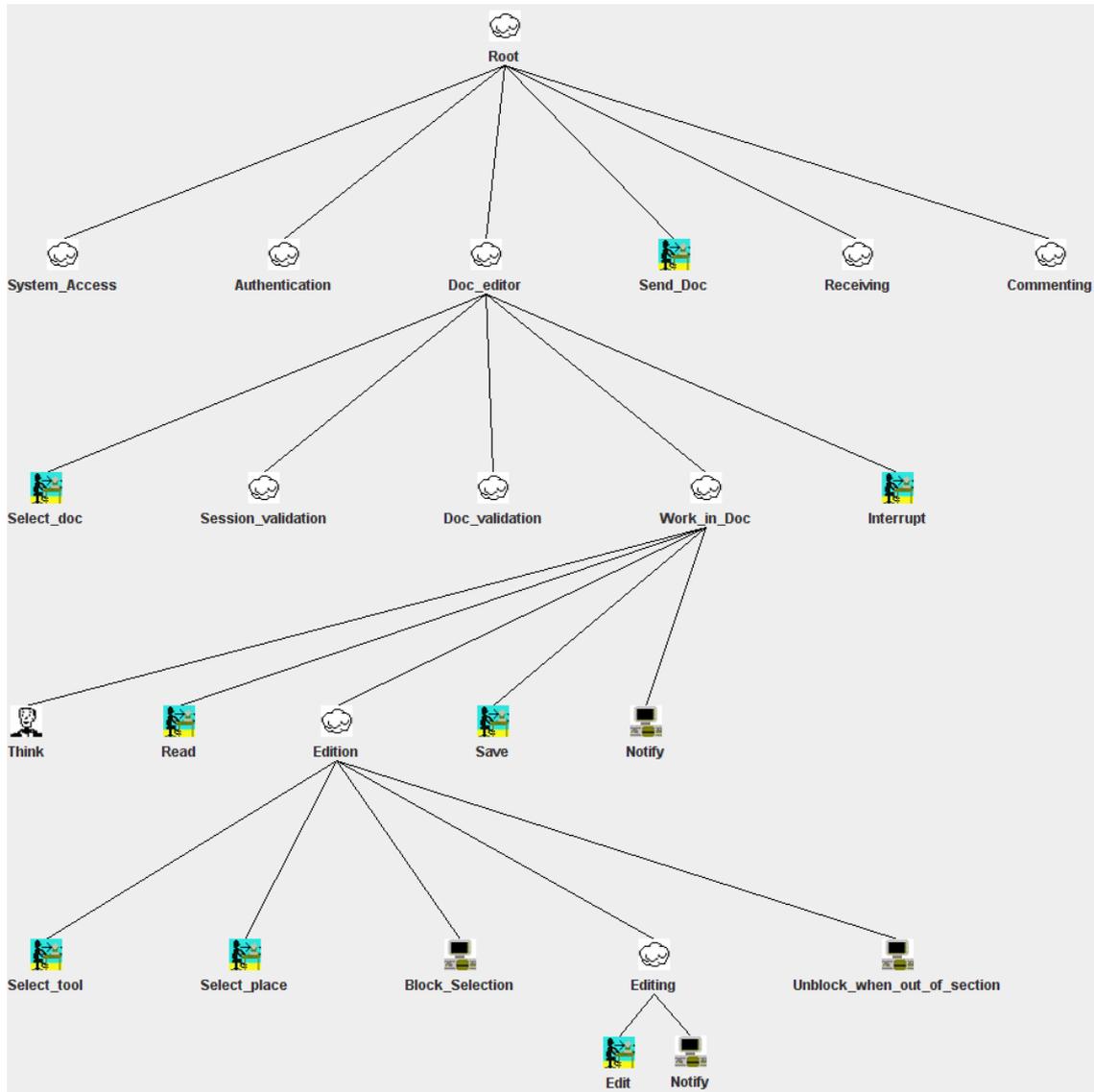


Figura 144. Diagrama de tareas de un rol en CTT a falta de introducir las relaciones entre las tareas

La Figura 145 y la Figura 146 muestran los diagramas relativos a la etapa de diseño realizados mediante TOUCHE CASE Tool: Diagrama de Interacción de Contenedores Abstractos (ACID) y Diagrama de Interfaces de Usuario Abstractas (AUID) respectivamente.

El ACID permite modelar la navegación entre los diferentes contenedores, a menudo ventanas, de la aplicación según el rol que se desempeñe y según qué tarea se lleve a cabo.

El AUID modela cómo sería cada uno de esos contenedores teniendo en cuenta, además de los aspectos relativos a HCI según los trabajos previos, aspectos típicos de las aplicaciones groupware que favorecen la percepción de los usuarios del grupo.

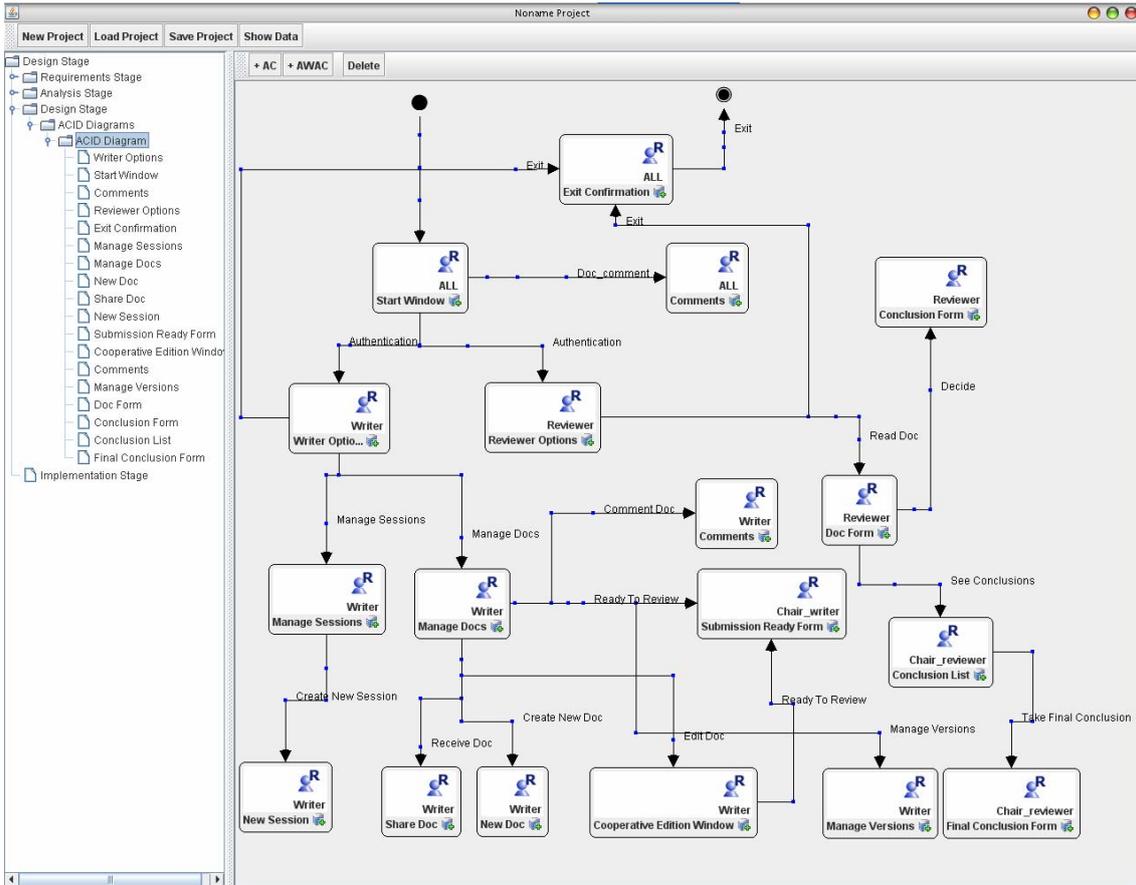


Figura 145. Diagrama de Interacción de Contenedores Abstractos (ACID) generado desde TOUCHE CASE Tool

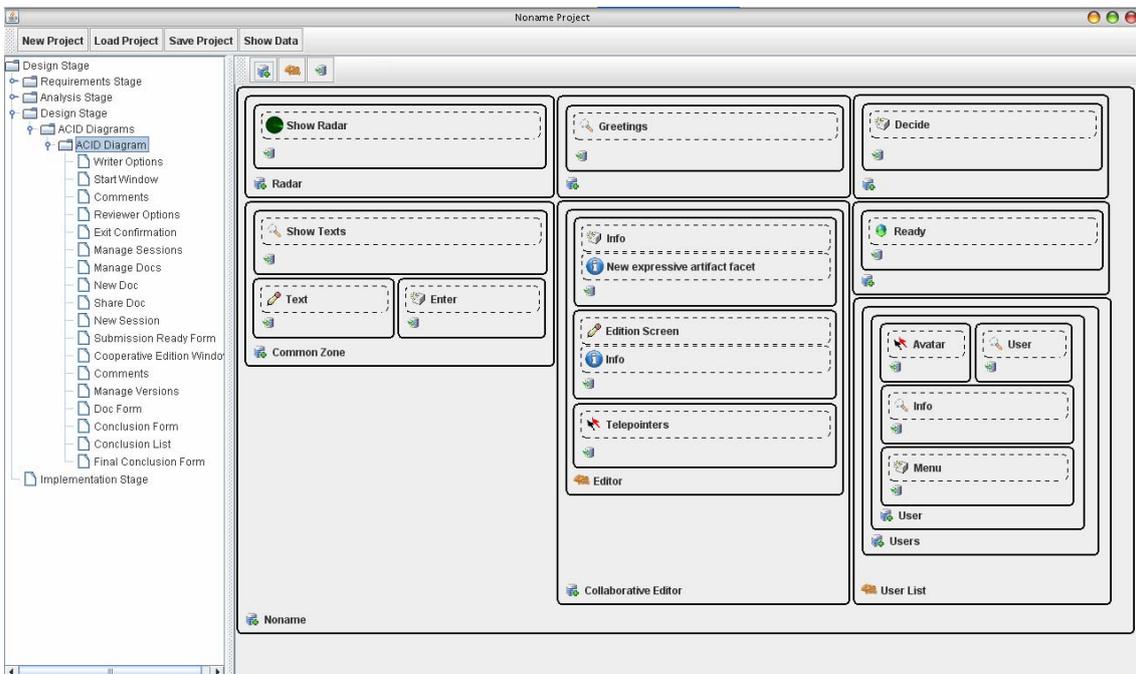


Figura 146. Diagrama de Interfaces de Usuario Abstractas (AUID) generado desde TOUCHE CASE Tool

CAPÍTULO 8.

CONCLUSIONES, APORTACIONES Y TRABAJOS FUTUROS

Este capítulo muestra las conclusiones a las que se llega después del trabajo abordado en la tesis, así como las aportaciones científicas en forma de publicaciones realizadas en este marco y los trabajos futuros que pueden realizarse a partir de la misma.

8.1 Conclusiones

El desarrollo del software ha sufrido importantes cambios en los últimos años. Además de los cambios tecnológicos, las aplicaciones actuales están experimentando una fuerte tendencia a ser empleadas por grupos de usuarios a través de Internet. Es decir, las necesidades de los usuarios finales han provocado un giro en el diseño de aplicaciones de manera que no son herramientas para resolver problemas de personas individuales, sino que son sistemas en los que hay una carga muy importante de comunicación, colaboración, cooperación y coordinación de diferentes usuarios. Este es el motivo que hace que las aplicaciones groupware sean tan importantes en la actualidad.

Este proceso se puede explicar del siguiente modo: en los últimos años se ha pasado de una programación centrada en la herramienta, a un desarrollo centrado en el usuario y ahora a un desarrollo centrado en el grupo. Este es un paso más que surge precisamente de los círculos del área del HCI: es un diseño de sistemas centrado en el usuario y su entorno.

Los sistemas CSCW tienen un grado de complejidad adicional pues se debe tener muy en cuenta, además de una correcta solución de la problemática y la interacción persona ordenador, al usuario como miembro del grupo, y por lo tanto toda una nueva forma de actuar frente a los sistemas. Teniendo en cuenta esta idea, las herramientas desarrolladas cumplirán con las expectativas de los usuarios y facilitarán su trabajo en grupo, la colaboración entre ellos, su comunicación y coordinación independientemente el lugar en el que se encuentren. Por tanto, el desarrollo de sistemas con estas características requiere diversas técnicas y modelos para abordarlos.

En este trabajo, en primer lugar se ha realizado un estudio de los sistemas CSCW, desde el uso de algunas aplicaciones hasta la propia terminología. Después se ha presentado un modelo de proceso y una metodología que asisten el desarrollo de interfaces de usuario para aplicaciones groupware desde el estudio de los requisitos particulares del sistema. Para ello se aplican conocimientos existentes, especialmente aquellos que provienen del campo de la interacción persona-ordenador (HCI), al mundo del trabajo cooperativo asistido por computador (CSCW). Es decir, se trata de cambiar el punto de vista del modelado y desarrollo de sistemas, al pasar de considerar únicamente la interacción de un solo usuario con una herramienta, a la preocupación –modelado- por el comportamiento social de un conjunto de individuos que interactúan, trabajan, se comunican y colaboran y cooperan unos con otros a través de las redes de ordenadores.

El trabajo ha producido una serie de aportaciones que se describen en el apartado 8.2 y varias publicaciones científicas en congresos, libros y revistas de carácter nacional e internacional que se describen en el apartado 8.3.

Esta tesis doctoral es un trabajo amplio que deja muchas puertas abiertas para continuar con la investigación realizada. Hubiera sido del todo imposible abarcar por completo cada uno de los aspectos abordados en la misma. El apartado 8.4 apunta algunos trabajos futuros que se podrían realizar continuando el trabajo.

8.2 Aportaciones

Como resultado del trabajo realizado para esta tesis han surgido algunas aportaciones de carácter científico-técnicas que se mencionan a continuación:

- La aportación más importante es la que cubre el objetivo principal de la misma, es decir, la definición precisa de un modelo de proceso para el desarrollo de interfaces de usuario de sistemas CSCW orientado a los usuarios y dirigido por tareas, así como la metodología a seguir en cada una de sus etapas.
- Un Método de clasificación de herramientas groupware basada en los trabajos de los investigadores más relevantes del área y que supera muchas de las limitaciones de métodos previos.
- Un Modelo conceptual para entornos CSCW que permite trabajar con un vocabulario común y que estructura términos relacionados.
- Adaptación de la etapa de elicitación de requisitos de Sistemas de Información a una elicitación centrada en entornos CSCW.
- Propuesta de la metodología de análisis en entornos CSCW: estructura y comportamiento. Definición de nuevos diagramas para su especificación.
- Adaptación del diseño de interfaces en IPO (interacción persona-ordenador) al diseño de interfaces para sistemas CSCW: definición de nuevas facetas en base a criterios de percepción, extensión de la notación para darle soporte (definición de nuevos AIOs), definición de nuevos diagramas para la representación de la navegación y la presentación.
- Adaptación de la implementación de interfaces en IPO a la implementación de interfaces para sistemas CSCW: nuevos CIOs, CIOs compuestos.
- Establecimiento de la trazabilidad inter- e intraetapa en el modelo de proceso.
- Prototipo de una herramienta CASE que da soporte al modelo de proceso y la metodología presentados.

8.3 Publicaciones

Fruto del desarrollo de esta tesis doctoral, los trabajos y aportaciones preliminares realizados han sido publicados en diferentes congresos y revistas nacionales e internacionales. Concretamente se han obtenido 6 revistas internacionales (5 LNCS), 9 congresos internacionales, 6 congresos nacionales y 1 capítulo de libro internacional. A continuación se detallan dichas publicaciones:

- **Víctor M. R. Penichet**, María D. Lozano, J.A. Gallud, R. Tesoriero: *Task Modelling for Collaborative Systems*. 6th International workshop on Task Models and Diagrams: TAMODIA 2007. Lecture Notes in Computer Science (LNCS), Springer Verlag. Toulouse, France, 2007.

Descripción de una propuesta de modelado de tareas para entornos CSCW.

- **Víctor M. R. Penichet**, María D. Lozano, J.A. Gallud: *An Ontology to Model Collaborative Organizational Structures in CSCW Systems*. Capítulo de libro internacional pendiente de publicar, editorial Springer. Seleccionado en Interacción 2006 para su extensión. 2007.

Presenta el modelo conceptual definido para describir entornos CSCW. Términos como rol, estructura organizativa, grupo, tarea de grupo, etc.

son conceptos fundamentales para la especificación de aplicaciones groupware. El modelo conceptual define y relaciona estos términos.

- **Victor M. R. Penichet**, María D. Lozano, J.A. Gallud, R. Tesoriero. *CE4WEB: Una Herramienta CASE Colaborativa para el Modelado de Aplicaciones con UML*. XII Jornadas de Ingeniería del Software y Bases de Datos, JISBD'2007 en el marco del II Congreso Español De Informática (CEDI 2007). Ed. Thomson-Paraninfo; Zaragoza, España. 2007.

Demostración del funcionamiento de la herramienta CE4WEB para la edición cooperativa de diagramas de clases UML que incluye gestión de versiones, sesiones, bloqueos, Chat, etc.

- **Victor M. R. Penichet**, Maria D. Lozano, José A. Gallud, Ricardo Tesoriero. *Análisis en un Modelo de Proceso CSCW. Organización, Roles e Interacción Persona-Ordenador-Persona*. VIII Congreso Internacional de Interacción Persona-Ordenador, Interacción 2007 (AIPO) en el marco del II Congreso Español De Informática (CEDI 2007). Ed. Thomson-Paraninfo; Zaragoza, España. 2007.

El trabajo describe una metodología para realizar el análisis en sistemas CSCW donde la estructura organizativa, los roles y las interacciones entre los actores del sistema se tienen en cuenta de un modo particular puesto que se trata de sistemas específicos.

- **Victor M. R. Penichet**, Ismael Marin, Jose A. Gallud, Maria D. Lozano, Ricardo Tesoriero. *A Classification Method for CSCW Systems*. Electronic Notes in Theoretical Computer Science, ENTCS. Ed. Elsevier Science Publishers B. V. ISSN: 1571-0661 Vol. 168, pp. 237-247. The Netherlands; 2007.

Se trata de una clasificación avanzada de herramientas basada en resultados de los investigadores más prestigiosos del área del CSCW que soluciona problemas en clasificaciones anteriores.

- Jose A. Gallud, Maria Lozano, Ricardo Tesoriero, **Victor M. R. Penichet**. *Using Mobile Devices to Improve the Interactive Experience of Visitors in Art Museums*. Proceedings of the 12th Human-Computer Interaction International 2007 - HCI International 2007. Lecture Notes in Computer Science, Springer Verlag. Volume 2, LNCS_4551, ISBN: 978-3-540-73106-1 Beijing International Convention Center; Beijing, China; 22-27 July 2007.

