

PROPUESTA DE MODELO INTEGRADOR PARA LA MEJORA DE ESPACIOS COLABORATIVOS VIRTUALES

Alumna

Srta. Jasna Mihalache P.

Director

Dr. Alejandro Armando Hossian

Co - Director

Dr. Darío Rodríguez

TRABAJO PRESENTADO PARA OBTENER EL GRADO
DE
MAESTRIA EN INGENIERIA DE SISTEMAS DE INFORMACIÓN

**ESCUELA DE POSGRADO
FACULTAD REGIONAL BUENOS AIRES
UNIVERSIDAD TECNOLÓGICA NACIONAL**

MARZO, 2016

DEDICATORIA

A Dios

Por haberme acompañado en todo este tiempo fortaleciendo mi corazón e iluminando mi mente para lograr este objetivo, por el privilegio de haberlo conocido y servirle. Gracias por su infinito amor.

A mi padre Constantino

Por los ejemplos de responsabilidad, perseverancia y constancia que lo caracterizaron y que me ha infundado siempre, por el valor mostrado para salir adelante. Aunque ya no está con nosotros seguramente estaría orgulloso de mí.

A mi madre Norma

Por haberme apoyado con amor, cariño, comprensión, consejos, valores y la motivación constante. Por soportar estos años lejos de ella, por acompañarme en los buenos y malos momentos. Por ayudarme a que este momento llegara.

A mis familiares

Al resto de mi familia, en especial a mi sobrino Dylan por su afecto, son una gran bendición de Dios, que dan muestra de las buenas cosas que tiene para dar.

AGRADECIMIENTO

A la Universidad Tecnológica Nacional, porque en sus aulas, recibimos el conocimiento intelectual y humano de cada uno de los docentes de la Escuela de Posgrado.

Especial agradecimiento al Director de Tesis el Dr. Alejandro Armando Hossian y al Dr. Darío Rodríguez por sus consejos y amistad.

RESUMEN

La solicitud al área de Sistemas para dar soporte a la comunicación de grupos de individuos u organizaciones durante la realización de tareas compartidas crece incesantemente. Sin embargo, por su propia naturaleza, esta clase de sistemas, los cuales conocemos como Sistemas Cooperativos, son bastante complejos. Los modelos existentes tienen características específicas que se destacan unos de otros, cubriendo así las necesidades de los diferentes grupos de trabajo.

ABSTRACT

The application systems to support communication of groups of individuals or organizations for the shared tasks grows steadily. However, by their nature, these types of systems, which we know as Cooperative Systems, are quite complex. Existing models have distinct characteristics that stand out from each other, covering the needs of different workgroups.

ÍNDICE

1. INTRODUCCIÓN	1
1.1 Contexto de la tesis.....	1
1.2 Objetivo de la tesis.....	1
1.3 Visión General de la Tesis.....	1
2. ESTADO DEL ARTE	4
2.1 Descripción de los modelos.....	4
2.1.1 Modelo Amenities.....	5
2.1.2 Modelo Touche.....	10
2.1.3 Modelo Ciam.....	15
2.1.4 Modelo R-bac.....	20
2.1.5 Modelo Sem-hp.....	23
2.1.6 Modelo Kaos.....	27
2.1.7 Modelo CTT.....	34
2.2 Consideraciones sobre los distintos modelos.....	38
2.2.1 Características consideradas en los modelos.....	38
2.2.1.1 Amenities.....	38
2.2.1.2 Touche.....	41
2.2.1.3 Ciam.....	43
2.2.1.4 R-bac.....	45
2.2.1.5 Sem-hp.....	47
2.2.1.6 Kaos.....	49
2.2.1.7 CTT.....	50
2.3 Cuadro Comparativo.....	51
2.4 Valoraciones del cuadro comparativo.....	56
3. DESCRIPCIÓN DEL PROBLEMA	58
3.1 Características o conceptos no existentes en los modelos estudiados	58
3.2 Discusión de la Integración de los modelos en una propuesta.....	62

3.3 Unificación de los modelos en una propuesta.....	62
4. SOLUCIÓN	64
4.1 Modelo Integrador.....	64
4.2 Propuesta para la representación de Modelos de Tareas usando Redes Hipermedia.....	68
4.3 Reseña de Patrones de Organización Joint Venture y su modelado con diferentes modelos.....	78
4.3.1 Un método para la construcción del modelo colaborativo en base a Patrones.....	89
4.3.2 Caso de estudio: Modelado de un proceso de aprendizaje colaborativo basado en la estrategia de Jigsaw.....	90
4.3.2.1 Modelado de la Organización.....	91
4.3.2.2 Modelado de los Roles.....	92
4.3.2.3 Modelado de las Actividades.....	93
5. CONCLUSIONES Y FUTUROS TRABAJOS DE INVESTIGACION	100
5.1. Aportes.....	100
5.2 Futuros trabajos de Investigación.....	101
6. REFERENCIAS	102

ÍNDICE DE FIGURAS

Figura 2.1	Esquema general de la metodología AMENITIES incluyendo las vistas del modelo cooperativo.....	8
Figura 2.2	El flujo de trabajo (etapas) de la metodología TOUCHÉ.....	13
Figura 2.3	Relación entre los diferentes modelos que se contemplan en el modelo conceptual en el que se basa TOUCHE.....	13
Figura 2.4	Esquema clasificatorio de las relaciones organizativas de un sistema colaborativo.....	14
Figura 2.5	Diagrama de Inter-Acción de CIAM.....	19
Figura 2.6	Elementos básicos del modelo RBAC.....	22
Figura 2.7	Estructura del modelo SEM-HP.....	24
Figura 2.8	Niveles meta, de dominio y de instancia de KAOS.....	29
Figura 2.9	Modelo de metas de KAOS.....	31
Figura 2.10	Modelo de objetos de KAOS.....	32
Figura 2.11	Modelo de responsabilidad de KAOS.....	33
Figura 2.12	Plantilla textual de una operación de KAOS.....	33
Figura 2.13	Detalle de la vista de la organización en KnowCat; modo supervisado y modo activo.....	39
Figura 2.14	Modelo conceptual basado en el modelo conceptual para interfaces de usuarios abstractas.....	42
Figura 2.15	Integración entre CIAM y UML.....	44
Figura 2.16	Elementos del modelo RBAC.....	45
Figura 2.17	Jerarquía de ejemplo.....	46
Figura 2.18	Modos de navegación permitidos en SEM-HP.....	49
Figura 3.1	Modelos estudiados con características no existentes.....	58
Figura 4.1	Modelos estudiados generando un modelo Integrador.....	64
Figura 4.2	Modelo de tareas.....	69
Figura 4.3	Diagrama de la organización Pictionary y Equipo.....	72

Figura 4.4	Red hipermedia (nodos sombreados, cambios de rol).....	72
Figura 4.5	Diagrama de rol JTurno.....	73
Figura 4.6	Diagrama de tarea Jugar y diagrama de subActividad TurnoJugada.....	74
Figura 4.7	Red hipermedia conexión rol-tarea JugadorTurno-Jugar y conexión tareas-subactividadesJugar-Turno-Jugada.....	75
Figura 4.8	Red hipermedia de la tarea Jugada.....	77
Figura 4.9	Vista parcial de la red de interconexión entre patrones.....	88
Figura 4.10	Diagrama de organización para la estrategia Jigsaw.....	92
Figura 4.11	Diagramas de rol para la estrategia Jigsaw.....	93
Figura 4.12	Diagrama de actividad para modelar la tarea cooperativa EjecutarJigsaw	95
Figura 4.13	Modelo correspondiente al patrón NEGOCIACIÓN NO MODERADA	96
Figura 4.14	Diagrama de actividad correspondiente a EjecutarTarea.....	97

ÍNDICE DE TABLAS

Tabla 2.1	Definición de conceptos para el marco de trabajo del modelo cooperativo de AMENITIES.....	9
Tabla 2.2	Tabla de participación.....	17
Tabla 2.3	Tipos de tareas en CIAM.....	17
Tabla 2.4	Ejemplo de modelo de Responsabilidad para un rol ficticio llamado Estudiante.....	18
Tabla 2.5	Operadores temporales definidos en la notación CTT.....	37
Tabla 4.1	Notación para el modelado de patrones de organización.....	79
Tabla 4.2	Plantilla de la descripción.....	80
Tabla 4.3	Descripción del patrón Joint Venture.....	81
Tabla 4.4	Estructura del catálogo.....	86

NOMENCLATURA

CSCW	Apoyo de trabajo cooperativo de computadora (de las siglas en inglés Computer supported co-operative work)
Amenities	Metodología para el Análisis y Diseño de Sistemas Cooperativos (de las siglas en inglés Methodology for Analysis and Design of Cooperative Systems)
UML	Lenguaje de modelado de sistemas de software (de las siglas en inglés Unified Modeling Language)
TOUCHÉ	Modelo de procesos orientados a tareas y centrado en el usuario para el desarrollo de Interfaces para ambientes humanos (de las siglas en inglés Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments)
CIOs	Objetos Concretos de Interacción (de las siglas en inglés Concrete Interaction Objects)
AIO	Objetos Abstractos de Interacción (de las siglas en inglés Abstract Interaction Objects)
UsiXML	XML basado en lenguaje de usuario (de las siglas en inglés USeR Interface eXtensible Markup Language)
HCI	Interacción de computadora (de las siglas en inglés Human computer interaction)
CIAM	Metodología de colaboración Interactiva de Aplicaciones (de las siglas en inglés Collaborative Interactive Applications Methodology)
CIAN	Notación de Aplicaciones de colaboración Interactivo (de las siglas en inglés Collaborative Interactive Applications Notation)
RBAC	Control de acceso basado en roles (de las siglas en inglés Role-based access control)

X-RBAC	Control de acceso basado en roles usando XML
NIST	Instituto Nacional de estándares y tecnología (de las siglas en inglés National Institute of Standards and Technology)
SEM-HP	SEMántico, Sistémico y Evolutivo para el desarrollo de Sistemas HiPermedia adaptativos (de las siglas en inglés Systemic and Evolutionary model to develop HyPermedia Systems)
KAOS	Adquisición de conocimientos en la memoria automatizado (de las siglas en inglés Knowledge Acquisition in autOmated Specification)
CTT	ConcurTaskTrees
KnowCat	Catalizador de conocimiento (de las siglas en inglés Knowledge Catalyser)
Awareness	La percepción que el usuario tiene de lo que el resto de usuarios hacen en el sistema
PMP	Pattern Modeling Profile
FreeWare	Software gratis

1. INTRODUCCION

Este trabajo de tesis describe diferentes aspectos de los sistemas que integran la participación de muchos usuarios en un mismo proyecto y que se encuentran en distintos lugares conectados a través de una red. Esta tecnología se denomina modelos de espacios virtuales, sistemas colaborativos o groupware que fue concebido para compartir información. Algunas características de estos sistemas son compartir un repositorio de archivos o chat para comunicar al grupo, compartir eventos de un calendario, etc.