Describe un sistema desarrollado para facilitar visitas interactivas en el Museo de la Cuchillería de Albacete. El sistema consiste, muy básicamente, en unas PDAs como dispositivos de interacción, una red inalámbrica como tecnología de interconexión y el software que lo maneja todo y que hace al usuario la visita más agradable, sencilla y didáctica.

- M. Lozano, R. Tesoriero, J. A. Gallud, **V. M. R. Penichet**. *A Mobile Software Developed for Art Museums: Conceptual Model and Architecture*. Proceedings del 3rd International Conference on Web Information Systems and Technologies - WEBIST 2007. Universidad de Cataluña, Barcelona, Spain; 06 Mar 2007.

Describe la arquitectura y el modelo conceptual empleados en el desarrollo de un sistema para museos que permite al usuario realizar una visita guiada aprovechando el uso de las nuevas tecnologías: PDAs, redes inalámbricas, etc.

- R. Tesoriero, M. Lozano, J. A. Gallud, **V. M. R. Penichet**. *Evaluating the Users' Experience of a PDA-Based Software Applied in Art Museums*. Proceedings del

3rd International Conference on Web Information Systems and Technologies - WEBIST 2007. Universidad de Cataluña, Barcelona, Spain; 06 Mar 2007.

El trabajo muestra una evaluación sobre una aplicación creada para sustituir las tradicionales audio-guías en los museos por sistemas más novedosos haciendo uso de tecnologías de última generación como las PDA y las redes inalámbricas.

- **Victor M. R. Penichet**, Maria D. Lozano, Jose A. Gallud, Francisco Montero. *Ontología para Estructuras Organizativas Colaborativas*. Proceedings del VII Congreso Internacional de Interacción Persona-Ordenador - Interacción 2006. ISBN 84-690-1613-X. Universidad de Castilla-La Mancha, Puertollano, Ciudad Real, Spain; 15 Nov 2006.

Se presenta un modelo conceptual para el modelado de la estructura organizativa y las interacciones cooperativas de un entorno colaborativo.

- Francisco L. Gutiérrez, **Victor M. R. Penichet**, José L. Isla, Francisco Montero, María D. Lozano, José Antonio Gallud, María Luisa Rodríguez. *Un Marco Conceptual para el Modelado de Sistemas Colaborativos Empresariales*. Proceedings del VII Congreso Internacional de Interacción Persona-Ordenador - Interacción 2006. ISBN 84-690-1613-X. Universidad de Castilla-La Mancha, Puertollano, Ciudad Real, Spain; 15 Nov 2006.

Se describe un marco conceptual para la modelización de sistemas interactivos que tienen lugar bajo un entorno organizativo complejo. Colaboración y entornos empresariales son dos conceptos clave en este trabajo.

- **Victor M. R. Penichet**, María Dolores Lozano, Jose A. Gallud. *Describing Group Tasks in Multi-User Systems*. Proceedings of the 4th Latin American Web Congress, La-Web 2006. IEEE Computer Society Press, ISBN: 0-7695-2693-4. Universidad de las Americas, Puebla, Cholula, Mexico; 26 Oct 2006.

Es un trabajo donde se describen las tareas cooperativas, cuáles son las propiedades que las caracterizan y que se han de tener en cuenta a la hora de desarrollar una aplicación groupware.

- **Victor M. R. Penichet**, C. Calero, Maria D. Lozano, M. Piattini. *Using WQM for Classifying Usability Metrics*. Proceedings of the IADIS International Conference WWW/Internet 2006 - ICWI 2006; ISBN: 972-8924-19-4. Murcia, Spain; 05 Oct 2006.

Se trata de un trabajo realizado junto con investigadores del campus de Ciudad Real (Universidad de Castilla-La Mancha) en el que, haciendo uso de un modelo tridimensional de calidad (WQM) elaborado en el grupo de investigación Alarcos, se clasifican una serie de métricas de usabilidad.

- Aurelio Martinez, **Victor M. R. Penichet**, Maria D. Lozano, Jose A. Gallud. *Propuesta de un Diseño de un Editor UML Colaborativo Basado en Web*. IV Taller en Sistemas Hipermedia Colaborativos y Adaptativos. Sitges, Spain; 03 Oct 2006.

CE4Web es una herramienta cooperativa que permite a varios usuarios diseñar diagramas de clase UML en tiempo real. El versionado, histórico, bloqueo / desbloqueo, etc. son características también contempladas en esta herramienta.

- **Victor M. R. Penichet**, J. A. Gallud, M. Lozano, M. Tobarra. *Web Management of Citizens' Complaints and Suggestions*. Proceedings of the 12th International

Workshop on Groupware - CRIWG 2006. Lecture Notes in Computer Science, Springer Verlag; ISBN: 3-540-39591-1. Valladolid, Spain; 20 Sep 2006. Índice de Aceptación = 35%

Descripción del análisis de un flujo de trabajo sobre un procedimiento de quejas y reclamaciones implantado con posterioridad en el Ayuntamiento de Albacete.

- Stéphane Chatty, José Creissac Campos, María Paula González, Sophie Lepreux, Erik G. Nilsson, **Victor M. Ruiz Penichet**, Mónica Santos, Jan Van den Bergh. *Processes: Working Group Report*. Proceedings DSV-IS 2006. Lecture Notes in Computer Science, Springer Verlag; ISBN: 978-3-540-69553-0. vol 4323, pp 262-264. Dublin, Ireland. 27 Jul 2006. Índice de Aceptación = 33%

Tras la reunión de trabajo mantenida en el congreso, se escribe un artículo conjunto sobre procesos. Una lluvia de ideas acerca de su estado, lo que ha pasado hasta ahora y por dónde deberían ir.

- **Victor M. R. Penichet**, Fabio Paternò, J. A. Gallud, M. Lozano; *Collaborative Social Structures and Task Modelling Integration*. Proceedings DSV-IS 2006. Lecture Notes in Computer Science, Springer Verlag; ISBN: 978-3-540-69553-0. vol 4323, pp 67-80. Dublin, Ireland. 27 Jul 2006. Índice de Aceptación = 33%

Fruto de la estancia de investigación realizada en el CNR de Pisa surge este trabajo que combina el modelado de tareas con la notación CTT (una de las más extendidas) con el estudio de estructuras colaborativas.

- **Victor M. R. Penichet**, J. A. Gallud, M. López, M. Lozano: *Administrative Information Web-based Collaborative Centre*. IADIS International Conference - WWW/Internet 2005. Lisboa, Portugal. 2005. Índice de Aceptación = 25%

Como fruto del trabajo realizado con el Ayuntamiento de Albacete sobre los diferentes procedimientos administrativos comunes y la problemática de su ordenación y fácil acceso, se elaboró una herramienta colaborativa basada en Web por medio de la cual los administrativos pueden almacenar y mantener actualizados los procedimientos administrativos, a la vez que los ciudadanos pueden acceder por medio de Internet al repositorio generado.

- **Victor M.R. Penichet**, J.A. Gallud, M.D. Lozano: *Clasificación No Excluyente de Funciones y Herramientas CSCW*. Interacción 2005 - VI Congreso de Interacción Persona-Ordenador (AIPO) en el marco del I Congreso Español De Informática (CEDI). Ed. Thomson-Paraninfo; ISBN: 84-9732-436-6. Granada, España. 2005.

Tras el estudio realizado sobre diferentes taxonomías de entornos CSCW se observan unas deficiencias provocadas por la evolución de las herramientas y sistemas que son cada vez más complicados. Estas complejidades dificultan la clasificación de las aplicaciones. En este trabajo se propone una nueva taxonomía no excluyente para solucionar este problema.

- **Victor M. R. Penichet**, J. A. Gallud, A. Fernández-Caballero, M. Lozano: *Complaints and Suggestions Web-Based Collaborative Procedure*. EGOV05 - International Conference on E-Government, Ed. Trauner Verlag, Linz. ISBN: 3-85487-830-3. Copenhagen, Dinamarca. 2005.

Explica la utilización de unos agentes inteligentes sobre el flujo de trabajo de un procedimiento de quejas y sugerencias que lo automatizan en mayor medida y que lo hacen más rápido y eficiente.

- Francisco Javier García; María Dolores Lozano; Francisco Montero; Jose Antonio Gallud; **Víctor M.R. Penichet**; Carlota Lorenzo: *Survey on Quality Models to Measure the Quality of Web Sites and Applications*. HCI Internacional 2005. Ed. Lawrence Erlbaum Associates, Inc (LEA); ISBN: 0-8058-5807-5. Las Vegas, Nevada, USA. 2005.

Es una revisión y estudio de algunos modelos de calidad que se aplican al desarrollo de Aplicaciones Web.

- Jose Antonio Gallud, **Víctor M. R. Penichet**, Luis Argandoña: *Digital Museum: A Multi-technological Approach*. HCI Internacional 2005. Ed. Lawrence Erlbaum Associates, Inc (LEA); ISBN: 0-8058-5807-5. Las Vegas, Nevada, USA. 2005.

El grupo LoUISE participó en la consultoría, análisis y gestión del proyecto telemático del Museo Municipal de la Cuchillería de Albacete. Se desplegó un sistema en el que diferentes usuarios accedían desde distintos dispositivos a través de una red inalámbrica para visitar el museo, gestionar las piezas que en él se encuentran, realizar compras, etc. virtualmente.

- **V. M. Ruiz Penichet**, J. A. Gallud, M. L. González, P. González: *Implantation Guide for Collaborative Web-Based Systems (IGCWS)*. DEXA Workshops 2004, IEEE Computer Society Press, ISBN: 0-7695-2195-9 ISSN 1529-4188; pp. 259 - 263. Zaragoza (Spain) 2004.

Fue la primera aproximación al despliegue de un sistema cooperativo, en la que se elaboró una guía básica de implantación de estos sistemas y que se experimentó con dos casos de estudio reales.

8.4 Trabajos Futuros

A continuación se describen algunos trabajos que se podrían realizar a partir de los estudios y aportaciones realizados en esta tesis doctoral. De hecho, algunos de los puntos ya están siendo extendidos con la tesis doctoral de otros compañeros:

- Extensión del modelo conceptual presentado de manera que considere aspectos dinámicos en sistemas CSCW como el cambio de rol de los usuarios, que adapte y amplíe el modelo de sesión adoptado a las necesidades concretas planteadas.
- Introducción de las fases de análisis y la validación de los requisitos en la primera etapa del modelo de proceso: elicitación de requisitos. Adicionalmente, para ello, habría que estudiar cómo afectan las modificaciones introducidas en esta metodología a la metodología de [Durán, 2000]. Entre otras cosas, se deberían considerar cómo afectan las extensiones colaborativas introducidas.
- Desarrollo de un toolkit con las herramientas más frecuentes en las aplicaciones groupware que facilite el desarrollo de este tipo de aplicaciones, de un modo similar a como se hace en el grupo de Saul Greenberg en Universidad de Calgary, pero pudiendo integrar un conjunto completo de estas herramientas para desarrolladores.
- Estudio y desarrollo de patrones CSCW que permitan realizar aplicaciones groupware de calidad en un tiempo menor.

- Aplicación de criterios de evaluación y calidad en sistemas CSCW. Este punto es el tema de la tesis de Manuel Tobarra, compañero del grupo LoUISE, que continua con el estudio de este tipo de sistemas desde esta perspectiva.
- Adaptación y ampliación de los criterios de accesibilidad y usabilidad a aplicaciones CSCW que permitan el uso de estas aplicaciones al mayor número de personas posibles (que por su discapacidad no puedan hacerlo) y de la forma más sencilla posible. Concretamente, en relación a este punto, la tesis ha suscitado un proyecto relacionado con accesibilidad. Un proyecto en el que se propone aportar nuevas técnicas de accesibilidad de sitios Web, considerando tanto aspectos tecnológicos -hardware y software-, como servicios específicos dirigidos al colectivo de discapacitados de la región, concretamente servicios jurídicos.
- El modelo de proceso presentado en la tesis facilita el diseño de interfaces de usuario para entornos CSCW pero no se centra en la funcionalidad de este tipo de aplicaciones como se ve reflejado al considerar únicamente el diseño de la interfaz de usuario en esta etapa. Puesto que el diseño de un sistema además se puede abordar desde la estructura interna de los *datos*, su *arquitectura* y desde el diseño *procedimental* de los componentes software, son tres nuevas posibles ampliaciones del trabajo realizado.
- Extender / enriquecer el modelo de proceso con otros aspectos como planificación, análisis de riesgos o etapas específicas para ubicar el prototipo y la evaluación.
- Mejora de la herramienta CASE desarrollada depurando y evolucionando el prototipo desde dos perspectivas: por un lado el diseño y usabilidad de la aplicación; y, por otro lado, mejorando la automatización y trazabilidad en la aplicación puesto que se ha visto la posibilidad de generar algunos diagramas e incluso la propia interfaz de usuario desde la etapa de elicitación de requisitos de forma automática o semiautomática.
- Extensión del lenguaje UsiXML con las propuestas realizadas en esta tesis para que contemple aspectos CSCW.
- Trabajar en la extensión de modelo de proceso para contemplar mecanismos que faciliten los protocolos de comunicación, el control de acceso y las notificaciones, aspectos fundamentales de las aplicaciones groupware [Ellis, 1991].

REFERENCIAS

- Aalst, 2004 Aalst, Wil van der; Hee, Kees van. *Workflow Management: Models, Methods, and Systems*; The MIT Press (March 1, 2004). ISBN: 0262720469W
- Alamán, 1999 Alamán, X.; Cobos, R.: *KnowCat: a Web Application for Knowledge Organization*. Proceedings of the World-Wide Web and Conceptual Modeling (WWWCM'99). París, Francia, Noviembre, 1999. P.P Chen et al (Eds). Lecture Notes in Computer Science 1727, pp. 348-359 (1999)
- Ali, 2004 Ali M.F., Pérez-Quiñones M.A., Abrams M.: *Building Multi-Platform User Interfaces with UIML*. In: Seffah, A., Javahery, H. (eds.): *Multiple User Interfaces: Engineering and Application Framework*. John Wiley, Chichester (2004) 95–118
- Andriessen, 2003 Andriessen, Erik, J.H. (2003): *Working with Groupware. Understanding and Evaluating Collaboration Technology*. Springer
- Annett, 1967 Annett J., Duncan K.: *Task Analysis and training in design*. *Occupational Psychology* 41,211-221, 1967
- Antunes, 1994 Antunes, P.; Guimaraes, N. (1994). *Multiuser Interface Design in CSCW Systems*. Technical University of Lisboa - INESC
- Bannon, 1989 Bannon, L.J. y Schmidt, K. (1989): *CSCW: Four characters in Search of a context*. Primera conferencia europea en CSCW
- Balzert, 1996 Balzert, H.: *From OOA to GUIs: The JANUS System*. *Journal of Object-Oriented Programming*, vol 8(9), pp. 43–47, feb, 1996
- Beck, 2000 Beck, K. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman Publishing Co., Inc. 2000
- Bernes-Lee, 2001 Berners-Lee, Tim; Hendler, James; and Lassila, Ora: *The Semantic Web*, *Scientific American*, May 2001.
<http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>
- Bodart, 1990 Bodart, F., Hennebert, A.-M., Leheureux, J.-M., Provot, I., Sacré, B. and Vanderdonckt, J.: *The TRIDENT project (Tools*

- for an Interactive Development Environment*). Working Conference IFIP WG 2.7, Namur, September 1990
- Bodart, 1995 Bodart, F. et al: *Towards a Systematic Building of Software Architectures: the TRIDENT Methodological Guide*. In Proc. Of Eurographics Workshop on Design, Specification, Verification of Interactive Systems. DSV-IS'95. Eurographics Series, pp. 237-253. June, 1995
- Boyle, 2002 Boyle, M. and Greenberg, S. (2002). *GroupLab Collabratory: A Toolkit for Multimedia Groupware*. In J. Patterson (Ed.) *ACM CSCW 2002 Workshop on Network Services for Groupware*, November
- Booch , 1999 Booch, G.; Rumbaugh, J.; Jacobson, I.: *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- Borst, 1997 Borst, W.N (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. CTIT Ph.D-thesis series No.97-14. University of Twente. Enschede, The Netherlands.
- Brackett, 1990 Brackett, J. W.: *Software Requirements*. Curriculum Module SEI-CM-19-1.2, Software Engineering Institute, Carnegie Mellon University, 1990.
- Brickley, 2005 Brickley, D., Miller, L: *FOAF Vocabulary Specification*. <http://xmlns.com/foaf/0.1/>, 2005
- BSCW, 1995 BSCW © 1995-2007 Fraunhofer FIT and OrbiTeam Software GmbH & Co. KG. <http://bscw.fit.fraunhofer.de/>
- Cachero, 2003 Cachero, C.: *OO-H: una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales*. Tesis doctoral. Universidad de Alicante, 2003
- Calvary, 2003 Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: *A Unifying Reference Framework for Multi-Target User Interfaces*. *Interacting with Computers*. Vol. 15, No. 3, June 2003, pp. 289-308
- Campos, 2005 Campos, M. I. (2005). *Relaciones Interpersonales*. Trabajo en equipo, Asociación Alcalá.
- Card, 1983 Card S., Moran T., Newell A.: *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum, Hillsdale - 1983
- Chatty**, 2006 Chatty, Stéphane; Creissac Campos, José; González, María Paula; Lepreux, Sophie; Nilsson, Erik G.; **Penichet, Víctor M. R.**; Santos, Mónica; Van den Bergh, Jan: *Processes: Working Group Report*. DSV-IS 2006. Lecture Notes in Computer Science, Springer Verlag, 262-264; I.S.B.N.: 978-3-540-69553-0. Dublin, Ireland. 27 Jul 2006
- Chung, 2004 Chung, G., Dewan, P: *Towards dynamic collaboration architectures*. Proceedings of the 2004 ACM conference on Computer supported cooperative work, 2004. p 1-10. ISBN:1-58113-810-5
- Cobos, 2002 Cobos, Ruth; Esquivel, José A.; Alamán, Xavier: *IT Tools for Knowledge Management: A Study of the Current Situation*. 2002. http://www.kmadvantage.com/docs/km_articles/IT_Tools_for_KM.pdf