En este capítulo se realiza una síntesis del siguiente trabajo de tesis. En tal sentido se presenta un contexto (sección 1.1), el objetivo de la tesis (sección 1.2) y la visión general (sección 1.3).

1.1. CONTEXTO DE LA TESIS

El contexto del presente trabajo de tesis se focaliza en los sistemas colaborativos. Dentro de este ambiente, se realiza un análisis de los modelos existentes que se consideran más relevantes en la actualidad.

1.2. OBJETIVO DE LA TESIS

El objetivo de este trabajo de tesis consiste en facilitar la evaluación de los sistemas colaborativos existentes, el análisis y contraste de los mismos, para generar finalmente un Modelo Integrador.

1.3. VISIÓN GENERAL DE LA TESIS

El término CSCW lo introducen Grief y Cashman como “Una vía para describir como la tecnología de los computadores puede ayudar a los usuarios a trabajar juntos en grupos”.

CSCW es la disciplina científica que describe como desarrollar aplicaciones Groupware, teniendo también por objeto el estudio periódico y práctico de cómo las personas trabajan en cooperación y como afecta al grupo de trabajo. Su objetivo es observar la forma en que las personas interaccionan y colaboran entre ellas.

Debido al carácter interdisciplinario del tema, la cantidad de campos en los que se podría aplicar Groupware es muy amplia. Desde un punto de vista más general, los sistemas tienden a ser colaborativos para facilitar la comunicación humana y constituir una herramienta útil en el desarrollo de los procesos y coordinación de actividades. En este sentido, se espera que la cantidad de aplicaciones, futuras mejoras, y formas de implementación de estos sistemas, se incremente de manera progresiva en los próximos años.

El término Groupware hace referencia al hardware y al software que soportan y optimizan el trabajo en grupo. El objetivo del Groupware no es eliminar otros tipos de trabajo cooperativo, sino añadir un nuevo componente en el proceso de colaboración. Por consiguiente, el Groupware es un conjunto de productos orientados al trabajo en grupo, los cuales ayudan a los grupos de personas a trabajar juntos.

Aunque los términos de CSCW y Groupware no son sencillos de definir, CSCW se orienta hacia la investigación, mientras que Groupware lo hace hacia la tecnología. Es común diferenciar estos términos diciendo que Groupware está enfocado tecnológicamente y facilita las formas de cooperación que abarca CSCW. Se considera que Groupware es la tecnología que permite, ya sea con hardware, software, servicios y/o técnicas, que las personas trabajen en grupo. Por su parte, CSCW se focaliza en el estudio de las herramientas y técnicas que utilizan los grupos de trabajo; así como su perfil social, psicológico y su impacto organizacional.

Suele expresarse que el Groupware colabora sustancialmente para mejorar la eficacia en tres niveles: Comunicación, Coordinación y Cooperación.

La comunicación es el proceso de intercambio de mensajes en las organizaciones y se intenta que la misma sea eficaz; en otras palabras, que quien envía y quien recibe la

información perciban el mismo concepto. En tal sentido, también es importante que sea eficiente en cuanto a un consumo mínimo de los recursos.

La coordinación es un conjunto de mecanismos del grupo u organización utilizados para establecer un enlace coherente entre las actividades de cada subunidad.

Las organizaciones emplean cinco mecanismos:

- I. El ajuste mutuo
- II. La supervisión directa
- III. Las estandarizaciones de los procedimientos de trabajo, de los resultados obtenidos y de los métodos utilizados
- IV. La cooperación
- V. La participación intencionada y coordinada de los miembros de un grupo.

Aspectos claves del CSCW son el concepto de grupo, la interfaz multiusuario, el control de concurrencia, la coordinación y comunicación dentro del grupo, los espacios de información compartida y el soporte de un entorno abierto heterogéneo que integre aplicaciones preexistentes.

2. ESTADO DEL ARTE

En este capítulo se describe la importancia de trabajar en equipo y las diferentes herramientas que nos han permitido mejorar día a día (sección 2), dentro de los modelos de espacios virtuales estudiados se encuentran el modelo AMENITIES (sección 2.1), modelo TOUCHE (sección 2.2), modelo CIAM (sección 2.3), modelo RBAC (sección 2.4), modelo SEM-HP (sección 2.5), modelo KAOS (sección 2.6), modelo CTT (sección 2.7).

2.1. DESCRIPCIÓN DE LOS MODELOS

Lo más importante en una organización son las personas, es muy importante generar espacios y brindarles herramientas para que puedan trabajar en conjunto y ayuden a la empresa a ser más productiva, funcionando correctamente.

Para ello existen metodologías en evolución como las herramientas de trabajo cooperativo soportadas por computador o CSCW.

Existen dos productos muy conocidos de este tipo de herramientas como lo son LOTUS NOTES de la compañía IBM y Microsoft Exchange, software usado como herramienta de soporte de e-mail a nivel corporativo.

El término CSCW, fue usado por primera vez por Irene Greif and Paul M. Cashman en 1984, este es un campo académico que trae a colación una concatenación de profesionales como Psicólogos sociales, sociólogos, antropólogos y científicos de computación, todos trabajando en conjunto.

El objetivo de estos profesionales es entender las características del trabajo en grupo interdependiente, con el objetivo de diseñar la tecnología adecuada basada en el uso del computador para soportar este trabajo colaborativo.

Existen tres dimensiones del trabajo colaborativo:

- **Conciencia:** Los individuos trabajan juntos con la necesidad de conseguir algún nivel de conocimiento de los demás individuos.

- Trabajo Articulado: Los individuos deben saber cómo partir el trabajo en unidades, dividir las entre sí, trabajar con ellas y reintegrarlas.
- Apropiación: Como un individuo o grupo adapta la tecnología a su propia situación particular.

2.1.1. MODELO AMENITIES

AMENITIES, a Methodology for Analysis and Design of Cooperative Systems, es una metodología para diseñar sistemas colaborativos que hace uso de UML, plantea el uso de diferentes modelos para comprender el funcionamiento de un sistema desde distintas perspectivas.

Esta metodología propone un marco conceptual para describir las principales entidades (y sus relaciones), que habitualmente aparecen en un sistema colaborativo. Conforme a este marco conceptual, una acción es una unidad atómica de trabajo. Su ejecución dirigida por eventos puede requerir/modificar/generar información explícita.

Una subactividad es un conjunto de otras subactividades y/o acciones. Una tarea es un conjunto de subactividades pensadas para alcanzar ciertos objetivos. Un rol es un identificador para un conjunto de tareas relacionadas a realizar. Un actor es un grupo, usuario, programa o entidad con ciertas capacidades adquiridas (destrezas, categoría, etc.), que puede desempeñar un rol al ejecutar (usando artefactos), o ser responsable de acciones. Los grupos realizan ciertas subactividades en función de ciertos protocolos de interacción. Un grupo es un conjunto de actores organizados en torno a una tarea y puede estar compuesto de otros subgrupos. Una tarea cooperativa es aquella que debe realizarse por más de un actor, desempeñando el mismo o diferentes roles. Una ley es una limitación o restricción del sistema que permite ajustar dinámicamente el conjunto de comportamientos posibles. Una organización consiste en un conjunto de roles relacionados. Finalmente, un sistema colaborativo consta de grupos, organizaciones, leyes, eventos y artefactos.

Cuatro vistas que facilitan detectar los aspectos que contemplan este modelo (figura 2.1):

➤ Vista de grupo

Primeramente debemos identificar los aspectos relacionados con la propia organización (el grupo), y las restricciones que impone esta asociación. Las organizaciones se articulan bajo el concepto de rol, que determina la relación entre los miembros del grupo y las tareas que deben llevar a cabo. A menudo esta relación está condicionada por una serie de restricciones impuestas al sistema colaborativo, y de las cuales podemos identificar las siguientes como más importantes:

- Capacidades: Esta es una restricción cognitiva que se impone a cada actor para participar con un rol determinado. Estas capacidades determinan los conocimientos que debe adquirir un usuario para participar con un rol concreto.
- Leyes: Este tipo de restricción viene impuesta por la propia organización e identifica las reglas que deben ser preservadas en el grupo. Normalmente estas reglas se deducen de la propia estructura social que se manifiesta en el grupo (democrática, jerárquica, etc.)

Ambas restricciones permiten modelar sistemas dinámicos, ya que es habitual que tanto la estructura del grupo como su funcionamiento se modifique en el tiempo (los participantes pueden adquirir nuevas capacidades, variar en número de miembros que lo conforman o bien, modificar las leyes que rigen el grupo al aplicar nuevas estrategias de trabajo).

➤ Vista cognitiva

La vista cognitiva representa el conocimiento que posee o adquiere cada miembro del grupo en el escenario colaborativo. Este conocimiento queda reflejado mediante la descripción de las tareas que puede llevar a cabo.

La descripción de las tareas implica un análisis profundo de las actividades que se deben realizar en el grupo, la división del trabajo y determinar las interrelaciones que existen entre ellas.

➤ Vista de interacción

Otro aspecto que debemos estudiar son los procesos que implican un diálogo entre participantes para analizar sus características, concretamente:

- El modo de diálogo que se producen entre participantes
- Los requisitos que impone ese diálogo sobre los medios a utilizar

Este modo de diálogo lo identificaremos mediante protocolos. Los protocolos se pueden analizar por separado dentro de la organización ya que en gran medida son independientes del dominio del problema, y por tanto, se pueden incorporar al análisis de tareas.

Por ejemplo, se pueden identificar protocolos democráticos (toma de una decisión por mayoría), consenso (aprobación unánime de una decisión), jerárquica, etc.