- Coleman, 1997 Coleman, David (1997): *Groupware: Collaborative Strategies for Corporate LANs and Intranets*, Amazon
- Coleman, 1998 Coleman, D.. A Use Case Template: Draft for Discussion. Fusion Newsletter, Abril 1998.
- Constantine, 1999 Constantine, L. L. and Lockwood, L. A. D., *Software for use: a practical guide to the models and methods of usage-centered design*, Addison Wesley, Reading, Mass, 1999.
- Coyette, 2005 Coyette, A., Vanderdonckt, J., *A Sketching Tool for Designing Anyuser, Anyplatform, Anywhere User Interfaces*, Proc. of Interact'2005
- Crease, 1999 Crease, M., Gray, P., and Brewster, S.A. Resource Sensitive Multi-Modal Widgets, in Volume II of the Proceedings of INTERACT '99 (Edinburgh, UK), British Computer Society, 1999, pp. 21-22.
- Čubranić, 2004 Čubranić, Davor; Murphy, Gail C.; Singer, Janice; Booth, Kellogg S.: *Learning from Project History: A Case Study for Software Development*. Proceedings of the 2004 ACM conference on Computer supported cooperative work, 2004. p 82-91
- Dery-Pynna, 2003 Dery-Pinna, A.-M., Fierstone, J., Picard, E.: Component Model and Programming: a First Step to Manage Human-Computer Interaction Adaptation. In Proc. of 5th Int. Symposium on Human-Computer Interaction with Mobile Devices and Services MobileHCI'2003 (Udine, September 8-11, 2003). Lecture Notes in Computer Science, Vol. 2795. Springer-Verlag, Berlin (2003) 456-460
- DeSanctis, 1987 DeSanctis, G., and Gallupe, B., 1987. *A foundation for the study of group decision support systems*. Management Science, 33, 5, 589-609
- DoD, 1994 DoD. Military Standard 498: *Software Development and Documentation*. Department of Defense of the United States of America, 1994. <http://www-library.itsi.disa.mil/mil std/498 win3.exe>
- Durán, 2000 Durán, Amador: *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. Tesis Doctoral. Universidad de Sevilla. 2000
- Dutta, 1997 Dutta, S. y De Meyer, A. (1997): *Knowledge Management at Arthur Andersen (Denmark): Building Assets in Real Time and in Virtual Space*. INSEAD, Fontainebleau.
- Dumont, 2001 Dumont, A.; Pietrobon, C.: A Method for Specification of Collaborative Interfaces through the Use of Scenarios. CSCWD2001, 2001
- Eckstein, 1998 Eckstein, R.: *Java Swing*. O'Reilly, 1st edition, 1998
- EclipseUML, Web EclipseUML, <http://www.eclipseuml.com/>
- Eisenstein, 2001 Eisenstein, J., Vanderdonckt, J., Puerta, A.: Model-Based User-Interface Development Techniques for Mobile Computing. In Lester J. (ed.): Proc. of 5th ACM Int. Conf. on Intelligent User Interfaces IUI'2001 (Santa Fe, January 14-17, 2001). ACM Press, New York (2001) 69-76

- Ellis, 1991 Ellis, C., S. Gibbs, et al: *Groupware: Some Issues and Experiences*. Communications of ACM 34(1): 39-58. 1991
- Ellis, 1994 Ellis, C. and Wainer, J. 1994. *A conceptual model of groupware*. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (Chapel Hill, North Carolina, United States, October 22 - 26, 1994). CSCW '94. ACM Press, New York, NY, 79-88
- Endsley, 1995 Endsley, M. (1995) Toward a Theory of Situation Awareness in Dynamic Systems, Human Factors, 37(1), 32-64
- Engelbart, 2004 Blog dedicado a Douglas Engelbart.
<http://douglasengelbart.motime.com/>
- England, 1998 England D., Gray P.D.: Temporal aspects of interaction in shared virtual worlds. *Interacting with Computers*, Vol. 11, pp 87-105, 1998
- Enterprise Architect, Web Enterprise Architect for UML,
<http://www.sparxsystems.com/ea.htm>
- Fernández, 2003 Fernández Breis, Jesualdo Tomás: *Un entorno de integración de ontologías para el desarrollo de sistemas de gestión del conocimiento*. Tesis Doctoral. Directores: Martín Rubio, Fernando; Martínez Béjar, Rodrigo. Mu-2060-2005/84-689-5048-3. Universidad de Murcia. 2003
- Foley, 1991 Foley, J.D.; Kim, W.C.; Kovacevic, S.; Murray, K.: *Intelligent User Interfaces*. Chapter 15. UIDE an Intelligent User Interface Design Environment, pp. 339-384. ACM Press, Addison Wesley, Reading, MA, USA., 1991. 18, 20, 375.
- Gallud, 2005** Gallud, Jose A.; **Penichet, Victor M. R.**; Argandoña, Luis, Gonzalez, Pascual; García, J. A.; *Digital Museum: A Multi-technological Approach*. HCI International 2005. Ed. Lawrence Erlbaum Associates, Inc (LEA); I.S.B.N.: 0-8058-5807-5. Las Vegas, Nevada USA; 26 Jul, 2005
- Gallud, 2007** Gallud, Jose A.; Lozano, Maria; Tesoriero, Ricardo; **Penichet, Victor M. R.**: *Using Mobile Devices to Improve the Interactive Experience of Visitors in Art Museums*. Proceedings of the 12th Human-Computer Interaction International 2007 - HCI International 2007. Lecture Notes in Computer Science, Springer Verlag. Beijing International Convention Center; Beijing, China; 22-27 July 2007.
- García, 2005** García C., F. Javier; Lozano, María; Montero, Francisco; Gallud, Jose Antonio; **Penichet, Victor M. R.**; Lorenzo, Carlota: *Survey on Quality Models to Measure the Quality of Web Sites and Applications*. HCI International 2005. Ed. Lawrence Erlbaum Associates, Inc (LEA); I.S.B.N.: 0-8058-5807-5. Las Vegas, Nevada USA; 26 Jul, 2005
- Garrido , 2003 Garrido Bullejos, José Luis; *AMENITIES: Una metodología para el desarrollo de sistemas cooperativos basada en modelos de comportamiento y tareas*. Tesis Doctoral. Granada 2003
- Garrido , 2005 Garrido, J.L., Gea, M., Rodríguez, M.L.: *Requirements Engineering in Cooperative Systems*. Requirements Engineering for Sociotechnical Systems, Chapter XIV, ISBN 1-59140-506.8, IDEA GROUP, Inc. USA. (2005) pp. 226-244

- Garrido, 2007 Garrido, J. L., Noguera, M., González, M., Hurtado, M. V., Rodríguez, M. L.: *Definition and Use of Computation Independent Models in an MDA-Based Groupware Development Process*. Journal of Science of Computer Programming 66, Elsevier, ISSN 0167-6423, (2007), pp. 25-43
- Goguen, 1994 Goguen, J.A. *Requirements Engineering as the Reconciliation of Social and Technical Issues*. En Requirements Engineering: Social and Technical Issues, páginas 165–199. Academic Press, 1994.
- Goleman, 1996 Goleman, D.: La Inteligencia Emocional. Kairós, 1996.
- Greenberg, 1991 Greenberg, S. (1991): Computer-supported co-operative work and groupware: an introduction to the special issue. Jornadas Internacionales Man Machine Studies
- Greenberg, 2000 Greenberg, S. (2000): *Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration*. Despartament of Computer Science, University of Calgary, Canadá
- Greif, 1988 Greif, I.; Computer-Supported Cooperative Work: A Book of Readings. Morgan Kaufmann, San Mateo CA, 1988
- Griffiths Griffiths, Richard: *Computer Supported Co-operative Work (CSCW) and Groupware*.
<http://www.it.bton.ac.uk/staff/rng/teaching/notes/CSCWgroupware.html>
- Gross, 2004 Gross, T. The Past, Present, and Future of Workgroups in a Theatre of Work. Presented at Workshop on Forecasting Presence and Availability at the Conference on Human Factors in Computing Systems - CHI 2004 (Apr. 26, Vienna, Austria). Tullio, J., Begole, J.B., Horvitz, E. and Mynatt, E., eds. 2004
- Gruber, 1993 Gruber, T.R.: *A translation approach to portable ontologies*. Knowledge Acquisition, 5(2):199-220, 1993
- Gruber, 2005 Gruber, Thomas. *Ontology of Folksonomy: A Mash-up of Apples and Oranges*. Invited paper/keynote to the First on-Line conference on Metadata and Semantics Research (MTR'05), November 2005.
- Grudin, 1988 Grudin, J.: *Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces*. 1988. Proc. CSCW 88, 85-93.
- Grudin, 1994 Grudin, J. 1994. *Computer-Supported Cooperative Work: History and Focus*. Computer 27, 5 (May. 1994), 19-26
- Grudin, 1994 (b) Grudin, J. 1994. *Groupware and social dynamics: eight challenges for developers*. Commun. ACM 37, 1 (Jan. 1994), 92-105
- Guerrero, 2007 Guerrero, J., Vanderdonckt, J., *FlowiXML: a Step towards Designing Workflow Management Systems*, Journal of Web Engineering, 2007
- Gutiérrez, 2006 Gutiérrez, Francisco L.; **Penichet, Victor M. R.**; Isla, José L.; Montero, Francisco; Lozano, María D.; Gallud, José Antonio; Rodríguez, María Luisa. *Un Marco Conceptual para el Modelado de Sistemas Colaborativos Empresariales*. Proceedings del VII

- Congreso Internacional de Interacción Persona-Ordenador - Interacción 2006. Universidad de Castilla-La Mancha, Puertollano, Ciudad Real, Spain; 15 Nov 2006.
- Gutwin, 1996 Gutwin, C.; Greenberg, S.: *Workspace Awareness for Groupware*, Proceedings of the ACM Conference on Human Factors in Computing Systems, 208-209 1996
- Gutwin, 1997 Gutwin, C., 1997. *Workspace awareness in real-time groupware environments*. Ph.D. Thesis, Department of Computer Science, University of Calgary, Calgary, Canada.
- Gutwin, 2002 Gutwin, C., and Greenberg, S. (2002) *A Descriptive Framework of Workspace Awareness for Real-Time Groupware*. *Computer Supported Cooperative Work*, 11(3-4), 411-446, Special Issue on Awareness in CSCW, Kluwer Academic Press
- Gutwin, 2004 Gutwin, C., and Greenberg, S. (2004): *The Importance of Awareness for Team Cognition in Distributed Collaboration*. In E. Salas and S. M. Fiore (Editors) *Team Cognition: Understanding the Factors that Drive Process and Performance*, 177-201, Washington:APA Press.
- Gutwin, 2004b Gutwin, Carl; Penner, Reagan; Schneider, Kevin; *Group awareness in distributed software development*. Proceedings of the 2004 ACM conference on Computer supported cooperative work, 2004. p 72-81
- Harel, 1987 Harel, D.: *Statecharts: A Visual Formalism for Complex Systems*. In *Science of Computing Programming* 8, 1987
- Haynes, 2004 Haynes, Steven R.; Puroo, Sandeep; Skattebo, Amie L.; *Situating evaluation in scenarios of use*. Proceedings of the 2004 ACM conference on Computer supported cooperative work, 2004. p 92-101
- Horn, 2004 Horn, Daniel B., Finholt, Thomas A., Birnholtz, Jeremy P., Motwani, D., Jayaraman, S.: *Six degrees of Jonathan Grudin: a social network analysis of the evolution and impact of CSCW research*. ACM conference on Computer supported cooperative work, 2004. p 582 - 591
- Hurwitz, 2001 Hurwitz Report (2001). *Collaborative UML Development*. White Paper, CanyonBlue Incorporated, November 2001. URL www.canyonblue.com/whitepapers.htm
- IEEE, 1990 IEEE. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Standard 610.12-1990, Institute of Electrical and Electronics Engineers, 1990.
- Jacobson, 1992 Jacobson, I.; Christerson, M.; et al. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1992.
- Jacobson, 1993 Jacobson, I.; Christerson, M.; Jonsson, P. y Övergaard, G.: *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 4a edición, 1993.
- Jacobson, 2000 Jacobson, I; Booch, G.; Rumbaugh, J.: *El Proceso Unificado de Desarrollo de Software*, Addison-Wesley, 2000

- Johansen, 1988 Johansen, R. (1988): *Groupware: Computer support for business teams*. New York: The Free Press
- Johnson-Lenz, 1981 Johnson-Lenz, P. and Johnson-Lenz, T.: *Consider the Groupware: Design and Group Process Impacts on Communication in the Electronic Medium*. In Hiltz, S. and Kerr, E. (Ed.): *Studies of Computer-Mediated Communications Systems: A Synthesis of the Findings*, (Research Report, Band 16) Computerized Conferencing and Communications Center, New Jersey Institute of Technology: Newark, New Jersey, 1981
- Khoshafian, 1995 Khoshafian, S., Buckiewicz, M. (1995): *Introduction to Groupware, Workflow, and Workgroup computing*. John Wiley & Sons, Inc. ISBN 0-471-02946-7
- Kirsh, 2004 Kirsh, David: *Methodologies for Evaluating Collaboration Behavior in Co-Located Environments*. CSCW 2004 Workshop: Methodologies for Evaluating Collaboration in Co-Located Environments. 2004
- Lee, 2000 Lee Chan, S. (2000): *Synchronous Collaborative Development of UML Models on the Internet*. International Conference on Computer Supported Cooperative Work in Design (CSCWD 2000)
- Lepreux, 2006 Lepreux S., Vanderdonckt, J., *Visual Design of User Interfaces by (De)composition*, Proc. of DSV-IS'2006
- Limbourg, 2004 Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., López Jaquero, V. *UsiXML: a Language Supporting Multi-Path Development of User Interfaces*, Proc. of 9th IFIP Working Conference on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004 (Hamburg, July 11-13, 2004). LNCS, Vol. 3425, Springer-Verlag, Berlin, Germany, 2005
- Limbourg, 2005 Limbourg, Quentin: *Multi-Path Development of User Interfaces*. PhD Thesis. Université catholique de Louvain. Belgium, 2005
- LookSmart, 2003 LookSmart, Ltd. *FURL*. <http://furl.net>. Founded by Mike Giles
- López, 2004 López-Jaquero, V.; Montero, F.; Molina, J.P.; González, P.; Fernández-Caballero, A.: *A Seamless Development Process of Adaptive User Interfaces Explicitly Based on Usability Properties*. Proc. of 9th IFIP Working Conference on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004 (Hamburg, July 11-13, 2004). Lecture Notes in Computer Science, Vol. 3425, Springer-Verlag, Berlin, 2005.
- López, 2005 López Jaquero, Víctor M.: *Interfaces de Usuario Adaptativas Basadas en Modelos y Agentes Software*. Tesis Doctoral. Universidad de Castilla-La Mancha, 2005
- Lozano, 2001 Lozano, María Dolores: *Entorno Metodológico Orientado a Objetos para la Especificación y Desarrollo de Interfaces de Usuario*. Tesis Doctoral. Universidad Politécnica de Valencia. 2001

- Lozano, 2002 Lozano, M., Molina, J.P., Montero, F., González, P. Ramos I. A *Graphical User Interface Development Too*. In the 6th Annual Conference on Human Computer Interaction (HCI'02). ISBN: 1-902505-48-4, Ed. British HCI Group. London, England September, 2002
- Lozano, 2007** Lozano, M.; Tesoriero, R.; Gallud, J. A.; **Penichet, V. M. R.:** *A Mobile Software Developed for Art Museums: Conceptual Model and Architecture*. Proceedings del 3rd International Conference on Web Information Systems and Technologies - WEBIST 2007. Universidad de Cataluña, Barcelona, Spain; 06 Mar 2007.
- Macromedia, 2004 Macromedia Inc., DevNet Resource Kit Volume 6, 2004. Available online: http://www.macromedia.com/software/drk/productinfo/product_overview/volume6/
- Markopoulus, 2000 Markopoulus P.: *Supporting Interaction Design with UML, Task Modelling*. In Proceedings of the TUPIS'2000 Workshop at the UML'2000
- Marshak, 1997 Marshak, R.T., *Workflow: Applying Automation to Group Processes*. In: Coleman, D. (ed.): *Groupware - Collaborative Strategies for Corporate LANs and Intranets*. Prentice Hall PTR, (1997) 143-181
- Mathes, 2004 Mathes, Adam. *Folksonomies: Cooperative Classification and Communication Through Metadata*. Computer Mediated Communication - LIS590CMC. December 2004
- Martínez, 2006** Martinez, Aurelio; **Penichet, Victor M. R.**; Lozano, Maria D.; Gallud, Jose A.: *Propuesta de un Diseño de un Editor UML Colaborativo Basado en Web*. IV Taller en Sistemas Hipermedia Colaborativos y Adaptativos. Sitges, Spain; 03 Oct 2006.
- MAP, 1995 MAP. Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1. Tecnos/Ministerio para las Administraciones Públicas, 1995.
- MAP, 2000 Ministerio de Administraciones Públicas de España. Metodología Métrica v3. <http://www.csi.map.es/csi/metrica3/>, 2000
- Martin, 2004 Martin, D. and Sommerville, I. (2004), 'Patterns of cooperative interaction: Linking ethnomethodology and design'. *ACM Trans. on Computer-Human Interaction (TOCHI)*, 11 (1), 58-89
- McCarthy, 2004 McCarthy, Joseph F.; McDonald, David W.; Soroczak, Suzanne; Nguyen, David H.; Rashid, Al M.: *Augmenting the social space of an academic conference*. Proceedings of the 2004 ACM conference on Computer supported cooperative work, 2004. p 39-48
- McGuinness, 2002 McGuinness, Deborah L.: *Ontologies Come of Age*. In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002

- Mehra, 2004 Mehra, Akhil; Grundy, John; Hosking, John (2004): *Supporting Collaborative Software Design with a Plug-in, Web Services-based Architecture*. Proceedings of the ICSE 2004 Workshop on Directions in Software Engineering Environments (2004)
- Meire, 2003 Meire, A. P.; Borges, M. R. S.; Araujo, R. M.: *Supporting Collaborative Drawing with the Mask Versioning Mechanism*, Proc. International Workshop on Groupware, Autrans, France, Lecture Notes in Computer Science, Berlin, Germany, Vol. 2806, p. 208-223, 2003. ISBN 3-540-20117-3, Springer-Verlag
- Mellouli, 2002 Mellouli, S., Mineau, G. W., and Pascot, D. 2002. *The integrated modeling of multi-agent systems and their environment*. First international Joint Conference on Autonomous Agents and Multiagent Systems: Part 1. AAMAS '02. ACM Press, New York, , 507-508
- Microsoft, 2003a Microsoft (2003): *Guía para la Planificación e Instalación de Microsoft Sharepoint Portal Server 2001*. <http://www.microsoft.com/spain/technet/asuntos/guiasps/default.asp>.
- Microsoft, 2003b Microsoft (2003): Web Oficial de Sharepoint Portal Server, <http://www.microsoft.com/sharepoint/server/default.asp>
- Molina, 2003 Molina Moreno, P.J.: *Especificación de Interfaz de Usuario: De los Requisitos a la Generación Automática*. Tesis doctoral, Universidad de Valencia, 2003
- Molina, 2006 Molina, Ana Isabel: *Una Propuesta Metodológica para el Desarrollo de la Interfaz de Usuario en Sistemas Groupware*. Tesis Doctoral. Universidad de Castilla-La Mancha. 2006
- Montero, 2005 Montero Simarro, Francisco: *Integración de Calidad y Experiencia en el Desarrollo de Interfaces de Usuario Dirigido por Modelos*. Tesis Doctoral. Universidad de Castilla-La Mancha. 2006
- Mori, 2004 Mori G., Paternò F., Santoro C. *Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions*. IEEE Transactions on Software Engineering (August 2004, pp.507-520)
- Morris, 2004 Morris DiMicco, Joan: *Methodologies For Evaluating Collaboration In Collocated Environments*. CSCW 2004 Workshop: Methodologies for Evaluating Collaboration in Co-located Environments. 2004
- MSVisio, Web MS Visio, <http://office.microsoft.com/visio/>
- Neches, 1991 Neches, Robert; Fikes, Richard; Finin, Tim; Gruber, Thomas; Patil, Ramesh; Senator, Ted; and Swartout, William R. *Enabling technology for knowledge sharing*. AI Magazine, 12(3):16-36, 1991.
- Nigay, 1995 Nigay L., Coutaz J., A Generic Platform for Addressing the Multimodal Challenge, in Proceedings of CHI'95, ACM Press, New York, 1995, pp. 98-105.