➤ Vista de información

Por último, deberemos recoger la información que es compartida en el escenario.

Esta información se puede describir de manera implícita en las actividades y acciones o bien, de modo explícito como flujo de información entre actividades. La información que fluye a través del sistema colaborativo serán los documentos (los objetos que son gestionados en el sistema), eventos y recursos.

Marco Conceptual del Modelo Cooperativo

AMENITIES define el marco conceptual en el que se basa para modelar la parte cooperativa de los sistemas. Así mismo, se define un conjunto de relaciones entre los conceptos mencionados con anterioridad (tabla 2.1):

- 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.

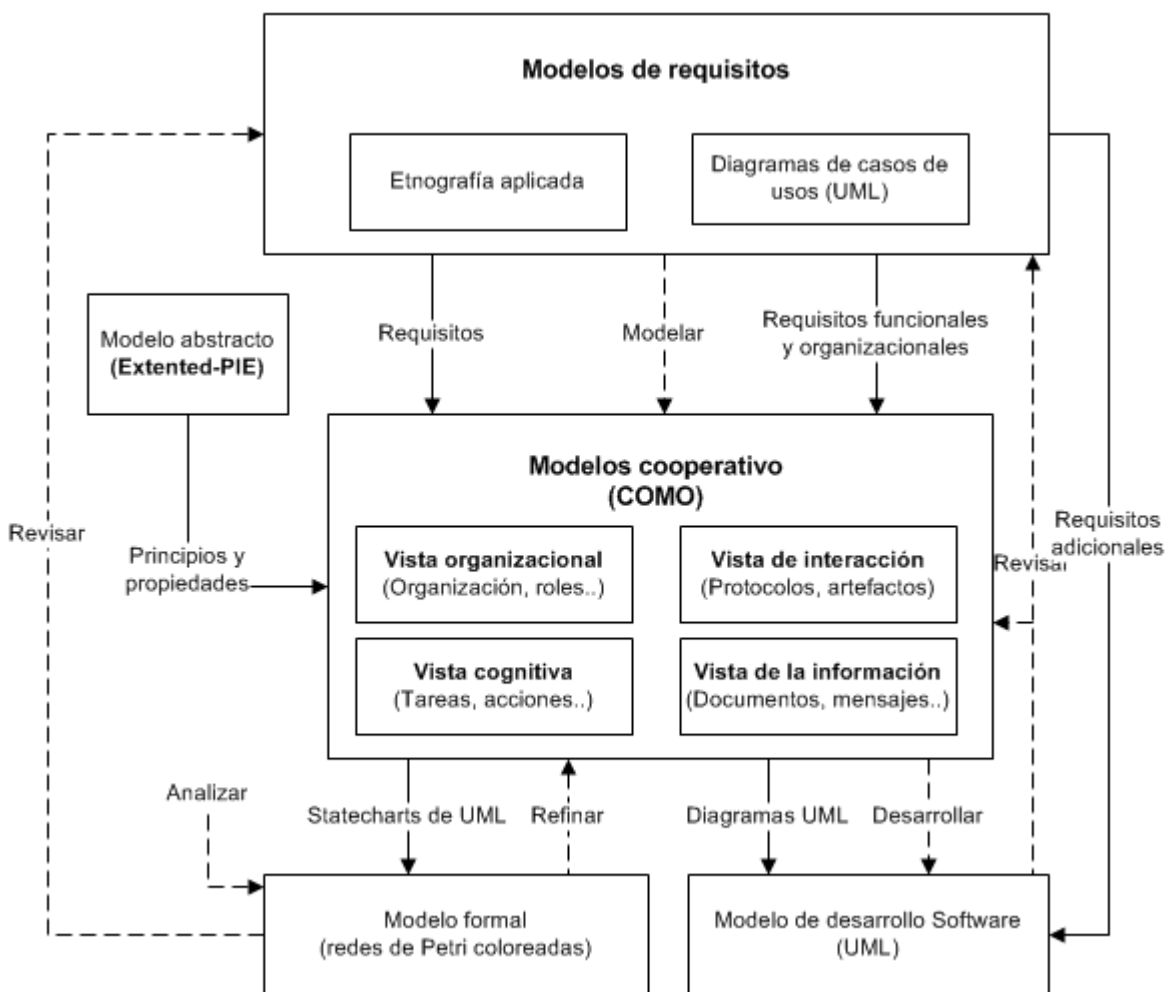


Figura 2.1. Esquema general de la metodología AMENITIES incluyendo las vistas del modelo cooperativo.

Concepto	Definición
Evento	Ocurrencia de algún hecho que tiene 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 subacciones/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.
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.

Tabla 2.1. Definición de conceptos para el marco de trabajo del modelo cooperativo de AMENITIES

2.1.2. MODELO TOUCHÉ

TOUCHÉ, es un modelo de proceso y una metodología para el desarrollo de interfaces de usuario para aplicaciones groupware desde la elicitación de requisitos hasta su implementación considerando las características y particularidades de estos sistemas desde el inicio.

El nombre “TOUCHÉ” viene de Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments o lo que es lo mismo Modelo de Proceso para el Desarrollo de Interfaces en Entornos CSCW Centrado en los Usuarios y Dirigido por Tareas.

TOUCHÉ es un modelo de proceso y metodología de desarrollo a la vez. Se trata de un modelo de proceso puesto que indica las etapas por las que se ha de seguir desde el inicio, desde el estudio de los requisitos del sistema, hasta la implementación de la interfaz de usuario.

Por otro lado, se trata también de una metodología puesto que indica, en cada una de esas etapas, lo que se ha de hacer para modelar el sistema a desarrollar: Se proponen una serie de artefactos (diagramas, plantillas, etc.) para modelar la realidad de un modo formal o semi-formal de manera que los artefactos guarden una relación entre sí y se mantenga, además, una coherencia entre ellos (trazabilidad).

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 “perderse en la abundancia”.

Por este motivo se concretan las técnicas a emplear en la metodología, sugiriendo aquellas 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 sólo de entornos colaborativos, hace imposible una independencia absoluta entre etapas. Existe pues una relación entre ellas 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. 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 modelo de procesos desarrollado en TOUCHÉ es una combinación entre un proceso de desarrollo tradicional de Ingeniería del Software y uno de Interacción Persona-Ordenador. En el primero se suele tener en cuenta la etapa de requisitos, mientras que en el segundo se suele comenzar desde el análisis de las tareas. El punto clave en esta “unión” ha sido la relación “Requisito - Tarea”. Las etapas del modelo de proceso son las siguientes (figura 2.2):

- **Elicitación de Requisitos:** La primera etapa del modelo reúne los requerimientos del sistema a ser desarrollado para cumplir los objetivos de los usuarios. Aquí consideramos las plantillas para la obtención de requerimientos y el Documento de Requerimiento de Sistema, el cual reúne toda la información.
- **Análisis:** La etapa del Análisis se trata del estudio del dominio. Describe los requerimientos del sistema sin describir las cuestiones de la implementación. Los roles y las tareas son identificadas y descritas. El análisis del sistema se

hace desde una perspectiva estructural a través de Diagramas de Clases y del Diagrama de la Estructura Organizacional; y desde la perspectiva del comportamiento a través del Diagrama de Tareas (La notación CTT ha sido adoptada en este caso) y del Diagrama de Co-Interacción (desarrollado para identificar relaciones entre los actores del sistema).

- **Diseño:** Esta etapa trata sobre cómo representar la información (visualización, entradas, controles, etc.) a los usuarios finales. Toda la información reunida hasta ahora en las etapas previas es procesada y transformada a una representación manejable y ejecutable por software. El “Awareness del Usuario”, el cual se refiere al conocimiento acerca de los otros usuarios del sistema, debería ser considerado a fin de obtener un buen Groupware. Los Objetos Abstractos de Interacción (AIO) son usados para diseñar Interfaces de Usuario Abstractas. Utiliza un esquema conceptual de UsiXML para la especificación de las interfaces de usuario. El modelo es enriquecido con un nuevo AIO y nuevas facetas las cuales proporcionan mayor expresividad para representar este tipo de sistemas.
- **Implementación:** La etapa de Implementación del sistema trata sobre la generación de las interfaces de usuario de los AIOs obtenidos en la etapa previa. Es un proceso en que cada componente pueda obtener más elementos concretos (Objetos Concretos de Interacción o CIO) de acuerdo con la implementación y los detalles de la plataforma. Uno de los últimos estudios sobre interfaces de usuario para entornos HCI es Cameleon.

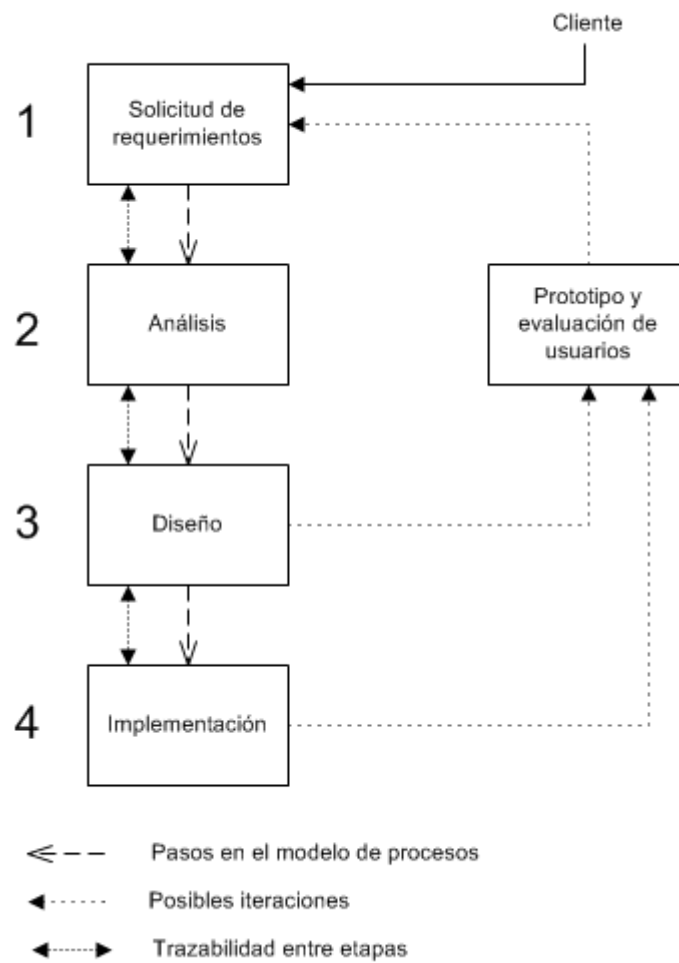


Figura 2.2. El flujo de trabajo (etapas) de la metodología TOUCHÉ.

Touche contempla una relación entre los diferentes modelos al modelo conceptual (figura 2.3). El modelo de organización sirve para escenificar la estructura organizativa en un entorno colaborativo, representando en él la estructura de los actores del sistema y las relaciones de colaboración que tienen lugar entre ellos.

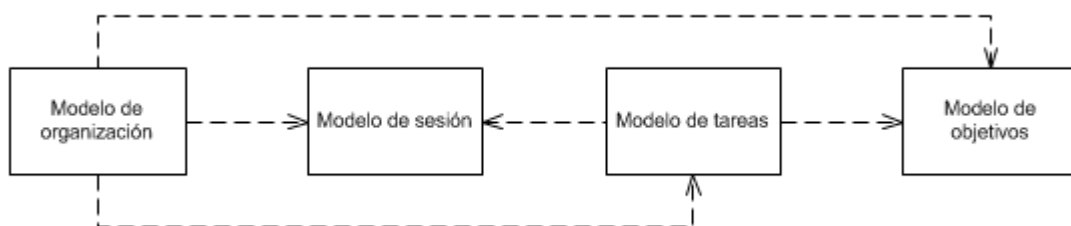


Figura 2.3. Relación entre los diferentes modelos que se contemplan en el modelo conceptual en el que se basa TOUCHÉ.

El modelo de sesión proporciona la vista del sistema en un momento determinado (en el que el usuario está usando dicho sistema). Representa a un escenario concreto.

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.

Una relación organizativa (figura 2.4) 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. La siguiente figura muestra un esquema clasificatorio de las relaciones organizativas del modelo TOUCHE.

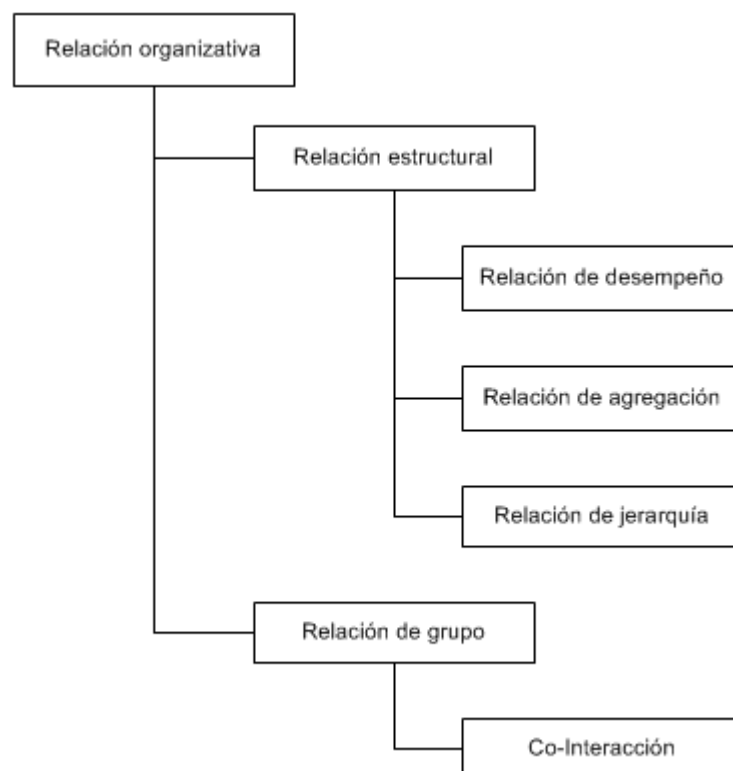


Figura 2.4. Esquema clasificatorio de las relaciones organizativas de un sistema colaborativo.

2.1.3. MODELO CIAM

La metodología CIAM (Collaborative Interactive Applications Methodology) está enfocada al desarrollo de sistemas Groupware.

El marco metodológico de CIAM está basado en la notación CIAN (Collaborative Interactive Applications Notation), la cual permite definir los distintos modelos requeridos en las etapas de la metodología CIAM.

Los primeros estados de la metodología se refieren al modelado centrado en los grupos, que va hacia un modelado más centrado en los procesos (colaborativos, cooperativos y de coordinación) en las etapas siguientes. Detallamos cada vez más profundo en los niveles de abstracción se alcanza un modelado más centrado en los usuarios, en el cual las tareas interactivas son modeladas. Las primeras dos etapas del modelado permiten definir el contexto en el cual se crea el modelo interactivo, además de que sirve como punto de inicio para la última fase.

La información especificada en cada etapa sirve como base para el proceso de modelado hecho en las siguientes etapas, con lo cual dicha información es extendida, relacionada y especificada a mayor nivel de detalle en las siguientes etapas del proceso.

Las etapas y los objetivos de la metodología CIAM son los siguientes:

- **Desarrollo del sociograma:** En este estado se modela la estructura organizacional, así como las relaciones que existen entre sus miembros. Los miembros que forman la organización se encuentran en las siguientes categorías: roles, actores, agentes de software, dando lugar a grupos o equipos de trabajo. Los elementos de estos diagramas pueden ser interconectados a través de tres relaciones básicas: herencia, realización y asociación.

- Modelado de las “Inter-Action”: En esta fase describe las principales tareas (o procesos) que definen el trabajo en grupo desarrollado en la organización definida en la etapa previa.
- Modelado de las responsabilidades: En esta fase, la atención es puesta sobre la perspectiva individual de cada rol en la organización, agregando a las responsabilidades compartidas aquellas que son exclusivas a cada rol. La información especificada en esta etapa es complementaria a la definida en las etapas previas, siendo necesaria para que los dos modelos sean coherentes.
- Modelado de las tareas de trabajo en grupo: En esta fase se describen a mayor detalle las tareas grupales identificadas en las etapas previas. Se distinguen dos tipos de tareas (tabla 2.3), las cuales se modelan de forma distinta: Tareas cooperativas y tareas colaborativas.
- Modelado de la interacción: En este último estado se modelan los aspectos de la aplicación puramente interactivos. Un modelo de interacción es creado para cada tarea de naturaleza individual.

En la etapa de modelado de Inter-Acción se crea la tabla de participación (tabla 2.2), en la cual, usando una especificación de naturaleza textual, permite a los diseñadores tener una primera idea acerca de la división de trabajo a niveles altos de abstracción. Esta tabla está compuesta por las filas que representan las tareas del sistema. Las columnas representan los roles y cada celda significa que un rol participa en una tarea. La columna final identifica el tipo de tarea, ya sea individual, colaborativa o cooperativa, según se muestra en la tabla de tareas.









Tareas	Roles	Estudiantes	Director	Comité académico	Inspector	Tipo
Proyecto del plan		X	X			
Examinar la propuesta		X	X			
Requerimiento		X				
Comité académico de valoración				X		
Cambios sugeridos				X		
DEP Desarrollo		X				
Post-DEP Procedimiento		X	X		X	
Examen		X			X	

Tabla 2.2. Tabla de participación




Tipos de tareas	Icono
Tarea individual	
Tarea cooperativa	
Tarea colaborativa	

Tabla 2.3. Tipos de tareas en CIAM

El modelo de Responsabilidades (tabla 2.4) expresa las responsabilidades que tiene un rol en particular sobre el conjunto de tareas en las que participa, según la Tabla de participación.







Responsabilidad	Tipo de tareas	Objeto en el modelo de dominio	Pre-requisito	
			Tareas	Datos
Proyecto del plan		C/R/W: Proyecto	INI	
Examinar la propuesta		R/W: Proyecto del plan	Proyecto del plan	Plan
Requerimiento		R/W: Proyecto del plan	<ul style="list-style-type: none"> • Proyecto del plan • Examina la propuesta 	Plan
DEP Desarrollo		C/R/W: DEP	Comité académico de valoración	Plan
Post-DEP Procedimiento		R/W: DEP	DEP Desarrollo	DEP
Examen		R/W: DEP	Post-DEP Desarrollo	DEP

Tabla 2.4. Ejemplo de modelo de Responsabilidad para un rol ficticio llamado Estudiante

Este modelo se expresa a través de la tabla 2.4 cuyas filas son las tareas en las que participa un rol específico. Es decir, por cada rol va a haber una tabla o modelo de responsabilidades. La primera columna indica el nombre de la tarea a la que se refiere. La segunda columna indica el tipo de tarea. La tercera columna indica el objeto u objetos del dominio que son manipulados, así como el tipo de acceso requerido (R, Leer; W, Escribir; C, Crear; o combinaciones de éstos). La cuarta y quinta columna expresan los requisitos previos para poder realizar la tarea. Dichos requisitos consisten en la tarea que debe completarse para poder realizar la tarea a la que se refiere la columna, y los datos u objetos del modelo de datos que deben crearse.

Este modelo permite establecer las dependencias temporales o de orden de ejecución entre los procesos principales, así como las dependencias de datos a presentar.

Una vez que las responsabilidades y las tareas principales están definidas, se puede realizar el modelo de inter-acción. Este modelo permite definir la operación completa del proceso grupal, el cual puede ser cooperativo, colaborativo o mixto.

En la siguiente figura se muestra el diagrama de Inter-Acción de CIAM.