- Nunes, 2001 Nunes, Nuno J., *Object Modeling for User-Centered Development and User Interface Design: The Wisdom Approach*. Tesis Doctoral. Universidad de Madeira, Portugal. 2001
- Odell, 2005 Odell, J., Nodine, M., and Levy, R.; *A Metamodel for Agents, Roles, and Groups*. Agent-Oriented Software Engineering (AOSE) V, James Odell, P. Giorgini, Jörg Müller, eds., Lecture Notes on Computer Science volume, Springer, Berlin, 2005
- OMG, 2005 Object Management Group. *UML Superstructure Specification, v2.0*; 2005
- OOTC, 1996 IBM OOTC. *Developing Object-Oriented Software: An Experience-Based Approach*. Prentice-Hall, 1996.
- O'Reilly, 2005 O'Reilly, Tim. *What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software*. Published on O'Reilly (<http://www.oreilly.com/>). <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>. 2005
- Orlikowski, 1997 Orlikowski, W.J. y Hoffman, J.D. (1997): *An improvisational model for change management: the case of groupware technologies*. Sloan Management Review
- Pastor, 1997 Pastor, O.; Insfrán, E.; Pelechano, V.; Romero, J.; Merseguer, J.: *OO-Method: An OO Software Production Environment Combining Conventional and Formal Methods*. 9th Conference on Advanced Information Systems Engineering (CAiSE'97). Barcelona, Spain. June 1997. LNCS (1250), pages 145-159. Springer-Verlag 1997. ISBN: 3-540-63107-0.
- Paternò, 1999 Paternò, F.; *Model-based Design and Evaluation of Interactive Applications*. F.Paternò, Springer Verlag, November 1999, ISBN 1-85233-155-0
- Paternò, 2001 Paternò, F.; *Towards a UML for Interactive Systems*. In Engineering for Human-Computer Interaction: 8th IFIP International Conference, EHCI 2001, Toronto, Canada, May 11-13, 2001. Volume 2254/2001. Lecture Notes in Computer Science. Publisher: Springer Berlin / Heidelberg. ISSN 0302-9743
- Paternò, 2006 Paterno, F; Santos, I.: *Designing and Developing Multi-User, Multi-Device Web Interfaces*, Chapter 9, in Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006 Bucharest, Springer Verlag, June 2006, pp. 111-122
- Penichet, 2003 **Penichet, Victor M. R.**: *Sistemas Groupware: Métodos y Herramientas. Estudio de Casos. Proyecto fin de Carrera*. UCLM, Albacete, 2003
- Penichet, 2004 **Penichet, Victor M. R.**; Gallud, Jose A.; López, Miguel; González, Pascual: *Implantation Guide for Collaborative Web-Based Systems (IGCWS)*. DEXA 2004 events, 4th International Workshop on Web Based Collaboration - WBC'04. IEEE Computer Society Press. ISBN: 0-7695-2195-9 ISSN 1529-4188; pp. 259 - 263. Zaragoza, España; Agosto, 2004

- Penichet, 2005c **Penichet, Victor M. R.**; Gallud, Jose A.; López, M.; Lozano, Maria D.: *Administrative Information Web-based Collaborative Centre*. IADIS International Conference - WWW/Internet 2005. Lisboa, Portugal; 20 Oct, 2005
- Penichet, 2005b **Penichet, Victor M. R.**; Gallud, Jose A.; Lozano, Maria D.: *Clasificación No Excluyente de Funciones y Herramientas CSCW*. Interacción 2005 - VI Congreso Internacional Interacción 2005. Ed. Thomson-Paraninfo; I.S.B.N.: 84-9732-436-6. Granada, España; 14 Sep, 2005
- Penichet, 2005a **Penichet, Victor M. R.**; Gallud, Jose A.; Fernández-Caballero, Antonio; Lozano, Maria D.: *Complaints and Suggestions Web-Based Collaborative Procedure*. EGOV05 - International Conference on E-Government, Ed. Trauner Verlag, Linz; I.S.B.N.: 3-85487-830-3. Copenague, Dinamarca; 22 Ago, 2005
- Penichet, 2006f **Penichet, Victor M. R.**; Lozano, Maria D.; Gallud, Jose A.; Montero, Francisco: *Ontología para Estructuras Organizativas Colaborativas*. Proceedings del VII Congreso Internacional de Interacción Persona-Ordenador - Interacción 2006. Universidad de Castilla-La Mancha, Puertollano, Ciudad Real, Spain; 15 Nov 2006.
- Penichet, 2006e **Penichet, Victor M. R.**; Lozano, Maria D.; Gallud, Jose A.: *Describing Group Tasks in Multi-User Systems*. Proceedings of the 4th Latin American Web Congress - La-Web 2006. IEEE Computer Society Press, I.S.B.N.: 0-7695-2693-4. Universidad de las Americas, Puebla, Cholula, Mexico; 26 Oct 2006.
- Penichet, 2006d **Penichet, Victor M. R.**; Calero, C.; Lozano, Maria D.; Piattini, M.: *Using WQM for Classifying Usability Metrics*. Proceedings of the IADIS International Conference WWW/Internet 2006 - ICWI 2006; I.S.B.N.: 972-8924-19-4. Murcia, Spain; 05 Oct 2006.
- Penichet, 2006c **Penichet, Victor M. R.**; Marin, Ismael; Gallud, Jose A.; Lozano, Maria D.; Tesoriero, Ricardo: *A Classification Method for CSCW Systems*. Proceedings of the Second International Workshop on Views On Designing Complex Architectures - VODCA 2006. Bertinoro University Residential Centre, Bertinoro, Italy; Sep 2006.
- Penichet, 2006b **Penichet, Victor M. R.**; Gallud, Jose A.; Lozano, Maria D.; Tobarra, M.: *Web Management of Citizens' Complaints and Suggestions*. Proceedings of the 12th International Workshop on Groupware - CRIWG 2006. Lecture Notes in Computer Science, Springer Verlag; I.S.B.N.: 3-540-39591-1. Valladolid, Spain; 20 Sep 2006.
- Penichet, 2006a **Penichet, Victor M. R.**; Paternò, Fabio; Gallud, Jose A.; Lozano, Maria D.: *Collaborative Social Structures and Task Modelling Integration*. Proceedings DSV-IS 2006. Lecture Notes in Computer Science, Springer Verlag, 67-80; I.S.B.N.: 978-3-540-69553-0. Dublin, Ireland. 27 Jul 2006
- Penichet, 2007e **Penichet, Victor M. R.**; Marin, Ismael; Gallud, Jose A.; Lozano, Maria D.; Tesoriero, R.: *A Classification Method for*

- CSCW Systems*. Journal: Electronic Notes in Theoretical Computer Science, ENTCS. Ed. Elsevier Science Publishers B. V.; The Netherlands - I.S.S.N.: 1571-0661 Vol. 168, pp. 237-247, 2007
- Penichet, 2007d **Penichet, Victor M. R.**; Lozano, Maria D.; Gallud, Jose A.: *An Ontology to Model Collaborative Organizational Structures in CSCW Systems*. Capítulo de libro internacional pendiente de publicar. Seleccionado en Interacción 2006 para su extensión. Ed. Springer. 2007
- Penichet, 2007c **Penichet, Victor M. R.**; Lozano, Maria D.; Gallud, Jose A.; Tesoriero, R.: *CE4WEB: Una Herramienta CASE Colaborativa para el Modelado de Aplicaciones con UML*. XII Jornadas de Ingeniería del Software y Bases de Datos, JISBD'2007 en el marco del II Congreso Español De Informática (CEDI 2007). Ed. Thomson-Paraninfo. Zaragoza, España; Sep, 2007
- Penichet, 2007b **Penichet, Victor M. R.**; Lozano, Maria D.; Gallud, Jose A.; Tesoriero, R.: *Análisis en un Modelo de Procesos CSCW. Organización, Roles e Interacción Persona-Ordenador-Persona*. Interacción 2007 - VIII Congreso Internacional Interacción 2007 en el marco del II Congreso Español De Informática (CEDI 2007). Ed. Thomson-Paraninfo. Zaragoza, España; Sep, 2007
- Penichet, 2007a **Penichet, Victor M. R.**; Lozano, Maria D.; Gallud, Jose A.; Tesoriero, R.: *Task Modelling for Collaborative Systems*. 6th International workshop on Task Models and Diagrams: TAMODIA 2007. Lecture Notes in Computer Science (LNCS), Springer Verlag. Toulouse, France, 2007.
- Pilone, 2005 Pilone, Dan; Pitman, Neil: *UML 2.0 in a Nutshell*. O'Reilly Media; ISBN-13: 978-0596007959 – 2nd edition. 2005
- Pinelle, 2003 Pinelle, D., Gutwin, C., Greenberg, S.: *Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration*. ACM Transactions on Computer-Human Interaction (TOCHI) Volume 10 , Issue 4, Pages: 281 - 311. (2003) ISSN:1073-0516
- Poltrock, 1994 Poltrock, S. and Grudin, J. 1994. *Computer Supported Cooperative Work and Groupware*. In Conference Companion on Human Factors in Computing Systems (Boston, Massachusetts, United States, April 24 - 28, 1994). C. Plaisant, Ed. CHI '94. ACM Press, New York, NY, 355-356
- Poltrock, 1999 Poltrock, S. and Grudin, J. 1999. *CSCW, groupware and workflow: experiences, state of art, and future trends*. In CHI '99 Extended Abstracts on Human Factors in Computing Systems (Pittsburgh, Pennsylvania, May 15 - 20, 1999). CHI '99. ACM Press, New York, NY, 120-121
- Poltrock, 2005 Poltrock, S. and Grudin, J. 2005. *Computer Supported Cooperative Work and Groupware (CSCW)*. In Interact 2005 (Rome, Italy, Ago 1999)
- Poseidon, Web Poseidon UML, <http://www.gentleware.com/>
- Puerta, 1996 Puerta, A.R.: *The Mecano Project: Comprehensive and Integrated Support for Model-Based Interface Development*.

- CADUI'96: Second International Workshop on Computer-Aided Design of User Interfaces, Namur, Belgium, June 1996, pp. 19-25.
- Puerta, 1997 Puerta, A.R.: A Model-Based Interface Development Environment. IEEE Software, pp. 40-47, 1997.
- RAE, 2005 Real Academia Española, www.rae.es
- Raghavan, 1994 Raghavan, S.; Zelesnik, G. y Ford, G.: *Lecture Notes on Requirements Elicitation*. Educational Materials CMU/SEI-94-EM-10, Software Engineering Institute, Carnegie Mellon University, 1994.
- Rational, Web Rational Rose, <http://www-306.ibm.com/software/rational/>
- Roberts, 1998 Roberts, D.; Berry, D.; Isensee, S. y Mullaly, J.: *Designing for the User with OVID: Bridging User Interface Design and Software Engineering*. New Riders Publishing, September 1998
- Roseman, 1992 Roseman, M. and Greenberg, S. (1992). *GroupKit: A groupware toolkit for building real-time conferencing applications*. In Proceedings of the ACM CSCW Conference on Computer Supported Cooperative Work, pp. 43-50, Toronto, Canada, November 1-4, ACM Press
- Rumbaugh, 1999 Rumbaugh, J.; Jacobson, I.; Booch, G.: *The Unified Modeling Language. Reference Manual*. Addison-Wesley. 1999
- Sawyer, 1997 Sawyer, P.; Sommerville, I. y Viller, S.: *Requirements Process Improvement through The Phased Introduction of Good Practice*. Software Process - Improvement and Practice, 3(1), 1997.
- Schümmer, 2001 Schümmer, Till; Schümmer, Jan; Schuckmann, Christian (2001): *COAST An Open Source Framework to Build Synchronous Groupware with Smalltalk*
- Silva, 2000 da Silva, P., Paton, N. W., *UMLi: The Unified Modeling Language for Interactive Applications*, in Conf. Proc. of UML00, UK, p.117-132, 2000
- Silva, 2002 da Silva, P. Pinheiro. *Object Modelling of Interactive Systems: The UMLi Approach*. PhD Thesis. University of Manchester. UK, 2002.
- Stanciulescu, 2005 Stanciulescu, A., Limbourg, Q., Vanderdonckt, J., Michotte, B., Montero, F., *A Transformational Approach for Multimodal Web User Interfaces based on UsiXML*, Proc. of ICMI'2005
- Sтары, 1999 Sтары, Chris: *TADEUS (Task Analysis / Design / End User Systems)*, University of Linz, 1999. <http://www.ce.uni-linz.ac.at>
- Stoiber, 2000 Stoiber, S.; Sтары, C.: *UML-Support for Model- and Task-Based Development of Interactive Software Systems*. In Proceedings of the TUPIS'2000 Workshop at the UML'2000
- Studer, 1998 Studer, R., Benjamins, R., Fensel, D. (1998) *Knowledge Engineering: Principles and Methods*. Data and Knowledge Engineering 25(1-2):161-197.
- Sukaviriya, 1995 Sukaviriya, P.; Muthukumarasamy, J.; Frank, M.; Foley, J.: *A Model-Based User Interface Architecture: Enhancing a*

- Runtime Environment with Declarative Knowledge*. In: F. Paterno (ed.): *Interactive Systems: Design, Specification and Verification*. Berlin: Springer, 1995, 181-197
- Szekely, 1996 Szekely, P. A., Sukaviriya, P. N., Castells, P., Muthukumarasamy, J., and Salcher, E. 1996. *Declarative interface models for user interface construction tools: the MASTERMIND approach*. In *Proceedings of the IFIP Tc2/Wg2.7 Working Conference on Engineering For Human-Computer interaction* L. J. Bass and C. Unger, Eds. IFIP Conference Proceedings, vol. 45. Chapman & Hall Ltd., London, UK, 120-150
- Tam, 2006 Tam, J., and Greenberg, S. (2006): *A Framework for Asynchronous Change Awareness in Collaborative Documents and Workspaces*. *International Journal of Human Computer Studies*, 64(7), p583-598, Elsevier.
- Tantek, 2005 Tantek, Çelik. *Microformats*. <http://tanket.com>
<http://microformats.org>
- Target, 1997 Target Internet Editores (1997): *Workflow, Groupware and the role of Ultimus*
<http://www.target.com.co/workflow/Que%20es%20Workflow/tiposdeg.htm>
- Taylor, 1959 Taylor, E.S.: *An Interim Report on Engineering Design*, Massachusetts Institute of Technology, 1959
- Tesoriero, 2007 Tesoriero, R.; Lozano, Maria D.; Gallud, Jose A.; **Penichet, Victor M. R.**. *Evaluating the Users' Experience of a PDA-Based Software Applied in Art Museums*. *Proceedings del 3rd International Conference on Web Information Systems and Technologies - WEBIST 2007*. Universidad de Cataluña, Barcelona, Spain; 06 Mar 2007.
- Tollinger, 2004 Tollinger, Irene; McCurdy, Michael; Vera, Alonso H.; Tollinger, Preston: *Collaborative knowledge management supporting mars mission scientists*. *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004. p 29-38
- Traetteberg, 1999 Traetteberg, H. *Modeling work: Workflow and Task Modeling*. CADUI'99. 275-280
- Tse, 2004 Tse, E. and Greenberg, S. (2004) *Rapidly Prototyping Single Display Groupware through the SDGToolkit*. *Proc Fifth Australasian User Interface Conference*, Volume 28 in the CRPIT Conferences in Research and Practice in Information Technology Series, (Dunedin, NZ January), Australian Computer Society Inc., p101-110
- Udell, 2004 Udell, Jon: *Collaborative knowledge gardening*. InfoWorld. August 20, 2004.
http://www.infoworld.com/article/04/08/20/34OPstrategic_1.html
- Vanderdonckt , 1993 Vanderdonckt, J.; Bodart, F.: *Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection*, in *ACM Proc. of the Conf. on Human Factors in Computing Systems INTERCHI'93* (Amsterdam, 24-29 April 1993), S. Ashlund, K.

- Mullet, A. Henderson, E. Hollnagel & T. White (eds.), ACM Press, New York, 1993, pp. 424-429
- Vanderdonckt , 2005 Vanderdonckt, Jean: *A MDA-Compliant Environment for Developing User Interfaces of Information Systems*, pp. 16-31, In: Proceedings of the 16th Conference on Advanced Information Systems Engineering, Oscar Pastor, João Falcão e Cunha (Ed.), Lecture Notes in Computer Science, Springer-Verlag, Porto, Portugal, Lecture Notes in Computer Science, Vol. 3520, June 2005, ISBN 3-540-26095-1
- Vander Wal, 2004 Vander Wal, Thomas. *Off the Top: Folksonomy*. <http://www.vanderwal.net/random/category.php?cat=153>. Año 2004.
- Van der Veer, 2000 Van der Veer, G. C.; Van Welie, M. 2000. *Task based groupware design: Putting theory into practice*. In Proceedings of the 2000 Symposium on Designing Interactive Systems. New York, ACM Press, 326-337
- Van Dyke Parunak, 2001 Van Dyke Parunak, H. and Odell, J. 2001. *Representing social structures in UML*. In Proceedings of the Fifth international Conference on Autonomous Agents. AGENTS '01. ACM Press, New York, NY, 100-101
- Wang, 2001 Wang, Alf Inge; Conradi, Reidar; Liu, Chunnian (2001): *A Multi-Agent Architecture for Cooperative Software Engineering*. Dept. of Computer and Information Technology Norwegian University of Science and Technology
- WfMC, 1999 WfMC. Workflow Management Coalition. *Terminology & Glossary (WFMC-TC-1011)*. Technical Report. United Kingdom: Workflow Management Coalition, 1999
- Wikimedia, 2007 Wikipedia: Wikimedia Foundation, Inc, <http://www.wikipedia.org>
- W3C, 1999 W3C consortium, HTML 4.01 specification, W3C Recommendation, 24 Dec. 1999. Available at <http://www.w3.org/TR/REC-html40/>
- W3C, 2006 World Wide Web Consortium (W3C). www.w3c.com
- W3C, 2004 W3C consortium, Voice Extensible Markup Language (VoiceXML) Version 2.0, W3C Recommendation, 16 March 2004 available at <http://www.w3.org/TR/voicexml20/>
- W3C-OWL, 2004 W3C, *OWL Web Ontology Language Guide*. W3C Recommendation 10 February 2004. Editors: Michael K. Smith, Electronic Data Systems; Chris Welty, IBM Research; Deborah L. McGuinness, Stanford University. 2004.
- Yahoo-Flickr, 2005 Yahoo!, Inc. *Flickr*. <http://flickr.com>
- Yahoo-del, 2005 Yahoo!, Inc. *del.icio.us*. <http://del.icio.us>
- Zhao, 2001 Zhao, G.; Deng, J.: *Cooperative Design Process Modeling*. CSCWD2001. 2001

ANEXO I: DESCRIPCIÓN DE LOS METADATOS DE LAS PLANTILLAS PARA ELICITACIÓN DE REQUISITOS [DURÁN, 2000]

Para facilitar el trabajo con la metodología se ha introducido en este anexo la información relativa a los metadatos aportados por [Durán, 2000] en las plantillas para la elicitación de requisitos. Esos metadatos se utilizan en algunas de las plantillas extendidas, modificadas o creadas para esta tesis (Capítulo 5).

Los metadatos aportados en este trabajo se han descrito en dicho capítulo, por lo que en este anexo sólo se han incluido aquellos que son originales de [Durán, 2000].

En la notación usada para describir los patrones-L (patrones lingüísticos), las palabras o frases entre < y > deben ser convenientemente reemplazadas, mientras que las palabras o frases que se encuentren entre { y } y separadas por comas representan opciones de las que se debe escoger una.

Nota: Adicionalmente se ha añadido el símbolo # delante de un identificador que aparezca en un patrón-L si ese identificador está haciendo referencia a un *identificador* válido y existente de otro elemento descrito por una plantilla. Así, se puede consultar la plantilla correspondiente para tener más información.

Los metadatos se estructuran de acuerdo a las plantillas empleadas en el Capítulo 5 de la tesis:

Plantilla general para requisitos

Nota: definido para *objetivos*, pero válido igualmente para *requisitos*.

- **Identificador y nombre descriptivo:** siguiendo la propuesta, entre otros, de [Sawyer, 1997], cada objetivo debe identificarse por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los objetivos comienzan con OBJ.

Nota: En esta metodología se utilizan las etiquetas *O- $\langle id \rangle$* , *RI- $\langle id \rangle$* , *RF- $\langle id \rangle$* , *RNF- $\langle id \rangle$* sobre la misma plantilla (sólo cambia la primera). El identificador será *O- $\langle id \rangle$* si identifica un objetivo del sistema, *RI- $\langle id \rangle$* si se trata de un requisito de información, *RF- $\langle id \rangle$* si es un requisito funcional, o bien *RNF- $\langle id \rangle$* si hace referencia a un requisito no funcional.

- **Versión:** para poder gestionar distintas versiones, este campo contiene el número y la fecha de la versión actual del objetivo.
- **Autores, Fuentes:** estos campos contienen el nombre y la organización de los autores (normalmente desarrolladores) y de las fuentes (clientes o usuarios), de la versión actual del objetivo, de forma que la rastreabilidad pueda llegar hasta las personas que propusieron la necesidad del requisito.
- **Importancia:** este campo indica la importancia del cumplimiento del objetivo para los clientes y usuarios. Se puede asignar un valor numérico o alguna expresión enumerada como vital, importante o quedaría bien, tal como se propone en [OOTC, 1996]. En el caso de que no se haya establecido aún la importancia, se puede indicar que está por determinar (PD), equivalente al TBD (To Be Determined) empleado en las especificaciones escritas en inglés.
- **Urgencia:** este campo indica la urgencia del cumplimiento del objetivo para los clientes y usuarios en el supuesto caso de un desarrollo incremental. Como en el caso anterior, se puede asignar un valor numérico o una expresión enumerada como inmediatamente, hay presión o puede esperar [OOTC, 1996], o PD en el caso de que aún no se haya determinado.
- **Estado:** este campo indica el estado del objetivo desde el punto de vista de su desarrollo. El objetivo puede estar en construcción si se está elaborando, pendiente de negociación si tiene algún conflicto asociado pendiente de solución, pendiente de validación si no tiene ningún conflicto pendiente y está a la espera de validación o, por último, puede estar validado si ha sido validado por clientes y usuarios.
- **Estabilidad:** este campo indica la estabilidad del objetivo, es decir una estimación de la probabilidad de que pueda sufrir cambios en el futuro. Esta estabilidad puede indicarse mediante un valor numérico o mediante una expresión enumerada como alta, media o baja o PD en el caso de que aún no se haya determinado.

La información sobre la estabilidad, bien a nivel de objetivos, bien a nivel de requisitos, ayuda a los diseñadores a diseñar software que prevea de antemano la necesidad de posibles cambios futuros en aquellos aspectos relacionados con los elementos identificados como inestables durante la fase de ingeniería de requisitos, favoreciendo así el mantenimiento y la evolución del software [Brackett, 1990].

- **Comentarios:** cualquier otra información sobre el objetivo que no encaje en los campos anteriores puede recogerse en este apartado.

Extensión específica para requisitos de información

- **Descripción:** para los requisitos de almacenamiento de información este campo usa un patrón-L que se debe completar con el concepto relevante sobre el que se debe almacenar información.
- **Datos específicos:** este campo contiene una lista de los datos específicos asociados al concepto relevante, de los que pueden indicarse todos aquellos

aspectos que se considere oportunos (descripción, restricciones, ejemplos, etc.).

- **Intervalo temporal:** este campo indica durante cuánto tiempo es relevante la información para el sistema. Puede tomar los valores pasado y presente, si la información es siempre relevante, o sólo presente si la información tiene un periodo de validez concreto.

Por ejemplo, si el concepto es empleados, y el intervalo de tiempo es pasado y presente, quiere decir que los ex-empleados son relevantes para el sistema, mientras que un periodo de tiempo de sólo presente indicaría que los ex-empleados no se deben considerar.

Un intervalo temporal de pasado y presente suele implicar considerar la necesidad de dispositivos de almacenamiento con grandes capacidades o la necesidad de algún tipo de archivos históricos.

Extensión específica para requisitos funcionales

- **Descripción,** para los requisitos funcionales, este campo contiene un patrón-L que debe completarse de forma distinta en función de que el caso de uso sea abstracto o concreto. Si el caso de uso es abstracto, deben indicarse los casos de uso en los que se debe realizar, es decir, aquellos desde los que es incluido o a los que extiende. Si, por el contrario, se trata de un caso de uso concreto, se debe indicar el evento de activación que provoca su realización. En versiones anteriores de este patrón-L, aparecían las expresiones caso de uso abstracto y caso de uso concreto. La experiencia durante la utilización de estas plantillas en los tres proyectos reales en los que se ha utilizado nos ha llevado a eliminar dichas expresiones, que resultaban difíciles de entender por los participantes en el proceso de elicitación.
- **Precondición,** en este campo se expresan en lenguaje natural las condiciones necesarias para que se pueda realizar el caso de uso.
- **Secuencia Normal,** este campo contiene la secuencia normal de interacciones del caso de uso. En cada paso, un actor o el sistema realiza una o más acciones, o se realiza (se incluye) otro caso de uso. Un paso puede tener una condición de realización, en cuyo caso si se realizara otro caso de uso se tendría una relación de extensión. Se asume que, después de realizar el último paso, el caso de uso termina.

Otras propuestas similares, por ejemplo [Coleman, 1998], proponen utilizar estructuras similares al pseudocódigo para expresar las interacciones de los casos de uso. En nuestra opinión, esto puede llevar a que dichas descripciones sean excesivamente complejas de entender para los participantes sin conocimientos de programación y se corre el peligro de especificar los casos de uso con un estilo cercano a la programación.

Para representar estructuras condicionales complejas se puede recurrir a añadir información aparte, por ejemplo una tabla de decisión, y referenciarla desde el paso o los pasos oportunos.

En el caso de estructuras iterativas, su uso puede evitarse con un uso cuidadoso del lenguaje natural. Por ejemplo, para indicar que se procesan todos los artículos de un pedido se puede optar por frases como "el sistema procesa todos los artículos del pedido introducidos por el usuario", en lugar de estructuras como:

REPETIR
procesar artículo del pedido introducido por el usuario
HASTA que no haya más artículos

- **Postcondición**, en este campo se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso.
- **Excepciones**, este campo especifica el comportamiento del sistema en el caso de que se produzca alguna situación excepcional durante la realización de un paso determinado.

Después de realizar las acciones o el caso de uso asociados a la excepción (una extensión), el caso de uso puede continuar la secuencia normal o terminar, lo que suele ir acompañado por una cancelación de todas las acciones realizadas en el caso de uso dejando al sistema en el mismo estado que antes de comenzar el caso de uso, asumiendo una semántica transaccional.

Inicialmente, la expresión utilizada para indicar una terminación anormal del caso de uso como resultado de una excepción era "este caso de uso aborta". La experiencia durante su aplicación nos llevó a la conclusión de que el término abortar resultaba emocionalmente molesto para algunos participantes [Goleman, 1996], por lo que se cambió por "este caso de uso termina" con el significado comentado anteriormente.

- **Rendimiento**, en este campo puede especificarse el tiempo máximo para cada paso en el que el sistema realice un acción.
- **Frecuencia esperada**, en este campo se indica la frecuencia esperada de realización del caso de uso, que aunque no es realmente un requisito, es una información interesante para los desarrolladores.

Extensión específica para requisitos no funcionales

- **Descripción**, se usa un patrón-L que debe completarse con la capacidad que deberá presentar el sistema.

Extensión CSCW de la plantilla general para requisitos

Nota: No presenta campos originarios de [Durán, 2000].

Plantilla para la Estructura Organizativa

- **Versión, Autores, Fuentes, Comentarios**, se definen de forma análoga a como se han hecho con anterioridad.

Plantilla para Actores

- **Versión, Autores, Fuentes, Comentarios**, se definen de forma análoga a como se han hecho con anterioridad.

Extensión específica para Grupos

Nota: No presenta campos originarios de [Durán, 2000].

Extensión específica para Individuos, Usuarios y Agentes

Nota: No presenta campos originarios de [Durán, 2000].

ANEXO 2: ACTUACIONES EXPERIMENTALES

Como se ha visto en el Capítulo 2 el número de aplicaciones groupware posibles es muy grande, y con el objetivo de conocer cómo funcionan estas herramientas se han desarrollado e ideado algunas en campos de aplicación como e-Administración o gestión de proyectos que se describen brevemente en este anexo.

El desarrollo de herramientas de estas características puede ser útil para conocer las necesidades reales antes de tener la metodología, para aplicar la metodología y obtener información útil de cara a mejorarla y para comparar los sistemas elaborados por medio de la metodología con los que han sido elaborados sin seguir unos determinados pasos.

En el proyecto del *Procedimiento de Sugerencias y Reclamaciones* se puede comprobar aspectos relativos al modelado de tareas con CTT, roles, notificaciones, comportamiento de los diferentes usuarios que interaccionan, etc.

El proyecto de la *Bitácora de Proyectos* se puede emplear para tener información y comprobar asuntos relativos a CTT, gestión de información, roles, colaboración, etc.

Procedimiento de Sugerencias y Reclamaciones

La Administración Pública trata continuamente de mejorar la calidad de sus servicios y para ello necesita sistemas informatizados que resuelvan sus necesidades. Uno de los procedimientos administrativos más demandados es el de *Sugerencias y Reclamaciones* que permite a los trabajadores de la Administración tener una retroalimentación, una información adicional desde los ciudadanos. Este proyecto se encuentra actualmente en ejecución y con la previsión de ser ampliado incluyendo unos agentes que se comentarán posteriormente.

La Figura 147 muestra una captura del sistema funcionando desde la Web Municipal en la dirección <http://www2.albacete.es/sqs>.

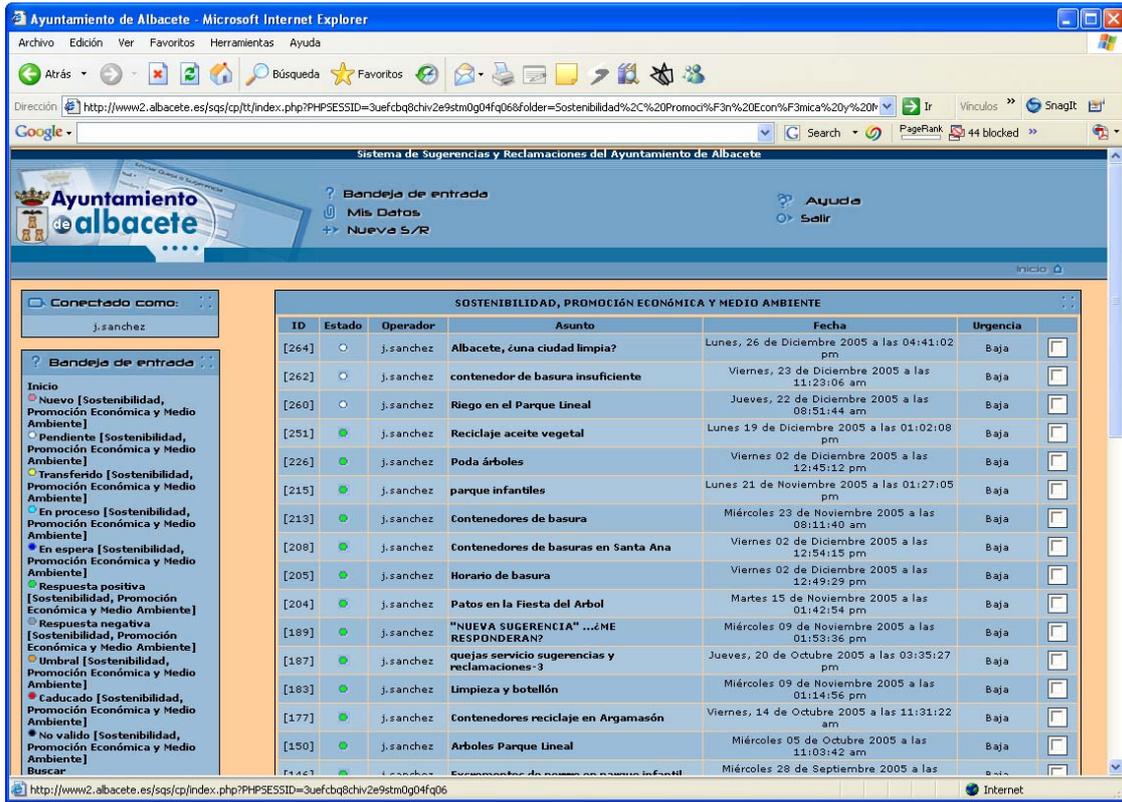


Figura 147. Captura del Procedimiento de Sugerencias y Reclamaciones en funcionamiento en la Web Municipal

Se trata de un proyecto de coordinación por medio de un workflow que indica lo que se ha de hacer en cada momento de forma automática. El proceso va realizando las notificaciones que estima oportunas tanto a los ciudadanos como a los administrativos, por medio de correos electrónicos y por medio de la intranet y la Internet, gestiona los siguientes pasos a realizar, controla tiempos, decide qué está bien y qué no semánticamente, clasifica la información y la redirecciona, archiva los comentarios, etc.

En la Figura 148 se puede observar el flujo completo del sistema.

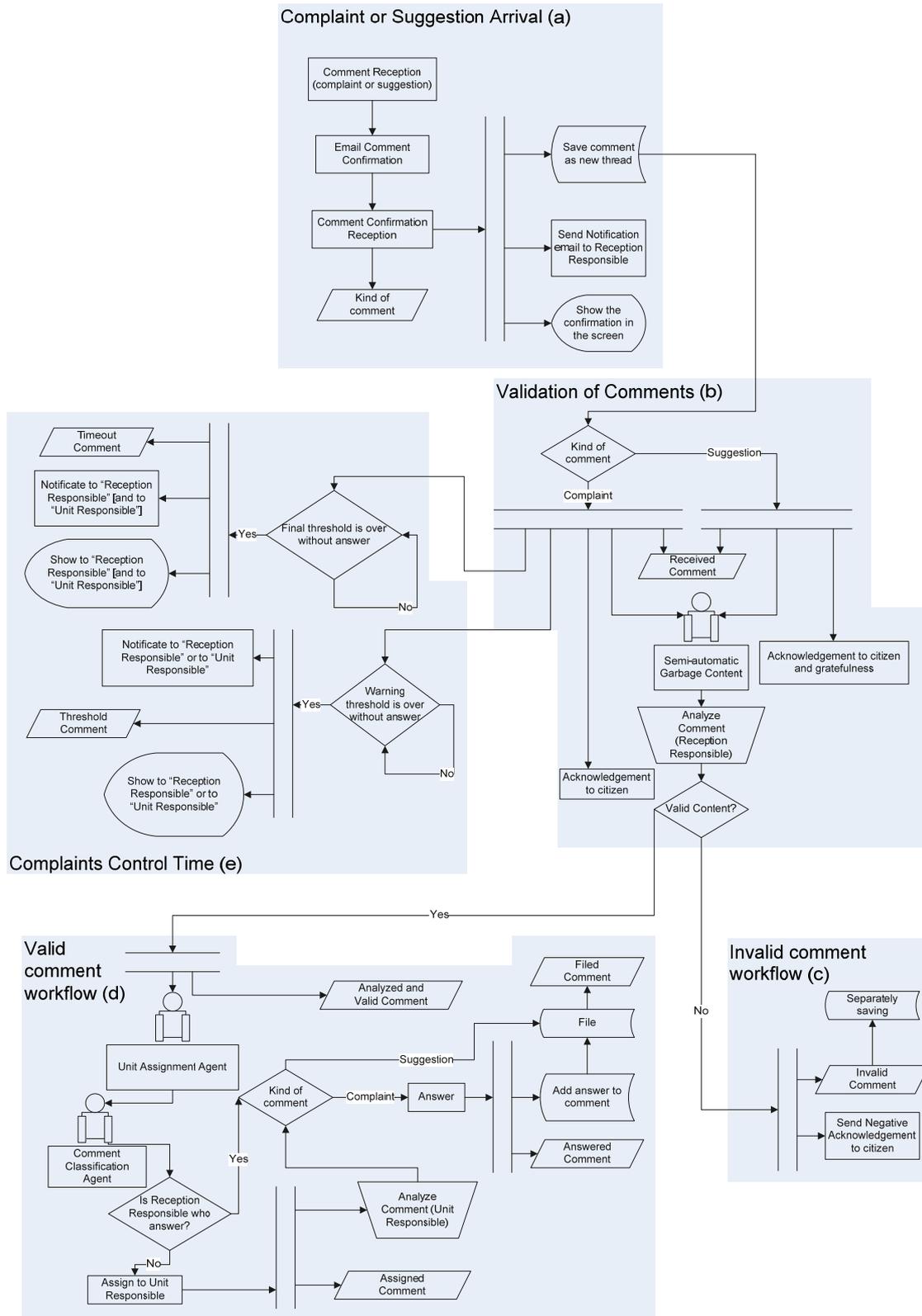


Figura 148. Flujo completo del Procedimiento de Sugerencias y Reclamaciones

Roles del sistema

Como en todos los sistemas, el uso del mismo depende del rol que tenga el usuario que está utilizándolo en ese momento. La identificación de usuarios se hará teniendo en cuenta una autenticación (por medio de nombre de usuario y contraseña desde la pantalla principal) para el caso de todos los usuarios del sistema excepto para el ciudadano que no necesitará un registro previo para facilitarle su acceso (en esta primera versión de la herramienta).