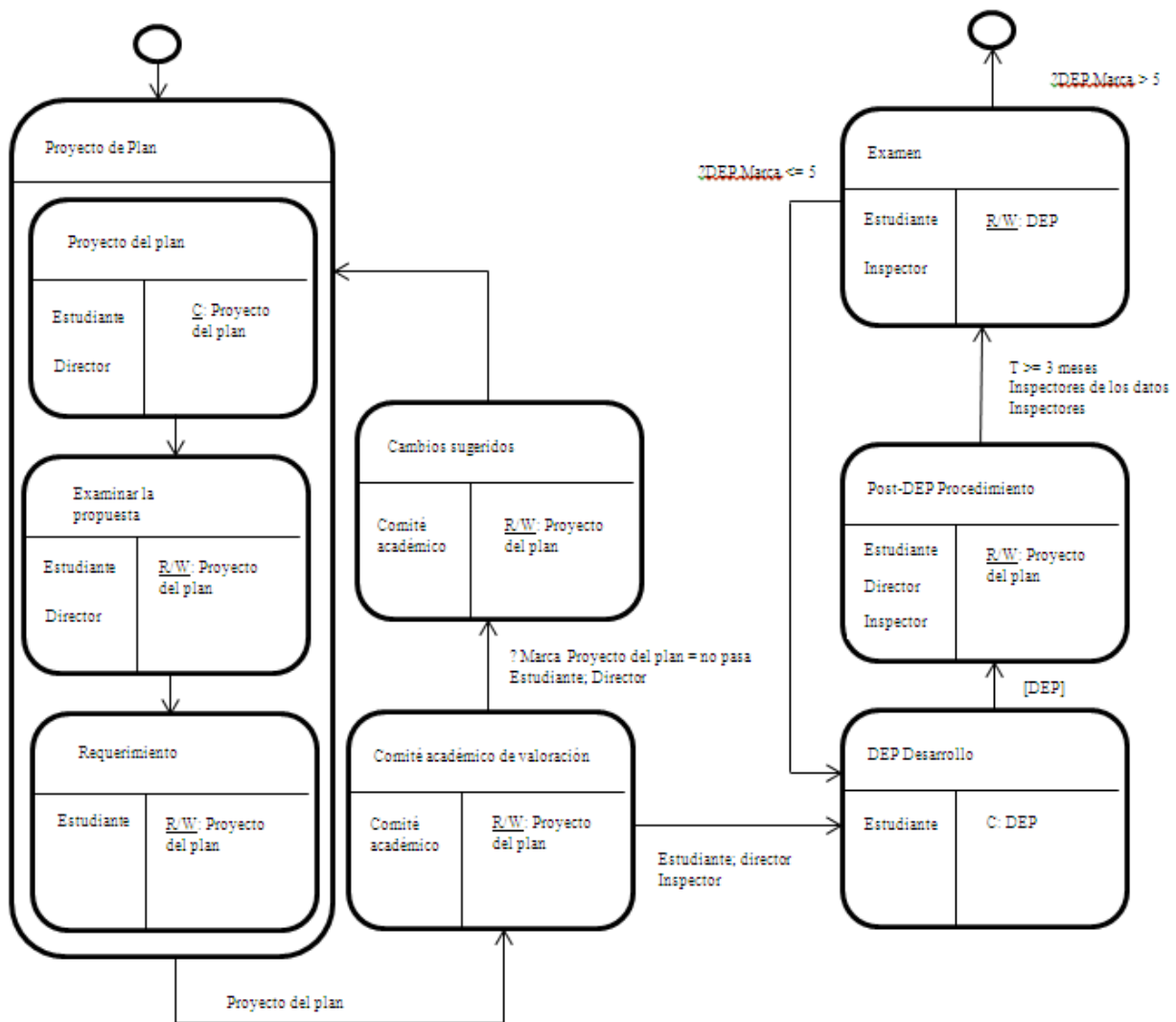


Figura 2.5. Diagrama de Inter-Acción de CIAM

2.1.4. MODELO RBAC

RBAC (Role-Based Access Control) está basado en la definición de un conjunto de elementos y de relaciones entre ellos. A nivel general describe un grupo de usuarios que pueden estar actuando bajo un conjunto de roles y realizando operaciones en las que utilizan un conjunto de objetos como recursos.

Entre estos cuatro elementos se establecen relaciones del tipo:

- Relaciones entre usuario y roles modelando los diferentes roles que puede adoptar un usuario.
- Conjunto de operaciones que se pueden realizar sobre cada uno de los objetos.

A los elementos de esta relación se les denomina permisos.

- Relaciones entre los permisos y los roles. Modelamos cuándo un usuario por estar en un rol determinado tiene permiso para realizar una operación sobre un objeto.

El modelo RBAC incluye un conjunto de sesiones donde cada sesión es la relación entre un usuario y un subconjunto de roles que son activados en el momento de establecer dicha sesión. Cada sesión está asociada con un único usuario y cada usuario tiene una o más sesiones. Los permisos disponibles para un usuario son el conjunto de permisos asignados a los roles que están activados en todas las sesiones del usuario, sin tener en cuenta las sesiones establecidas por otros usuarios en el sistema.

RBAC añade la posibilidad de modelar una jerarquía de roles de forma que se puedan realizar generalizaciones y especializaciones en los controles de acceso y se facilite la modelización de la seguridad en sistemas complejos.

El control de acceso basado en roles permite expresar de forma sencilla y natural la política de accesos a los recursos de una organización compleja. Al usar este modelo como representación de la seguridad en un sistema colaborativo estamos integrando los aspectos de seguridad con los funcionales.

Sin embargo pensamos que RBAC presenta una serie de carencias para el control de acceso en procesos de naturaleza colaborativa:

- En RBAC la naturaleza de los roles puede ser denominada estática, ya que carecen de flexibilidad y sensibilidad para el entorno en el cual son usados.
- RBAC soporta la noción de roles activos para un usuario con el concepto sesión, obteniendo a partir de estos roles activos el conjunto de permisos disponibles para un usuario, pero no tiene en consideración las sesiones establecidas por otros usuarios en el sistema, es decir que el modelo no engloba todo el contexto asociado con el sistema. Por ejemplo, en un entorno educativo, RBAC no permite dar temporalmente permisos exclusivos del rol Director al rol Subdirector como consecuencia de la ausencia en el sistema de un usuario ejerciendo el rol Director.

- No es capaz de especificar un control de grano fino sobre usuarios individuales en ciertos roles y sobre instancias de objetos individuales. Un escenario donde sería preciso establecer un control de grano fino es, por ejemplo, en el ambiente de un hospital donde se crea un grupo de trabajadores sanitarios para dar asistencia médica a un paciente en concreto, en este caso sólo los miembros de este grupo podrán tener acceso al expediente del paciente, además los miembros del grupo que ejerzan el rol Celador no tendrán acceso a las pruebas médicas del paciente.
- En el escenario descrito anteriormente se observa la necesidad de establecer permisos comunes a grupos de usuario. Esto es conseguido en el modelo RBAC creando un rol específico y asignando de forma individual este rol a cada usuario perteneciente al grupo. La posibilidad de la existencia de un gran número de grupos de usuarios en los sistemas colaborativos y que la mayoría de estos grupos sean de carácter temporal, provoca que el sistema de control de acceso sea mas difícil de comprender y de controlar.

A pesar de las limitaciones comentadas del modelo RBAC pensamos que puede ser usado como modelo de partida para la definición del control de acceso en un sistema colaborativo. En la siguiente figura se muestran los elementos básicos del modelo RBAC.

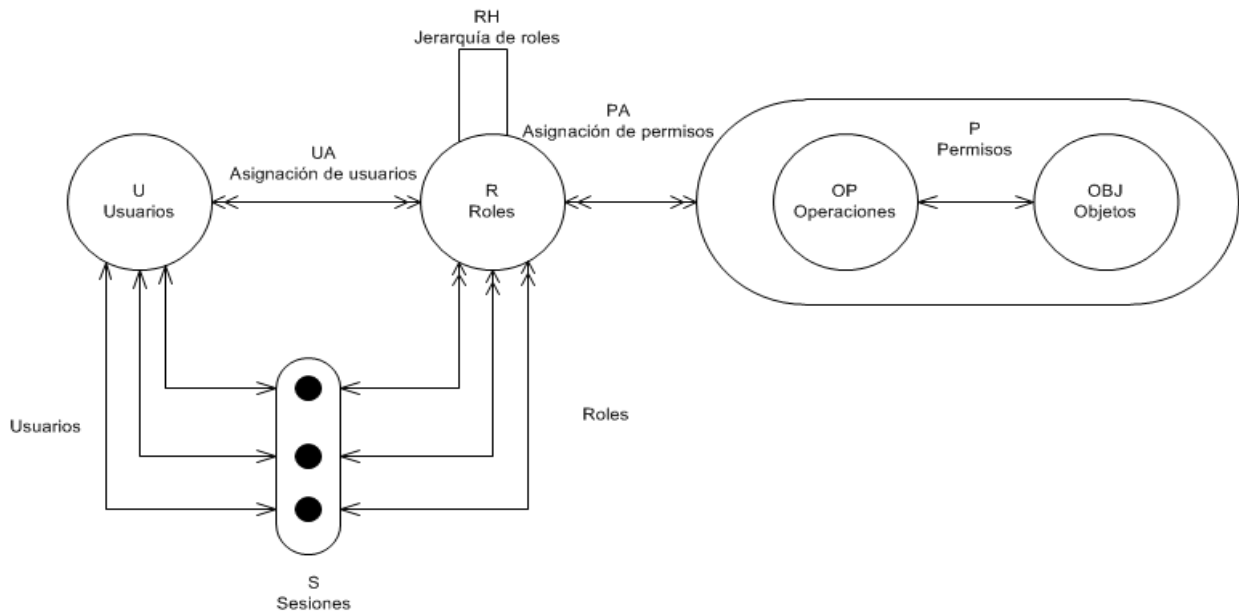


Figura 2.6. Elementos básicos del modelo RBAC.

Para la implementación del modelo, es necesario personificar los requisitos de las políticas, lo que hace necesario la utilización del lenguaje de definición XML y la especificación XML-Schema, ya probado con éxito en la seguridad de aplicaciones Web basadas en RBAC (Damian E. et al, 2002), (Bertino E. et al, 2001), (Hada S. and Kudo M., 2000). Así al modelado de las componentes RBAC usando XML la denominamos X-RBAC (Chandramouli R.,2000) e involucra la definición de las componentes del estándar NIST descritos anteriormente (usuarios, roles, permisos, sesiones) sujetos a las descripciones de cada entidad del modelo de aprendizaje basado en grupo interviniente, especificando nombre, atributos, relación, cardinalidad, restricciones y tipos de datos. La arquitectura de X-RBAC genera un repositorio XML para el almacenamiento de las componentes X-RBAC. La definición en XML de la componente “Rol”: La definición del tipo de dato “validRole” lista el conjunto de roles permisibles en el entorno de aprendizaje colaborativo, mientras que, el “roleLimit” es usado para especificar un número que coloca un máximo y un mínimo de usuarios que pueden ser asignados a ese rol.