El usuario con rol de "Administrador del sistema" podrá gestionar los usuarios (modificar, crear y eliminar, así como asignarle los roles).

Los roles son restrictivos por autenticación y permisivos por derechos, de manera que si un usuario tiene rol de "Ciudadano" y de "Responsable de Recepción", deberá registrarse en el sistema pero tendrá toda la funcionalidad del "Responsable de Recepción".

Un usuario con el rol de "Ciudadano" no necesita autenticación en el sistema (no tienen que introducir usuario y contraseña para facilitar su acceso). Sólo podrá enviar las sugerencias y reclamaciones así como recibir la (-s) respuestas oportunas.

Un "Administrador del Sistema" puede ejecutar su funcionalidad más la del "Responsable de Recepción", necesita autenticación en el sistema. Puede gestionar los usuarios (modificar, crear y eliminar, así como asignarle los roles).

Un "Responsable de Recepción" puede ejecutar su funcionalidad más la del "Responsable de Unidad", necesita autenticación en el sistema. Puede recepcionar directamente los comentarios (sean quejas o sugerencias) y responderlos directamente o redireccionarlos a un "Responsable de Unidad".

Un "Responsable de Unidad" puede ejecutar su funcionalidad más la del "Ciudadano", necesita autenticación en el sistema. En principio no puede recepcionar directamente los comentarios. Los recepciona indirectamente pues se los envía el "Responsable de Recepción" y los responde.

Descripción del Workflow

El ciudadano que quiera enviar una queja o una sugerencia al sistema de Sugerencias y Reclamaciones debe hacerlo accediendo al formulario de la Web que está habilitado a tal efecto.

Dicho formulario tiene una serie de campos a rellenar por el ciudadano, algunos de ellos obligatorios. En concreto los obligatorios son el texto del comentario (lógicamente), si se trata de una queja o de una sugerencia, y el correo electrónico del ciudadano para poder establecer un feedback (notificaciones). En todo momento, el ciudadano estará informado de lo que pasa, y de lo que pasará posteriormente por medio de su email.

Al ciudadano le llegará un correo electrónico inmediatamente después de que se haya recibido el comentario para que confirme su queja o sugerencia. Entonces comienza el proceso en sí.

El sistema está basado en una aplicación Web de servidor. La lógica de negocio se lleva a cabo en el servidor, quedando constancia persistentemente de la información que va surgiendo. Por tanto, el correo electrónico que se envía como notificación al "Responsable de Recepción" es sólo una notificación porque el comentario se almacena en el sistema de forma persistente.

Si el comentario es una *sugerencia*, previamente se ha seleccionado como tal en el formulario de envío por parte del ciudadano, entonces tan sólo se envía un "Acusar Recibo al ciudadano y Agradecimiento". El proceso finaliza. En caso contrario, el comentario es una *queja*, y seguirá otros pasos diferentes: se realiza un acuse de recibo al correo electrónico del ciudadano para que tenga conocimiento de que se ha recibido.

Se establece un umbral condicionado por el tiempo máximo permitido para responder el comentario. Por ejemplo, si el tiempo máximo permitido para responder el comentario son diez días, el umbral, que lógicamente será siempre menor, podría ser ocho. De esta manera, se puede controlar si se está cerca de sobrepasar el tiempo máximo. Es de utilidad de cara a avisar al responsable de realizar la respuesta al comentario en caso de que esté cerca su caducidad, y evitar en la medida de lo posible que se pase del tiempo comprometido con el ciudadano. Si el tiempo umbral de aviso permitido para responder el comentario se ha sobrepasado, se envía un correo electrónico al "Responsable de Recepción" si no se ha redireccionado o al "Responsable de Unidad" al que se ha redireccionado. En definitiva, se envía un correo electrónico a la persona que debe contestar al comentario.

Del mismo modo, se establece un tiempo máximo permitido para responder el comentario, por ejemplo 10 días, tiempo del que se informa al ciudadano para darle confianza en el sistema. De esta manera, se puede controlar si se sobrepasa el tiempo máximo. Es de utilidad para avisar al "Responsable de Recepción" en caso de que un comentario que pase del tiempo comprometido con el ciudadano no haya sido respondido.

El "Responsable de Recepción" lee el comentario, lo analiza "manualmente". Tras analizar el comentario, el "Responsable de Recepción" decide si tiene un contenido válido (sin comentarios desmedidos, que puedan herir la sensibilidad o contengan contenidos degradantes, vejatorios, molestos y/o no constructivos, cuyo objetivo no sea expresar la queja o sugerencia para el que ha sido diseñado). Se ha ideado un agente inteligente para la siguiente versión de la herramienta de manera que permita realizar esta identificación de forma automática.

Si el comentario era válido el "Responsable de Recepción" puede responderlo o bien redireccionarlo. Si se trata de una *sugerencia* no se responde sino que se archiva. De nuevo se ha ideado otro agente inteligente para realizar esta redirección automáticamente en caso de que así lo decida el "Responsable de Recepción". Un último agente clasificaría este comentario para tener una idea previa y saber el tema antes de analizarlo finalmente.

En caso de que el "Responsable de Recepción" decida redireccionarlo, en su zona privada donde visualiza todos los comentarios dispondrá de una lista de usuarios a los que se puede redireccionar el comentario. Estos usuarios tienen el rol de "Responsable de Unidad".

El usuario del sistema encargado de responder el comentario (queja sólo) lo hace y se almacena definitivamente el resto de datos: respuesta, notas, etc.

Estados de los Comentarios

Los comentarios van pasando por una serie de estados a lo largo del proceso. De esta manera, el ciudadano puede ver en todo momento en qué estado se encuentra la queja o sugerencia que introdujo en el sistema.

Estos estados no son excluyentes. Por ejemplo, un comentario podría haber sobrepasado el *umbral* y a la vez encontrarse en el estado de *redireccionado*.

Los estados por los que pasa un comentario son los siguientes:

- *Tipo*, el comentario puede ser una "queja" o una "sugerencia".
- *Recibido*, recibido y almacenado, listo para procesarse pero nada más.
- *No válido*, no es válido porque su contenido no lo es.
- *Analizado y válido*, tiene un contenido válido.
- *Timeout*, se ha excedido el tiempo máximo permitido para responder.
- *Umbral*, está cerca del tiempo máximo permitido para responder.
- *Redireccionado*, el "Responsable de Recepción" lo ha redireccionado a un "Responsable de Unidad"
- *Respondido*, se ha concluido con la respuesta al comentario.
- *Archivado*, el comentario se ha archivado

Bitácora de Proyectos

Tras la elaboración del documento de definición de un proyecto cualquiera y su posterior aprobación por todas las partes, el proyecto en sí comienza a desarrollarse. Este desarrollo se basa, o se debería basar precisamente en este documento que contiene todos los fundamentos, todos los requisitos que los clientes quieren para su futuro sistema. Pero en todo proyecto surgen cambios, a veces muy simples y otras no tanto.

Si el proyecto se lleva a cabo como es debido, todos esos cambios se verán reflejados de alguna manera mediante actas de las reuniones donde se han acordado esas modificaciones, o realizando una nueva versión del documento de definición del proyecto con su correspondiente aprobación, etc. Lo cierto es que en no pocas ocasiones, los proyectos son tan dinámicos que los cambios constantes son irremediables y van creciendo poco a poco, es decir, lo que en principio fue un pequeño cambio sin importancia, con el tiempo y diversas iteraciones se ha podido convertir en una modificación importante sin que se haya tomado cuenta de ello.

Antes de la liberación de un proyecto, esto puede ser un gran problema porque lo que el cliente final quiere no es lo que tiene reflejado en el DDP¹. Para evitar cualquier tipo de duda, incoherencia o posibilidad de malos entendidos surge la Bitácora de Proyectos.

La Bitácora de Proyectos consiste en una **aplicación colaborativa basada en la Web** orientada al seguimiento y control de la **información** que gira en torno a un **proyecto**. Facilita algunos esqueletos o guías para ciertos documentos como el DDP o el Informe de Estado.

De esta manera acompaña la elaboración de un proyecto desde su inicio hasta su retirada, almacenando la información, **toda la información**, que gira en torno a él y que con frecuencia no se recoge, se pierde y se convierte en un problema de difícil solución.

El sistema es compatible con cualquier metodología de desarrollo que se siga, porque lo que realmente facilita es la documentación global del proyecto,

¹ DDP: Documento de Definición de Proyecto. Un documento inicial que recoge todos los aspectos de la futura solución. Es la base para sentar el lo que hará, lo que no hará, la planificación, los compromisos y el presupuesto.

recogiendo y relacionando datos incluso de las impresiones de los programadores que lo desarrollan.

No está centrado en la información técnica propiamente dicha. Es decir, la información que se generará en la memoria del proyecto y que es el objeto de esta aplicación no son sólo los detalles técnicos sino más bien todas las decisiones que giran en torno al proyecto. Se trata de elaborar un diario del proyecto que permita almacenar, de forma persistente, lo que ocurre para que quede constancia.

La aplicación está actualmente en funcionamiento y es accesible desde la dirección <http://161.67.133.48/bitacoras/default.aspx?alias=bitacora> como una acción piloto del portal Sicman de la Fundación Campollano en Albacete.

Descripción General de Requisitos

Roles y Usuarios del Sistema

Como en todos los sistemas, el uso del mismo depende del rol que tenga el usuario que está utilizándolo en ese momento. La identificación de usuarios se hará teniendo en cuenta una autenticación (por medio de nombre de usuario y contraseña desde la pantalla principal) para el caso de todos los usuarios del sistema excepto para el *Anónimo* que no necesitará un registro previo para facilitarle su acceso a la parte pública del portal Web.

El *Administrador del Sistema* tiene la capacidad de crear nuevos *grupos de usuarios* y *usuarios del sistema*. Cuando se crea un usuario se ha de asociar a uno o varios grupos. En cualquier caso podría no estar asociado a ninguno. El *Administrador del Sistema* puede asignar los roles tanto a usuarios individuales como a grupos de usuarios (aunque, por organización, es altamente recomendable el uso de los grupos). De este modo los usuarios podrán realizar algunas tareas y visualizar cierta información o no según los roles que tengan asignados.

Un usuario que esté en varios grupos y, adicionalmente, tenga asignado otro rol estará en posesión de la suma de todos los roles.

Hay tres clases de usuarios que siempre existen: el *Administrador del Sistema*, los usuarios que se autentican en el sistema por medio de usuario y contraseña y tienen asignados roles que se denominan *Usuario del Sistema* y, por último, cualquier persona que entre al sistema sin autenticarse, el usuario *Anónimo*, que únicamente tiene la posibilidad de ver la parte pública del sistema.

Además los usuarios y los grupos de usuarios se pueden ordenar por *jerarquías*. Por ejemplo, directores en una jerarquía, trabajadores en otra y cliente en otra. De esta manera, los usuarios que pertenezcan a jerarquías superiores, tendrán ciertos privilegios sobre los usuarios situados en jerarquías inferiores.

Los tipos de usuario se describen a continuación:

- **Administrador del Sistema**

El *Administrador del Sistema* es el único usuario, junto al *Anónimo*, que existe en el sistema desde el principio.

Un usuario con este rol sí necesita autenticación en el sistema (por medio de un nombre de usuario y una clave).

Puede gestionar los *grupos de usuarios*, es decir, crearlos y eliminarlos, así como asignar o quitar roles.

Puede gestionar los *usuarios del sistema*, esto es, crearlos, eliminarlos, asociarlos o desasociarlos a *grupos de usuarios*, así como asignar o quitar roles.

Así mismo puede realizar todas las tareas y visualizar toda la información, pues tiene asignados todos los roles por defecto.

Un *Administrador del Sistema* puede asignar un rol de *Administrador del Sistema* a otro usuario. De esta manera habrá un nuevo administrador general con todos los derechos. Sin embargo, siempre ha de haber al menos un usuario administrador, es decir, un usuario *Administrador del Sistema* no se puede eliminar ni quitar ese rol así mismo.

Puede dar de alta nuevos proyectos en los que trabajarán diferentes usuarios. Lógicamente, los roles se han de asignar nuevamente en cada proyecto para adaptarlo a las necesidades y particularidades de cada uno. Es decir, el usuario existe para el sistema, pero los grupos, la asociación de los usuarios del sistema a los grupos y la asignación de roles dependen del proyecto.

Puede generar y administrar las marcas de tiempo (las marcas de tiempo se describen posteriormente).

- Usuario del Sistema

Todos los usuarios que crea el *Administrador del Sistema* son de este tipo y necesitan autenticación en el sistema. Según los roles que tengan asignados (o bien por pertenecer a *grupos de usuarios*, o bien por tenerlos asignados individualmente) podrán realizar unas acciones u otras y ver una información u otra.

- Anónimo

Este usuario no necesita autenticación en el sistema (no tienen que introducir usuario y clave para acceder).

Entra en el sistema pero sólo puede visualizar la parte pública. Realmente todos los usuarios que entran en el sistema lo hacen siendo anónimos hasta que se autentican.

Existen un grupo de roles fijos disponibles en el sistema que permiten hacer algunas tareas y/o visualizar cierta información. Al contrario que los *grupos de usuarios* y los *usuarios del sistema*, los roles no son creados por el usuario *Administrador del Sistema* sino que se encarga de asignarlos únicamente, ya que existen previamente. Mediante la combinación de los diferentes roles se permite una amplia flexibilidad pues los grupos y usuarios se crearán según las necesidades particulares de cada proyecto y del mismo modo se le asignan los roles correspondientes.

A continuación se describen los diferentes roles del sistema:

- Administrador del Sistema

El rol de Administrador del Sistema es un rol especial y coincide con el usuario Administrador del Sistema, por tanto, la descripción de este punto es la misma que la descripción de dicho usuario (ver descripción del usuario Administrador del Sistema).

- Autor

- Lector

Un usuario con el rol de lector podrá ver todo el sistema pero no tendrá acceso a ningún tipo de creación o modificación los documentos o elementos de memoria en general.

En cualquier caso sólo verá los elementos de memoria que estén activos señalados como activos. De esta manera, los usuarios autores del sistema pueden trabajar sobre elementos que sólo ellos verán hasta que, conscientemente, los activen para que puedan leerlos los lectores también.

La Figura 149 muestra un ejemplo de la organización de los usuarios de un sistema ficticio.

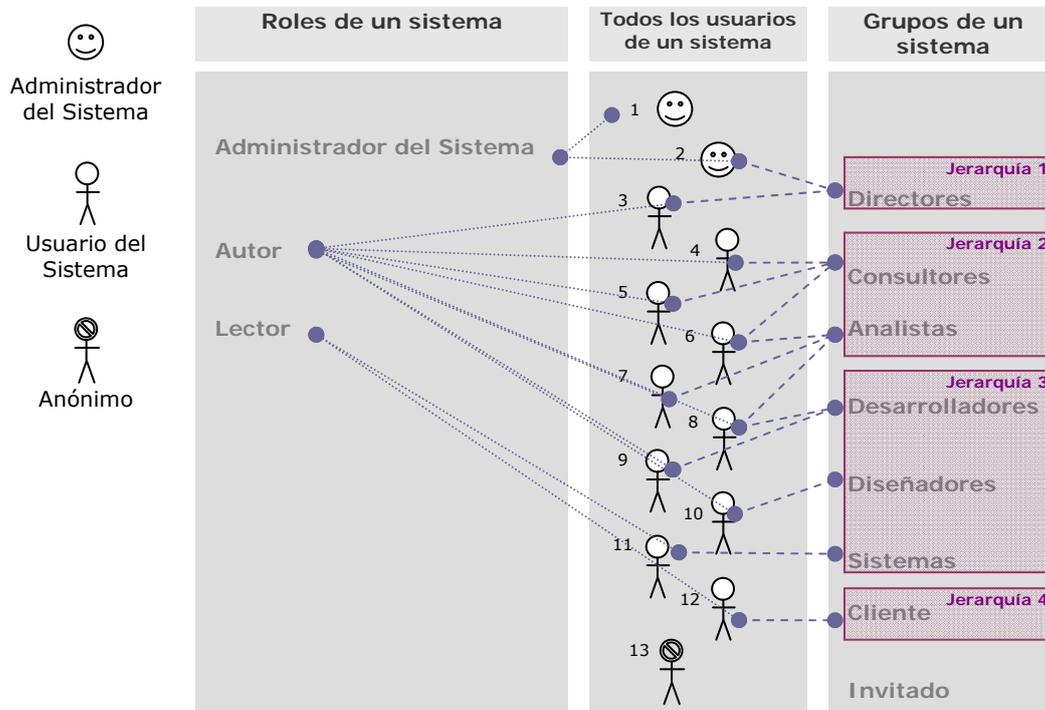


Figura 149. Ejemplo de organización de usuarios para un proyecto concreto

Elementos de memoria

Los elementos de memoria son el centro del sistema. Mediante ellos se almacenan todos los datos que giran en torno a un proyecto independientemente del lugar en el que nos encontremos.

Cada dato, cada pieza de información (comentario, decisión, documento adjunto, texto completo, definición...) se denomina *elemento de memoria*. En todos los *elementos de memoria* en su conjunto forman la bitácora completa del proyecto.

Por ejemplo, si por medio del teléfono el cliente y una de las personas que trabaja en el proyecto acuerdan una pequeña variación en el mismo, ésta se almacenará en la *Bitácora de Proyectos* como un *elemento de memoria*. De esta manera se guarda toda posible variación y tanto los clientes como los ejecutores del proyecto tienen acceso a ver la definición del sistema. Se evitan, por tanto, algunas situaciones que podrían ser tensas puesto que el proyecto está definido en un sitio al que todos los interesados pueden acceder. Está claro cómo es y cómo no es el proyecto y si surgen incidencias o modificaciones quedarán igualmente registradas.