Así el esquema general en XML define: Un esquema genérico del modelo de entorno de aprendizaje colaborativo, los datos de control de acceso al entorno (usuarios, rol,

asignación de rol a usuario, etc.), las restricciones estructurales (especificación de atributos obligatorios y opcionales, identificación de atributos cuyos valores deben ser únicos, restricciones de cardinalidad mostrando el número de veces que un componente del modelo RBAC contenga los datos del control de acceso en el documento XML, por ejemplo.), las restricciones elementales de dominio, la especificación y validación y los conflictos.

El lenguaje de definición XML y la especificación XML-schema han ido ganando gran aceptación como estándares para representar, intercambiar y presentar metadatos y modelos de contenido de una manera independiente de la plataforma. Sin embargo, estas están basadas en XML-DTDs, por lo que pierden capacidades que si proporciona el XML-schema, es por esto que proponemos la utilización de XML como lenguaje de definición para codificar los datos asociados al dominio y realizar la validación de estos contra el XML-schema, que representará el modelo de control de acceso de la organización. El modelo de control de acceso X-RBAC tiene la ventaja que puede ser reutilizado por cualquier aplicación o entorno que tenga como contexto un framework de características similares descritos en este documento, especialmente los referidos al aprendizaje colaborativo basado en grupo, actualmente el modelo se encuentra en etapa de implementación

2.1.5. MODELO SEM-HP

El modelo SEM-HP es un modelo SEMántico, Sistémico y Evolutivo para el desarrollo de Sistemas HiPermedia adaptativos, que pretende facilitar tanto la tarea de diseño del autor como la navegación de los usuarios del sistema hipermedia.

SEM-HP concibe un sistema hipermedia adaptativo (SHA) compuesto por cuatro subsistemas interrelacionados entre sí y en interacción: Memorización, Presentación, Navegación y Aprendizaje. El modelo SEM-HP (figura 2.7) proporciona un conjunto de acciones evolutivas para que éste pueda realizar sobre el sistema los cambios que considere oportunos, de forma que una acción evolutiva solo es ejecutada si satisface

un conjunto de restricciones necesarias para que el cambio sea consistente. Además, es posible que al modificar un elemento de alguno de los cuatro subsistemas se genere la necesidad de modificar otros elementos del propio subsistema (propagación interna del cambio) o incluso de otros subsistemas (propagación externa del cambio). Mediante el concepto de Metasistema, el modelo SEM-HP proporciona soporte a los cambios realizados por el autor y a la propagación automática generada por estos.

El Subsistema de Memorización almacena, estructura y mantiene el conocimiento que el sistema ofrece. El elemento principal de este subsistema es la estructura conceptual EC. La EC es una red semántica con dos tipos de nodos: Conceptos e ítems. Los conceptos son ideas etiquetadas y los ítems contienen trozos de información relativos a los conceptos.

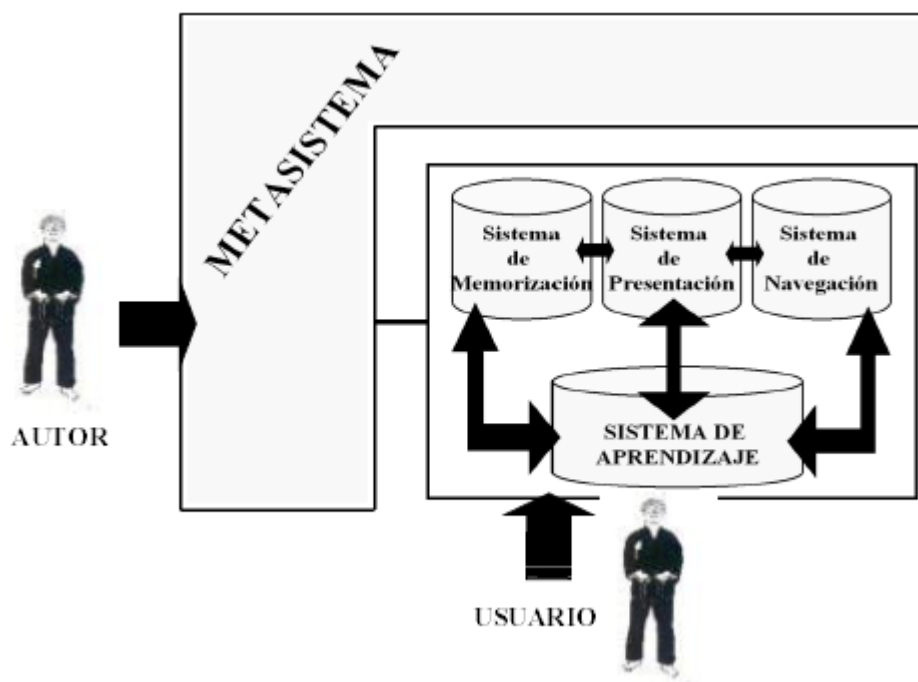


Figura 2.7. Estructura del modelo SEM-HP

Los enlaces de la EC son relaciones entre conceptos (asociación conceptual) o relaciones entre ítems y conceptos (asociación funcional).

El *Subsistema de Presentación* permite filtrar la EC creada en el subsistema de memorización. A través del filtrado, el autor selecciona un subconjunto de los conceptos, ítems y relaciones incluidas en la EC inicial.

El *Subsistema de Navegación* permite al autor añadir restricciones de orden sobre la EC de presentación. Estas restricciones establecen un orden parcial entre los trozos de información ofrecidos por el sistema, es decir entre los items de la EC.

El *Subsistema de Aprendizaje* es el encargado de realizar la adaptación del sistema hipertexto. Los elementos fundamentales de este subsistema son:

- Reglas de conocimiento: Establecen qué ítems debe conocer el usuario y con qué grado de conocimiento para poder alcanzar un determinado ítem. Por lo tanto, restringen la navegación del usuario en función de su conocimiento.
- Reglas de actualización: Cuando un usuario visita un ítem, estas reglas actualizan el grado de conocimiento que el usuario posee sobre ese y posiblemente otros items de la EC.
- Reglas de peso. Calculan el grado de conocimiento que el usuario posee sobre cada concepto de la EC de navegación, utilizando como base el conocimiento del usuario acerca de los items asociados al concepto.
- Modelo de usuario. Almacena información sobre el conocimiento, preferencias e intereses del usuario. Usamos una red de Petri para formalizar el proceso mediante el cual el sistema calcula el conocimiento del usuario. Existe un nodo lugar en la red de Petri para cada ítem de la EC de navegación. La selección del usuario dispara una transición, siempre que se cumplan las reglas de conocimiento asociadas, y ejecuta la regla de actualización ligada al arco de salida de la transición.

El subsistema de aprendizaje utiliza distintas técnicas y métodos para realizar la adaptación de la EC de navegación. Las técnicas de adaptación empleadas en SEM-HP permiten ocultar enlaces que llevarían al usuario a items para cuya lectura no está preparado (no satisface las reglas de conocimiento). Además, los items y conceptos de la EC son anotados indicando el grado de conocimiento que el usuario posee sobre ellos o si han sido visitados anteriormente por el usuario.

Los Sistemas Hipertexto Adaptativos (SHA) desarrollados de acuerdo al modelo SEM-HP calculan el conocimiento de cada usuario individual para, en función de este

conocimiento, realizar una determinada adaptación sobre el sistema hipermedia. Sin embargo, en entornos de aprendizaje colaborativo, resulta interesante calcular el conocimiento compartido por el grupo de usuarios. En este tipo de sistemas podría ser útil restringir la navegación de un usuario en función del conocimiento de otros miembros del grupo de trabajo, y no sólo de su propio conocimiento. La adaptación permite mejorar el proceso colaborativo del grupo de forma muy similar a como mejorar el proceso de navegación individual de un único usuario. Por ejemplo, utilizando técnicas de adaptación el sistema puede mostrar a un usuario qué documentos han sido leídos por otro miembro del grupo, cual es el conocimiento medio del grupo sobre un determinado concepto o qué documentos no han sido leídos por ningún miembro. Para que un SHA pueda soportar un proceso de trabajo colaborativo es necesaria una estructura de navegación concurrente que permita al sistema conocer en todo momento la situación actual de todos los participantes que están colaborando simultáneamente. Esta estructura concurrente podría ser modelada mediante redes de Petri similares a las utilizadas en el subsistema de aprendizaje.

El subsistema de aprendizaje del modelo SEM-HP infiere, a partir de la navegación, la EC que los usuarios tienen en su mente para permitir al autor mejorar la EC original, adecuándola a los patrones de navegación comunes a la mayoría de los usuarios del sistema. Esta técnica de adaptación podría ser utilizada en entornos colaborativos para beneficiar a nuevos miembros del grupo, ya que a menudo a los usuarios que buscan información dentro de un dominio que le es desconocido, les resulta más beneficioso el conocimiento de otros usuarios ya familiarizados con el dominio que una interfaz personalizada.

2.1.6. MODELO KAOS

La metodología KAOS (Knowledge Acquisition in autoMated Specification) tiene como finalidad el apoyo del proceso entero de elaboración de requisitos, incluyendo las metas de alto nivel que deben satisfacer los requisitos y los objetos y operaciones

que se asignen a los agentes del sistema, que abarca tanto el software como su entorno. Así, se tienen en cuenta las preguntas “¿Por qué?”, “¿Quién?” y “¿Cuándo?” además de la habitual “¿Qué?” que tratan las técnicas de especificación comunes. La metodología suministra un lenguaje de especificación multiparadigma, un método de elaboración dirigido por metas y un meta-nivel para guiar el proceso de elaboración y representación, y para determinar el contenido y estructura de los documentos de requisitos.

El lenguaje combina una semántica de redes para el modelado conceptual de metas, restricciones, agentes, objetos y operaciones del sistema. Para la especificación de metas, restricciones y objetos se puede usar el lenguaje natural o lógica temporal, y especificaciones basadas en estados para la especificación de operaciones. Además, proporciona una separación entre los requisitos y las descripciones del dominio.

El método consiste básicamente en la identificación y refinamiento de metas hasta que se asignan a agentes individuales, identificación progresiva de objetos y operaciones a partir de las metas, derivación de requisitos sobre los objetos y operaciones para cumplir las restricciones, y asignación de restricciones, objetos y operaciones a los agentes.