En la Figura 150 se puede observar un prototipo de cómo sería la introducción de un elemento de memoria por parte de un usuario capacitado para ello. A continuación se explican todas las partes de un elemento de memoria y sus características.

- Título del Elemento de memoria
- Identifica rápidamente acerca de qué va el *elemento de memoria* que se va a introducir. Ha de ser breve pero claro.
- Activo
- Un *elemento de memoria* que está desactivado no es visible por usuarios que no tengan asociado el rol de *Autor*. De esta manera los autores del proyecto pueden trabajar un *elemento de memoria* en diferentes sesiones y sólo estará visible para los usuarios con el rol de *lector* (por ejemplo un cliente) cuando considere que es una versión definitiva de dicho *elemento de memoria*.
- Por defecto está inactivo

Bitácora de Proyectos [Salir](#)

Usuario del Sistema >> Nuevo Elemento de Memoria Proyectos - <<Atrás

[P] Nombre del Proyecto DDP - Actas - Informe de Estado -
- Ver Elementos de Memoria - Nuevo Elemento de Memoria -

Título del Elemento de mem: Activo: Sí No

Descripción:

Adjuntar Fichero: [Más ficheros...](#)

Bajo...

¿Dentro?
 Sí No

Relacionado con...
 (Ctrl + clic para seleccionar varios)

Grupos de interés:
 (Ctrl + clic para seleccionar varios)

No visible para:
 (Ctrl + clic para seleccionar varios)

Suscribir a:
 (Ctrl + clic para seleccionar varios)

Tipo:

Día: Mes: Año:

Figura 150. Prototipo de interfaz para la introducción de un nuevo elemento de memoria

- Descripción
- Es el contenido del elemento de memoria, ya sea este un comentario, una aclaración, una modificación, etc. Se trata de texto con la posibilidad de formatearlo de forma sencilla con el propio editor del formulario que permite

texto en negrita, cursiva, subrayado, centrado, justificado, alineado a la izquierda y derecha, inclusión de imágenes, etc.

- Adjuntar Fichero
- Aunque no es recomendable por no complicar el proyecto, se permite adjuntar ficheros en cada *elemento de memoria*.
- Tipo
- Los elementos de memoria tienen un tipo para poder diferenciar unos de otros y enriquecer las posibilidades de la aplicación. No es lo mismo un elemento de memoria de un DDP, uno de un acta, que otro general.
- Es el administrador del sistema el que puede crear tipos de elemento de memoria además de los que ya existen (*DDP, Acta, Incidencia, General*). Estos que ya existen no pueden ser eliminados porque se consideran básicos en el sistema. En cualquier caso, por un uso más sencillo no se recomienda crear más tipos a no ser que sea estrictamente necesario.
- Desde este formulario sólo se pueden etiquetar los elementos de memoria como tipos *Incidencia, General* y los definidos por el administrador. Los tipos *DDP* y *Acta* están aquí deshabilitados pues estos elementos se introducirán de otra manera como se verá posteriormente.
- Bajo...
- Con el fin de ordenar los *elementos de memoria* de forma personalizada, se permite al usuario del sistema incluir un *elemento de memoria* bajo cualquier otro ya existente. Además, se permite anidar elementos de manera que si están muy relacionados puedan pertenecer unos a otros. Esto se consigue mediante la opción "¿Dentro?".
- En el cuadro que muestra los *elementos de memoria* muestra una etiqueta delante de cada uno de ellos indicando su tipo por una mayor claridad. Las etiquetas son de la forma [DDP], [Act], [Inf], [Gen] para mostrar los DDP, actas, incidencias y generales respectivamente. Así mismo, se despliega la lista en el orden definido por el usuario teniendo en cuenta el anidamiento.
- Por defecto el elemento se incluirá detrás del último.
- Relacionado con...
- Los comentarios que se introducen, a menudo están relacionados con otros previos. De esta manera se puede explicitar la relación existente entre un *elemento de memoria* y otros introducidos con anterioridad o posterioridad.
- Por defecto está seleccionado "Ninguno".
- Grupos de interés
- Tanto los grupos de usuarios como los usuarios de forma individual se pueden señalar como personas a las que este *elemento de memoria* nuevo que se va a introducir les interesa. De esta manera se puede acotar la información evitando así "perdersse en la abundancia".
- En el formulario se muestran tanto los grupos (en primer lugar) como los usuarios, por orden jerárquico y alfabético.
- Por ejemplo, se pueden introducir *elementos de memoria* que sólo les interés a los diseñadores, las personas que se encargarán de que el diseño gráfico sea el adecuado. Cuando entren en la bitácora de proyectos, si lo desean pueden ver únicamente los elementos que han sido especialmente señalados para ellos,

abstrayéndose de todo lo demás. En cualquier caso siguen teniendo la opción de ver el resto de elementos para los que tienen acceso.

- Por defecto está seleccionado "Todos".
- No visible para
- Los usuarios de una jerarquía superior pueden decidir si un elemento de memoria es relevante o no para usuarios de una jerarquía inferior.
- En el formulario se muestran tanto los grupos (en primer lugar) como los usuarios, por orden jerárquico y alfabético. No se muestran las jerarquías igual y superiores.
- Por ejemplo, supongamos que en la Figura 150 hay cuatro jerarquías. De mayor a menor nivel en la primera se encuentra el grupo de Directores, en la segunda los grupos de Consultores y Analistas, en la tercera los Desarrolladores, Diseñadores y Administradores de Sistemas y en la cuarta el grupo de Clientes y un usuario llamado "Juan" al que hemos dado acceso como lector para que opine sobre el proyecto. En esta configuración, un usuario del grupo Consultores puede introducir un elemento de memoria que no sea visible por las jerarquías inferiores, es decir, por usuarios del grupo Desarrolladores, Diseñadores y Administradores de Sistemas, Clientes y "Juan". Lo que no puede es hacer un elemento invisible para usuarios de jerarquías iguales o superiores.
- Se ha de tener en cuenta que los documentos generados por las diferentes categorías pueden diferir precisamente por esta restricción. En la generación de los documentos se podrá especificar para quién está orientado el documento que se generará.
- Por defecto está seleccionado "Ninguno: Todos pueden verlo".
- Suscribir a
- Un usuario suscrito recibirá un correo electrónico con la inserción del elemento de memoria y con modificaciones ocurridas posteriormente. Se facilita la información en todo momento.
- En el formulario se muestran tanto los grupos (en primer lugar) como los usuarios, por orden jerárquico y alfabético. No se muestran las jerarquías superiores.
- Se aplican las mismas reglas de jerarquía que en la opción "No visible para".
- Por defecto está seleccionado "Ninguno".
- Fecha
- Por defecto aparece la fecha actual. Independientemente, el sistema guarda en otro campo y automáticamente la fecha real en la que se ha guardado el elemento de memoria (ver *fecha de inserción*).
- Reestablecer Campos
- Restablece los valores iniciales y por defecto en el formulario.
- Guardar
- Guarda el *elemento de memoria* en el sistema.
- Cancelar
- Cancela la inserción del *elemento de memoria* en el sistema.
- Otros datos internos

- Fecha de inserción, identificador único, autor.

Los elementos de memoria se visualizan en una pantalla cuyo prototipo muestra la Figura 151. Dicha visualización será personalizada para cada usuario. Cada usuario sólo ve los elementos a los que tiene acceso y las acciones a las que tiene derecho. Por ejemplo, un usuario con el rol de lector ni siquiera verá el enlace de modificar el elemento, puesto que no tiene los permisos necesarios para ello.

Adicionalmente existen unas opciones de visualización que se almacenan en el perfil del usuario. De esta forma, la próxima vez que acceda al sistema seguirá viendo los elementos tal y como los vio la última vez.

- Usuarios
 - Permite mostrar únicamente los elementos de memoria de los usuarios y/o grupos de usuarios seleccionados.
- Tipo
 - Permite mostrar únicamente los elementos de memoria de los tipos seleccionados.
- Otros
 - Permite mostrar todos los elementos de memoria que el usuario ha marcado como de su interés y/o aquellos a los que se ha suscrito. Conviene recordar que la suscripción y el interés lo pueden forzar usuarios de jerarquías superiores sobre usuarios de jerarquías inferiores.
 - Además se puede elegir el modo en el que se quieren visualizar los elementos. "Sólo títulos" muestra solamente los títulos de los elementos en diferentes líneas. Haciendo clic en el título del elemento de memoria deseado se puede ver su información completa. "Breves" permite ver una pequeña descripción sobre el elemento. Esta descripción se corresponde con las doscientas cincuenta primeras letras de la descripción del elemento seguidas de un enlace para poder ver el resto de información (también haciendo clic en el título). "Completas" es una opción que muestra toda la información. Se recomienda el uso de las dos primeras puesto que al resto de la información se puede acceder siempre.
- Orden
 - Los tres tipos de orden posibles son: "original", que hace referencia al orden que el usuario del sistema ha elegido a la hora de insertar los elementos; "fecha", se refiere a la fecha que ha decidido dar el usuario al elemento en el sistema en su inserción; "autor" organiza los elementos de memoria agrupados por autores que están ordenados alfabéticamente.
 - Se puede seleccionar un orden ascendente o descendente.
- Ver Marcas de Tiempo
 - Para comprender este punto es necesario conocer previamente la información relativa a las *marcas de tiempo* (ver apartado correspondiente).
 - Desde esta opción el usuario podrá configurar qué *elementos de memoria* desea visualizar, según esas marcas temporales, lo que permite acotar en el tiempo la información relativa al desarrollo del proyecto.

Junto al título aparecerán una serie de enlaces para activar o desactivar el elemento, modificarlo o decidir si es un elemento de mi interés o no, o quiero subscribirme a él o darme de baja en la suscripción.

Si el elemento de memoria está siendo modificado por otro autor sólo puede ver en modo lectura y aparecerá un icono de un candado para mostrarlo. Al pasar el puntero por encima de este icono el texto alternativo mostrará quién está modificando el elemento. Mediante este mecanismo de check-in/check-out se impide que haya incoherencias en el sistema.

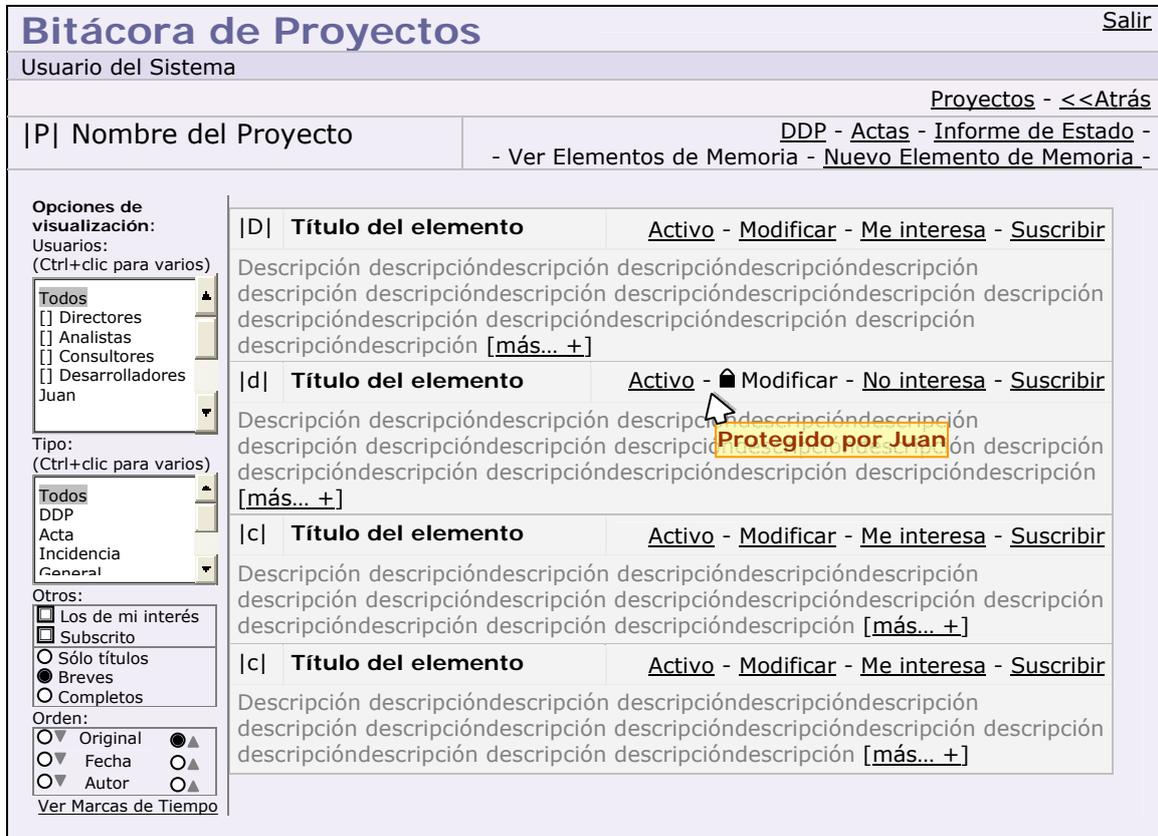


Figura 151. Prototipo de interfaz para visualización de *elementos de memoria*

Interfaz Web

[Descripción breve general e Interfaz por usuarios]

Interfaz Pública

La interfaz pública muestra el nombre de los proyectos que han sido publicados por la empresa de manera que cualquier usuario (*Anónimo*) puede verlos. Haciendo clic en cada uno de ellos se puede ver una serie de datos que los usuarios con rol de *Autor* han introducido. Estos datos son el *nombre del DDP*, *subtítulo* e *imagen* si la tienen, nombre de los autores, un *texto opcional* como pequeña nota o descripción del proyecto, unas *palabras clave* descriptivas si las hay y la *fecha de creación*.

Esta interfaz pública es ligeramente personalizable para cada empresa que la use. El Administrador del Sistema puede:

- Cambiar el color de fondo,
- Cambiar el color de fondo de la barra de navegación,

- Cambiar el color de fondo de la barra de enlaces
- Cambiar el color de las fuentes
- Cambiar el color del fondo y de las fuentes de las cabeceras de las tablas de nivel 1. Los colores del fondo de la cabecera y de los bordes coinciden. Los colores de la fuente de la cabecera y el fondo del resto de la tabla coinciden.
- Cambiar el color del fondo y de las fuentes de las cabeceras de las tablas de nivel 2. Los colores del fondo de la cabecera y de los bordes coinciden. Los colores de la fuente de la cabecera y el fondo del resto de la tabla coinciden.
- Introducir una imagen pequeña como el logotipo de la empresa en la cabecera superior a la izquierda del letrero "Bitácora de Proyectos".
- Introducir una imagen para el centro de la página.
- Modificar el nombre de la empresa y el texto.

Interfaz Privada: Creación del DDP

Sólo los Usuarios del Sistema pueden acceder a la interfaz privada.

Para situar un poco el significado de un DDP según la Real Academia Española:

- **Documento** es un escrito en que constan datos fidedignos o susceptibles de ser empleados como tales para probar algo.
- **Definición** es una proposición que expone con claridad y exactitud los caracteres genéricos y diferenciales de algo material o inmaterial.
- **Proyecto** es una planta y disposición que se forma para la realización de un tratado, o para la ejecución de algo de importancia. O bien, un conjunto de escritos, cálculos y dibujos que se hacen para dar idea de cómo ha de ser y lo que ha de costar una obra de arquitectura o de ingeniería.

Un DDP es un escrito que surge de un análisis y de una serie de reuniones anteriores al comienzo de un proyecto, **recoge toda la información** obtenida hasta el momento, dando idea de la envergadura del mismo. Indica lo que **sí** se va a hacer y lo que **no** se va a hacer. Incluye una **planificación** del proyecto para dar cuenta de lo que puede durar en el tiempo aproximadamente y un **presupuesto** con el coste lo más cercano posible a la realidad.

La Bitácora de Proyectos permite generar un DDP a partir de *elementos de memoria especiales*, que se corresponden precisamente con los diferentes apartados de un DDP. En la Figura 152 se puede observar un pequeño prototipo de la interfaz de generación de un DDP a partir del propio sistema.

Tras la generación del DDP siempre se puede visualizar completo en formato html o PDF. La generación del DDP en formato PDF permite guardar el fichero del DDP en otra máquina o imprimirlo.

El nombre del fichero será del tipo "Nombre_del_Proyecto_150805_1844.pdf", es decir, el nombre del proyecto sin espacios seguido de una fecha y hora para evitar confusiones al descargar diferentes versiones. La fecha y la hora se corresponden con la fecha y hora de los últimos elementos de memoria que están formando este documento. Es decir, si se decide generar un documento en un punto determinado del tiempo pasado, la fecha será esa (ver apartado *Marcas de Tiempo*). En este ejemplo la fecha es el 15 de agosto de 2005 a las 18:44h.