El meta-nivel proporciona abstracciones independientes del dominio en términos de los conceptos específicos del dominio que se obtienen. Está compuesto de meta-conceptos (como meta, agente, relación, operación, etc.), meta-relaciones (refinamiento, responsabilidad, etc.), meta-atributos de los meta-conceptos y meta-relaciones, y meta-restricciones sobre los meta-conceptos y las meta-relaciones. A partir del meta-nivel se puede obtener un modelo del dominio compuesto por instancias específicas de meta-conceptos y meta-relaciones. Así, los componentes de un modelo de requisitos se adquieren como instancias específicas de los elementos del meta-nivel. Los conceptos de este nivel pueden a su vez instanciarse. Esta relación entre niveles (figura 2.8).

Un modelo de KAOS está compuesto por varios submodelos relacionados por medio de reglas de consistencia. Estas reglas de consistencia proporcionan trazabilidad entre

el contenido de los distintos submodelos, de manera que uno se crea en base a lo representado en otros. Estos submodelos son cuatro: El modelo de metas, el de responsabilidad, el de objetos y el de operaciones.

Modelo de metas

El modelo de metas permite al analista estructurar las metas en un grafo dirigido acíclico. De este modo, cada meta del modelo (a excepción de la meta raíz) se justifica por al menos otra meta que explica el porqué de su introducción y se refina como una colección de submetas que describen cómo se puede alcanzar la meta (a excepción de las metas hoja).

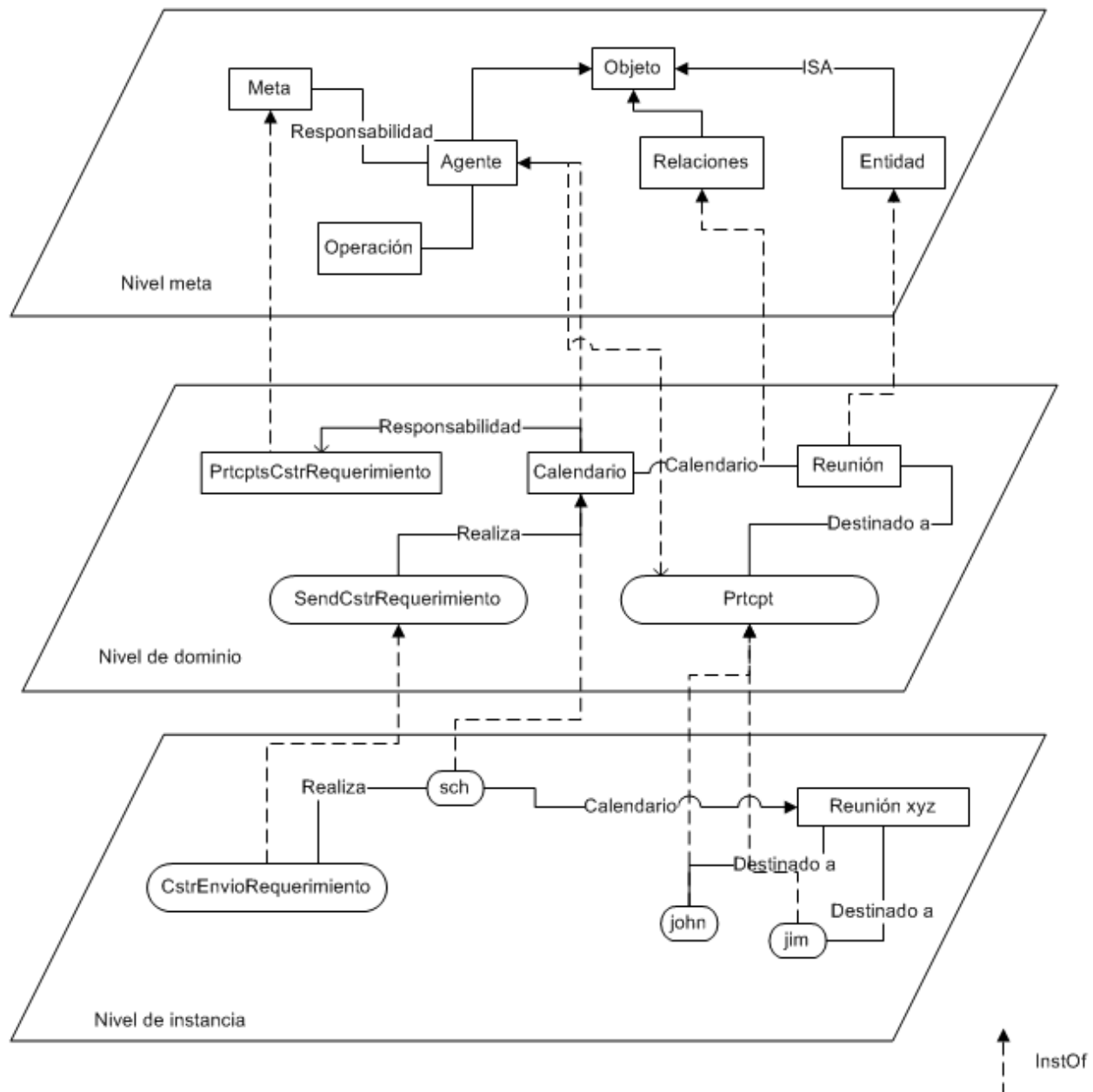


Figura 2.8. Niveles meta, de dominio y de instancia de KAOS

Las metas en la parte superior del grafo representan metas estratégicas o de negocio, mientras que las inferiores son requisitos del sistema. En el modelo se pueden representar las características del sistema a desarrollar, las propiedades del dominio y condiciones que se espera que se cumplan, denominadas expectativas.

Las metas se definen como objetivos del sistema que habitualmente se satisfacen a través de la cooperación de varios agentes. Cada meta tiene un nombre y una

definición. Respecto a su tipo, existen distintas clases de metas que se clasifican en base al patrón que caracterizan.

Estas clases son metas de mantenimiento, si requieren que alguna propiedad siempre se cumpla; de logro, si requieren que alguna propiedad se cumpla eventualmente; de optimización de algún criterio; de cese de alguna propiedad en algún instante; débiles si no tienen un criterio claro de satisfacción (concepto similar al de i^* y NFR); y para evitar que alguna propiedad se cumpla. Esta taxonomía se extiende con las categorías de metas, apareciendo así metas de satisfacción, que son metas de logro relativas a deseos de los agentes que se cumplen; de seguridad, que son metas de mantenimiento relativas a estados que se deben evitar; de protección, que son metas de mantenimiento relativas a amenazas del sistema a evitar; de información, que son metas de logro relativas a agentes que obtienen una información de su entorno; y de precisión, que son metas de mantenimiento relativas a la precisión de las creencias de un agente sobre su entorno. Además, las categorías se organizan en una jerarquía de especialización, como, por ejemplo, las metas de protección que se especializan en confidencialidad, disponibilidad, autenticación, etc.

Existen enlaces de refinamiento de metas que relacionan una meta con un conjunto de submetas. Este conjunto refina a la meta padre si la satisfacción de todas las submetas es suficiente y completo, es decir, conlleva el cumplimiento de la meta padre, es mínimo, y es consistente. Una meta puede refinarse en conjuntos alternativos de metas, que puede denotar alternativas en el diseño del sistema. Durante este proceso de refinamiento pueden aparecer conflictos entre metas si no se pueden cumplir a la vez u obstáculos que violen una meta, un requisito o una expectativa, hechos que deben ser tratados.

En el modelo de metas (figura 2.9) se introducen agentes a los que se les asignan metas, de manera que sean responsables de ellas. En este sentido, se considera que una meta pasa a ser un requisito en el momento en el que es asignada a un único agente. En la siguiente figura se detalla un ejemplo para el modelo de ambulancia.

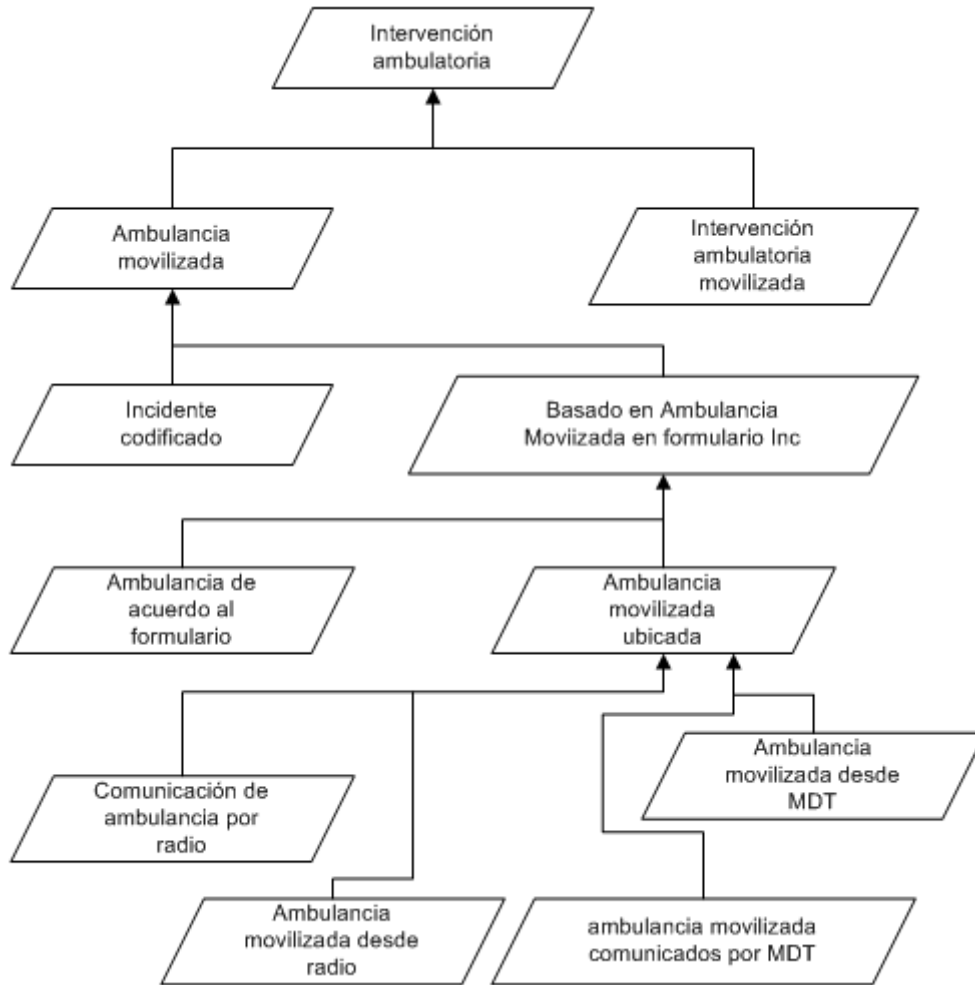


Figura 2.9. Modelo de metas de KAOS

Un requisito se puede refinar en otros, pero implícitamente bajo la responsabilidad del agente del requisito padre. Esta relación de responsabilidad entre agentes y metas conduce a la aparición de dos criterios de completitud para el modelo. El primero establece que un modelo de metas es completo respecto una relación de refinamiento si y sólo si cada meta hoja es una expectativa, una propiedad del dominio o su requisito, mientras que en el segundo se considera que el modelo es completo si y sólo si cada requisito está bajo la responsabilidad de uno y sólo un agente.