Bitácora de Proyectos		Salir																								
Usuario del Sistema >> DDP																										
		Proyectos - <<Atrás																								
P Nombre del Proyecto	DDP - Actas - Informe de Estado - - Ver Elementos de Memoria - Nuevo Elemento de Memoria -																									
1º Portada - 2º Contactos - 3º Secciones --- Ver PDF - Ver Html																										
<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="3">PORTADA</th> </tr> </thead> <tbody> <tr> <td>Nombre del DDP:</td> <td>Nombre del Proyecto</td> <td>Activo: <input type="radio"/> Sí <input checked="" type="radio"/> No</td> </tr> <tr> <td>Subtítulo:</td> <td><input type="text"/></td> <td>Fecha de Creación *:</td> </tr> <tr> <td>Imagen Portada:</td> <td><input type="text"/> <input type="button" value="Examinar..."/></td> <td>Día <input type="text" value="13"/></td> </tr> <tr> <td>Autor *:</td> <td><input type="text"/></td> <td>Mes <input type="text" value="Agosto"/></td> </tr> <tr> <td>Texto:</td> <td><input type="text"/></td> <td>Año <input type="text" value="2005"/></td> </tr> <tr> <td>Palabras clave:</td> <td><input type="text"/></td> <td>Separadas por ;</td> </tr> <tr> <td colspan="2"></td> <td style="text-align: right;"><input type="button" value="Guardar"/> <input type="button" value="Guardar"/></td> </tr> </tbody> </table>			PORTADA			Nombre del DDP:	Nombre del Proyecto	Activo: <input type="radio"/> Sí <input checked="" type="radio"/> No	Subtítulo:	<input type="text"/>	Fecha de Creación *:	Imagen Portada:	<input type="text"/> <input type="button" value="Examinar..."/>	Día <input type="text" value="13"/>	Autor *:	<input type="text"/>	Mes <input type="text" value="Agosto"/>	Texto:	<input type="text"/>	Año <input type="text" value="2005"/>	Palabras clave:	<input type="text"/>	Separadas por ;			<input type="button" value="Guardar"/> <input type="button" value="Guardar"/>
PORTADA																										
Nombre del DDP:	Nombre del Proyecto	Activo: <input type="radio"/> Sí <input checked="" type="radio"/> No																								
Subtítulo:	<input type="text"/>	Fecha de Creación *:																								
Imagen Portada:	<input type="text"/> <input type="button" value="Examinar..."/>	Día <input type="text" value="13"/>																								
Autor *:	<input type="text"/>	Mes <input type="text" value="Agosto"/>																								
Texto:	<input type="text"/>	Año <input type="text" value="2005"/>																								
Palabras clave:	<input type="text"/>	Separadas por ;																								
		<input type="button" value="Guardar"/> <input type="button" value="Guardar"/>																								
* Campos obligatorios																										

Figura 152. Prototipo de interfaz para la generación de DDPs: Portada del DDP

El DDP tendrá una portada (Figura 152), un apartado para los contactos y uno para las diferentes secciones. En la portada son necesarios los siguientes datos:

- *Nombre del DDP* es un dato que no es necesario introducir puesto que ya lo ha introducido el *Administrador del Sistema* cuando dio de alta el proyecto.
- *Subtítulo opcional* es un texto que se puede introducir y aparecerá bajo el título principal (*Nombre del DDP*) en un tamaño de letra inferior. Opcional.
- *Imagen Portada opcional*, imagen que se puede introducir en la portada.
- *Autor*: nombre del autor o autores del documento.
- *Texto opcional*, por si se quiere introducir alguna nota o una pequeña descripción.
- *Palabras clave* es una serie de palabras separadas por `;` que describen el proyecto, y que facilitará búsquedas posteriores.
- *Fecha de creación* del proyecto, por defecto la fecha actual.
- *Activo* permite que se vea o no en la página Web. Si el DDP está activo, los usuarios con rol de lector podrán visualizar el documento. Así mismo, el proyecto aparecerá en la página inicial de la Bitácora de Proyectos de la empresa.

En la interfaz correspondiente a la inserción de los contactos se introducen las personas que estarán relacionadas con el proyecto y que merece la pena incluir para que quede constancia y puedan ser contactadas en un determinado momento.

En primer lugar se selecciona en número de entidades relacionadas con el proyecto, desplegándose automáticamente los formularios correspondientes. Se pueden introducir hasta tres contactos por entidad.

Al menos existirá una entidad que es la propia empresa, en la que lógicamente no será necesario introducir el nombre.

Por último, las secciones de un DDP se introducirán como un *elemento de memoria* de tipo DDP (ver apartado *Elementos de Memoria*), como se puede apreciar en la Figura 153.

A diferencia del *elemento de memoria* de tipo *Otro* o uno definido por el usuario, el apartado "Bajo..." sólo muestra los *elementos de memoria* que son DDPs y no otros. Esto es así para poder generar el documento correspondiente sin mezclarse. Los *elementos de memoria* aparecen marcados con su tipo en los formularios para poder identificarlos fácilmente.

Pongamos un ejemplo:

Se quiere introducir una sección que explique el *Usuario del Sistema* en un documento como éste. El título del elemento de memoria podría ser "Usuario del Sistema", la descripción es el contenido de la sección, no se adjuntaría ningún fichero anexo, se colocaría "Bajo..." el *elemento de memoria* "Administrador del Sistema". "¿Dentro?" estaría marcado como "No", puesto que sólo va debajo de "Administrador del Sistema" y no es una subsección suya.

Bitácora de Proyectos
[Salir](#)

Usuario del Sistema >> DDP

Proyectos - <<Atrás

|P| Nombre del Proyecto
DDP - [Actas](#) - [Informe de Estado](#) -
- [Ver Elementos de Memoria](#) - [Nuevo Elemento de Memoria](#) -

1º Portada - 2º Contactos - 3º Secciones --- [Ver PDF](#) - [Ver Html](#)

SECCIONES

Introducción

Título del Elemento de mem:

Descripción:

[Pequeño editor (para incluir imágenes, formatear texto...)]

Activo: Sí No

Adjuntar Fichero: [Más ficheros...](#)

Bajo...

[DDP] Introducción
[DDP] Descripción General de Requisitos
-[DDP] Roles y Usuarios del Sistema
--[DDP] Administrador del Sistema
--[DDP] Anónimo

¿Dentro?
 Sí No

Relacionado con...

(Ctrl + clic para seleccionar varios)

Ninguno
[DDP] Introducción
[DDP] Descripción General de Requisitos
-[DDP] Roles y Usuarios del Sistema
--[DDP] Administrador del Sistema

Grupos de interés:

(Ctrl + clic para seleccionar varios)

Todos
[] Directores
[] Analistas
[] Consultores
[] Desarrolladores
Juan

Visible para:

(Ctrl + clic para seleccionar varios)

Ninguno: Todos pueden verlo
[] Directores
[] Analistas
[] Consultores
[] Desarrolladores
Juan

Suscribir a:

(Ctrl + clic para seleccionar varios)

Ninguno
[] Directores
[] Analistas
[] Consultores
[] Desarrolladores
Juan

Tipo:

Día Mes Año

[+ Otra Sección](#)

Figura 153. Prototipo de interfaz para la generación de DDPs: Secciones del DDP

Interfaz Privada: Creación de Actas

Según el diccionario de la Real Academia Española, un acta es una relación escrita de lo sucedido, tratado o acordado en una junta. Certificación, testimonio, asiento o constancia oficial de un hecho.

Es un documento interno de una entidad que resume la información que fluye de una reunión.

Como los DDPs, las actas se pueden generar a partir de elementos de memoria especiales.

Tras la generación de las actas, éstas se pueden visualizar en formato html o PDF. El nombre del fichero pdf será del tipo "Acta_Título del acta_150805_1844.pdf", es decir, la palabra clave acta seguida de *Título del acta* y de una fecha y hora. La fecha y la hora se corresponden con la fecha y hora de los últimos elementos de memoria que están formando este documento. Es decir, si se decide generar un documento en un punto determinado del tiempo pasado, la fecha será esa (ver apartado *Marcas de Tiempo*). En este ejemplo la fecha es el 15 de agosto de 2005 a las 18:44h.

Para cada acta se crearán algunos apartados especiales, que no son elementos de memoria, tal y como se hizo con la *portada* y los *contactos* del DDP. Estos apartados son: *Cabecera*, *Participantes*, *Observaciones*, *Conclusión* y *Liberación* son partes del acta que no se forman a partir de elementos de memoria.

El apartado *Cabecera* contiene datos fáciles de leer, rápidamente detectables para hacer una búsqueda más sencilla. Los datos son los siguientes:

- *Nombre de la empresa* desde la que se realizan las actas.
- *Título del acta* que la identifique claramente. Este campo se utilizará para generar los archivos pdf. Es obligatorio.
- *Ciudad* (y *país* en su caso) donde se realiza la reunión.
- *Grupo o departamento* de la empresa que se ha encargado de realizarla.
- *Fecha* con día, mes y año.
- *Logotipo* correspondiente a la propia empresa. Se trata de una pequeña imagen.
- *Nombre del proyecto* para el cual tiene lugar la reunión. Es un campo que se rellena de forma automática y no se puede modificar.
- *Responsable del proyecto*, a pesar de que probablemente haya un gran número de personas involucradas en él, es interesante que se destaque al responsable principal de forma clara.
- *Lugar* donde se realiza, dentro de la ciudad, en el edificio de qué institución, entidad pública o privada...
- *Hora* a la que empieza la reunión.
- *Duración* de la misma.
- *Motivo de la reunión*, que aclare brevemente el objetivo para el que se han citado esas personas a la reunión.

El apartado *Participantes* se genera de la misma forma que en el DDP, solo que se deja un campo más junto al nombre para indicar si la persona que estaba citada a la reunión ha asistido o no (ver Capítulo 5). El nombre de este campo es "¿Asiste?" y sólo se puede seleccionar "Sí" o "No".

Los apartados *Observaciones* y *Conclusión* están formados por un único campo de texto. Las observaciones se utilizan por si hubiera algo especial que mencionar acerca del transcurso de la reunión, como la explicación de algún suceso. En las conclusiones se destacará, si es necesario, algún aspecto relevante que se haya concluido y existe un campo adicional para establecer la fecha de la próxima reunión ("Fecha próxima reunión").

El apartado Liberación contiene los siguientes datos:

- *Firmas*: tantos campos como entidades participan en la reunión. En estos campos se podrá escribir el nombre del representante de cada entidad.

Asociados a cada uno de estos campos hay otro tres que permiten introducir la *fecha*, *hora* y el *asunto* de un posible correo electrónico. Las actas se pueden enviar por correo electrónico en formato pdf. La respuesta positiva a este correo por parte de las entidades puede considerarse una firma. Para dejar constancia de ello, de esta manera se recuerda qué correo electrónico ha servido de respuesta.

- *Fecha del acta* en el formato "En _____, a ____ de _____ de 20__".

Los apartados *Orden del día*, *Desarrollo de la reunión* y *Acuerdos tomados* se forman a partir de los elementos de memoria. Tienen un tipo especial: *OD*, *DR*, *AT* respectivamente

El orden de los apartados es el que sigue a continuación, pero para tener una idea más clara se puede ver el apartado de presentación de datos (apartado *Visualización y Generación de los Documentos*).

- Cabecera, Participantes, Observaciones, Orden del día, Desarrollo de la reunión, Acuerdos tomados, Conclusión y Liberación.

Interfaz Privada: Visualización y Generación de los Documentos

El usuario del sistema tendrá la posibilidad de generar documentos en formato html o pdf con las siguientes particularidades:

- Los usuarios con el rol de lector sólo podrán generar la información relativa a los lectores. Por ejemplo, los elementos que no se hayan publicado todavía no aparecerán en sus documentos.
- La información se genera según las opciones de visualización presentes en el instante de realizar la generación de documentación. En cualquier caso, en un paso previo a dicha generación aparecerá un resumen de las opciones marcadas.
- Si hay elementos de memoria marcados como no visibles para el usuario, éstos no aparecerán en el documento.
- Si el usuario que genera el documento es un autor podrá decidir si prefiere que los elementos inactivos aparezcan o no en el documento generado. En cualquier caso habrá una marca en el documento indicando que se trata de un elemento no activo en esta fecha.
- En caso de haber algún fichero adjunto relacionado con el elemento de memoria, sólo se advertirá de ello indicando el nombre del fichero adjunto.
- El resumen de las opciones de visualización aparecerá en el documento generado para evitar confusiones, así como la fecha de generación, el nombre del usuario que lo ha generado y sus características (roles, jerarquía, etc.).

- Además se mostrará la fecha del primer elemento de memoria introducido y la fecha del último, para conocer qué intervalos de tiempo se manejan.
- Un usuario podría indicar que el documento a generar es para un usuario de jerarquía igual o inferior. De esta manera el documento generado tiene las opciones de visualización del usuario que lo genera, pero las restricciones del usuario objetivo. Esta información se vería reflejada en el resumen de opciones de visualización.
- En el caso de que se genere un:
 - DDP
 - Cada título de *elemento de memoria* se corresponde con un título de sección en el documento a generar, tal y como se puede observar en el formato de este mismo documento ("Bitácora de Proyectos").
 - Si un elemento está "Bajo..." otro elemento, se mostrará como una subsección suya (2, 2.1, 2.2, 3, etc.).
 - El DDP comenzará con una portada con el nombre del DDP, subtítulo, imagen de portada, autor, texto, palabras clave separadas por `;`, la fecha de creación y si se encuentra activo o no. A continuación seguirá un índice con los nombres de las secciones y subsecciones. Después la lista de los contactos. Y seguidamente cada una de las secciones y subsecciones con toda la información relativa y con el formato que el usuario ha dado haciendo uso del editor.
- Acta
 - Al contrario que el DDP, existe más de un acta. De hecho, tantas como se consideren oportunas. Haciendo clic sobre el asunto del acta, el nombre o la fecha se puede visualizar el documento.
 - Si un elemento está "Bajo..." otro elemento, se mostrará como una subsección suya (2, 2.1, 2.2, 3, etc.).
- Informe de estado
 - El *informe de estado* se genera mostrando cada elemento marcado como *Incidencia y/o General* (opción seleccionable).
 - Cada *elemento de memoria* se corresponde con un punto en el documento a generar (típica lista de puntos de un procesador de textos).
 - Si un elemento está "Bajo..." otro elemento, se mostrará en el párrafo siguiente pero no dentro de un punto diferente, puesto que podría ser una corrección, comentario, etc. al anterior.
 - Se visualizan toda la información en un intervalo de fechas. Este intervalo se muestra en la portada del documento.
 - La información se mostrará claramente, diferenciando elementos de memoria de los diferentes tipos (DDP, Acta, etc.).
- Informe de Relaciones
 - Da cuenta de todas y cada una de las relaciones que tienen todos y cada uno de los elementos de memoria (por agilidad sólo se muestran los títulos con las fechas de creación).
 - En el formato HTML los elementos son enlaces para poder ver el resto de la información. En el pdf sólo se muestran los títulos y fechas.

- Las restricciones se mantienen por usuario, con lo que los elementos que no puedan ser visualizados por el usuario que genera el documento no podrá ser mostrado.
- Informe de Usuarios
 - Para cada usuario se muestra su información, así como los elementos que tiene destacados como de su interés, los que son visibles para él y a los que está suscrito. También se muestran los elementos de los que es autor.
 - En el formato HTML los elementos son enlaces para poder ver el resto de la información. En el pdf sólo se muestran los títulos y fechas.
 - Las restricciones se mantienen por usuario, con lo que la información que no pueda ser visualizada por el usuario que genera el documento no podrá ser mostrada.
- Informe de Elementos de Memoria
 - Para cada título de elemento de memoria se muestra la información del Informe de relaciones más los grupos y usuarios de interés asociados, los grupos y usuarios suscritos, para quién es visible.
 - En el formato HTML los elementos son enlaces para poder ver el resto de la información. En el pdf sólo se muestran los títulos y fechas.
 - Las restricciones se mantienen por usuario, con lo que la información que no pueda ser visualizada por el usuario que genera el documento no podrá ser mostrada.
- Informe de Suscripciones
 - Muestra las suscripciones por usuario o por título de elemento.
 - En el formato HTML los elementos son enlaces para poder ver el resto de la información. En el pdf sólo se muestran los títulos y fechas.
 - Las restricciones se mantienen por usuario, con lo que la información que no pueda ser visualizada por el usuario que genera el documento no podrá ser mostrada.

Marcas de Tiempo

Una de las ventajas de la *Bitácora de Proyectos* proviene de la existencia de *marcas de tiempo* que permiten retroceder en el estado del proyecto. En todo momento se puede observar el estado en el que estaba la documentación del proyecto en un determinado instante de su vida. Por ejemplo, se podría ver la información relativa al proyecto antes de que entrara una persona a en él, o antes de que se tomara cierta decisión. Se mostraría exactamente la misma información que se hubiera visto ese día.

Así mismo, esta característica temporal permite la incorporación de nuevas personas al proyecto de manera natural, puesto que pueden introducirse al mismo de una forma gradual, tal y como ha ido desarrollándose. La Bitácora se convierte en el primer tutor.

Todos los *elementos de memoria* del sistema tienen una *marca de tiempo* que es precisamente el momento en el que el ingresa en el sistema.

Existen dos posibilidades a la hora de realizar las *marcas de tiempo*:

- *Automática:* el *Administrador del Sistema* podrá decidir la periodicidad de las marcas. Si decide que las marcas se establecerán mensualmente, cada mes habrá una marca de tiempo cuyo nombre será "(dd/mm/aa) Marca Automática xxx", donde "xxx" es un conjunto de números consecutivos que se van incrementando y "dd/mm/aa" es la fecha en la que se establece la marca en formato día, mes, año. Por defecto la marca automática está deshabilitada.
- *Manual:* el administrador podrá realizar marcas de tiempo de forma manual en cualquier momento. En este caso podrá dar un nombre a la marca de tiempo que está generando. La fecha se concatena automáticamente. Por ejemplo: "(dd/mm/aa) Nombre de marca de ejemplo".

Es importante destacar que no es necesario configurar las marcas de tiempo al inicio del proyecto, sino que se puede realizar en otro instante.

Un Usuario del Sistema tiene la opción de visualizar el proyecto según las marcas de tiempo de la siguiente manera:

- Toda la información del proyecto hasta una marca de tiempo.
- Toda la información del proyecto desde una marca de tiempo.
- Toda la información del proyecto entre dos marcas de tiempo.

Ventajas del uso de la Bitácora de Proyectos

A continuación se enumeran algunas ventajas que proporcionaría este sistema colaborativo:

1. Permite recoger la información puntual, es decir, la información y los datos que se generan fuera del entorno formal del proyecto: tras una llamada telefónica, en una reunión informal, conversaciones extraordinarias, decisiones personales de analistas o programadores, etc. De esta manera se evitan males entendidos y la imposibilidad de demostrar decisiones puntuales no registradas.
2. Posibilita introducir información adicional sobre cambios en el proyecto desde cualquier lugar y en cualquier momento, pues se trata de una aplicación basada en tecnología Web que siempre está disponible a través de un portal: el sistema reside en un servidor y se accede desde cualquier máquina a través de Internet.
3. Facilita que todos los miembros de la o las organizaciones encargadas del proyecto estén informadas al detalle. Toda la información está guardada en este sistema y puede ser consultada en cualquier momento.
4. El *sistema de procesado progresivo de la información* permite ver el avance de la información. Esto es, facilita que un nuevo miembro que acaba de entrar a formar parte en el proyecto se auto forme viendo la progresión del mismo, en lugar de encontrarse con una cantidad importante de información y se pierda en la abundancia.
5. Orienta el modo de trabajo en una organización poniendo a disposición plantillas para documentos importantes como:
 - a. Documento de definición de proyecto (plantilla)
 - b. Actas sobre las reuniones (plantilla)
 - c. Informes de estado periódicos (plantilla)

6. Por medio de los *elementos de memoria* se pueden generar automáticamente documentos importantes en formato pdf como podrían ser un informe de estado del proyecto en un momento puntual, un resumen de las actas de las reuniones mantenidas, una memoria completa de lo acontecido hasta el momento, etc.
7. Se puede realizar una búsqueda semántica de elementos de memoria relacionados para conocer qué se decidió en torno a algo concreto o para ver simplemente qué es lo que se ha estado comentando sobre ese algo.

La herramienta se podría ir ampliando para que contemple de una manera particularizada roles, agenda compartida de grupo, contactos, integración con otras aplicaciones, áreas de comunicación síncrona y asíncrona internas, áreas privadas según roles, vistas diferentes según roles, etc.