Modelo de objetos

En el modelo de objetos (figura 2.10) se definen y documentan los conceptos del dominio de la aplicación en una mina que son relevantes para los requisitos y para proporcionar restricciones estáticas que cumplirán éstos. Se pueden encontrar objetos

tanto pertenecientes al dominio de las partes interesadas como otros introducidos como apoyo de los requisitos y restricciones.

Los tipos de objetos que aparecen en el modelo pueden ser entidades, que representan objetos independientes y pasivos, agentes, que representan objetos independientes y activos, y asociaciones, que son objetos dependientes y pasivos. Con la calificación de dependencia se establece si el objeto necesita referirse a otros en el modelo, y con actividad si el objeto realiza operaciones.

La identificación de objetos se realiza durante el proceso de definición de metas, ya que se suele hacer referencia a objetos que deben ser modelados y documentados. También se pueden identificar estudiando los requisitos y descubriendo componentes del sistema que sean necesarios para su logro. Cuando se identifica un objeto, éste debe definirse en el modelo de objetos y relacionarlo con los conceptos existentes, mediante asociaciones o relaciones de herencia.

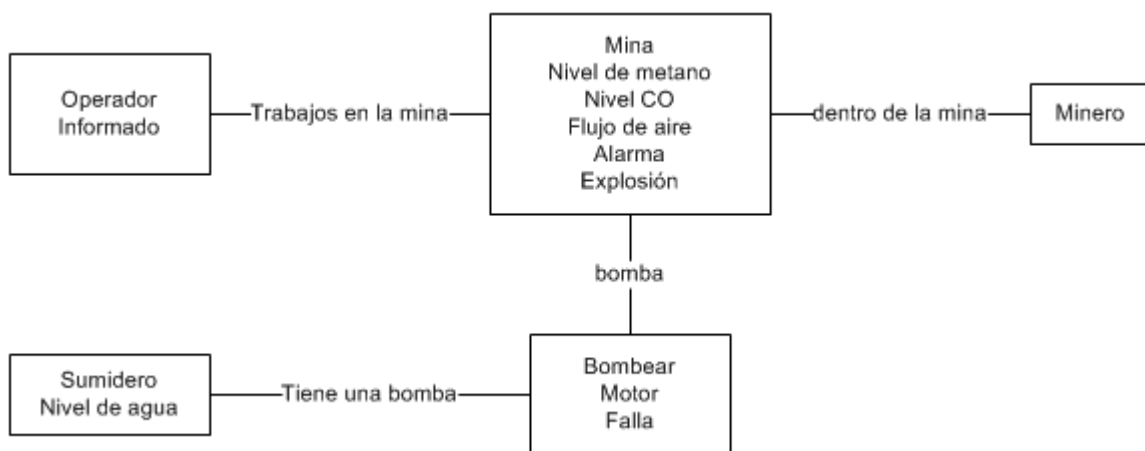


Figura 2.10. Modelo de objetos de KAOS

Modelo de responsabilidad

El modelo de responsabilidad (figura 2.11) describe a un ascensor con los requisitos y expectativas de los que es responsable un agente. El desarrollo de este modelo es muy sencillo, ya que sólo hay que revisar el modelo de metas y extraer para cada agente los elementos que caen bajo su responsabilidad.

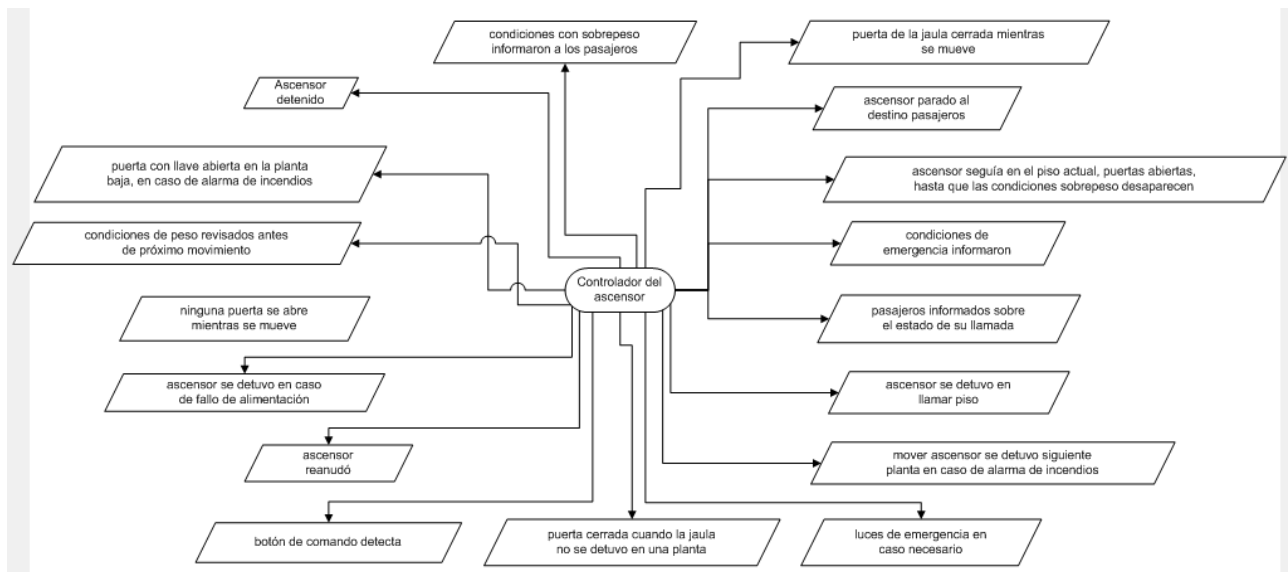


Figura 2.11. Modelo de responsabilidad de KAOS

Modelo de operaciones

Este modelo describe todos los comportamientos que los agentes necesitan para cumplir sus requisitos (figura 2.12). Los comportamientos se expresan en términos de operaciones desarrolladas por los agentes y que tienen un efecto sobre los objetos, como su creación, cambio de estado o activación de sus operaciones.

Las operaciones tienen entrada, salida, precondiciones y postcondiciones. Además, en el modelo pueden aparecer eventos que provoquen el inicio o fin de una operación.

Operación Reunión de Plan

Input reunión {arg m}

Destinado a

Restricción

Salida reunión / {planificada, fecha}

DomPre m. planificado

DompPost m. planificado

ReqPostFor Mantener [fecha de la reunión Conveniente]

(p: participante): Recomendada (p, m) -> restricción [p, m] exclset.

Figura 2.12. Plantilla textual de una operación de KAOS

2.1.7. MODELO CTT

La notación CTT (ConcurTaskTrees) desarrollada por Paternò, permite representar relaciones temporales existentes entre las tareas y entre los usuarios necesarios para llevarlas a cabo. Fue desarrollada usando la notación LOTOS, notación formal que supuso un cambio en el desarrollo de interfaces de usuario, pues permitió a los desarrolladores describir el comportamiento y los estados modificados. Sin embargo, LOTOS presentaba algunas limitaciones, ya que había una necesidad de añadir nuevas operaciones para representar de una forma más amplia el comportamiento dinámico en la interacción persona-ordenador y permitir aportar información adicional para su uso en el análisis de modelo de tareas. Además, la sintaxis de LOTOS, puede generar expresiones complicadas incluso cuando el comportamiento descrito es simple. Por todo esto, surgió la necesidad de desarrollar una nueva notación: ConcurTaskTrees. Esta notación puede ser usada para describir el comportamiento de aplicaciones reales con unas dimensiones de mayor tamaño. Este es el problema de muchas otras notaciones en las que se observan a menudo limitaciones cuando se trata del estudio de casos reales. CTT aporta un conjunto muy amplio de operadores gracias a la notación LOTOS.

Estos permiten describir las relaciones temporales entre las tareas: Concurrencia, opcionalidad, activación, desactivación e interrupción y además, para cada tarea proporciona información adicional como por ejemplo el tipo de la tarea.

Proporciona una representación gráfica en forma de árbol y muestra la descomposición jerárquica de las tareas del sistema. Una estructura jerárquica es algo muy intuitivo, de hecho, cuando la gente tiene que resolver un problema, generalmente tiende a descomponerlo en pequeños problemas manteniendo las relaciones entre varias partes de la solución. La estructura jerárquica de la notación CTT tiene una serie de ventajas: Proporciona una amplia gama de granularidad que permite que una estructura de tareas tanto grandes como pequeñas pueda ser reutilizadas, lo cual permite beneficiarse de estructuras de tareas ya realizadas para

definir tanto un alto como bajo nivel. Proporciona una notación concurrente, lo que hace que se pueda definir un amplio abanico de posibles relaciones entre tareas.

También cabe destacar que en la notación CTT, a diferencia de otros tipos de modelado, dos tareas pueden sincronizarse. Esto es un hecho positivo, por ejemplo cuando se necesita intercambiar información debido a que la información de salida de una tarea es la información de entrada de la tarea siguiente.

Esto hace a la notación CTT una técnica de modelado flexible, pues permite representar actividades concurrentes e iterativas donde un objetivo puede ser alcanzado realizando diferentes tareas, teniendo la posibilidad de realizar colaboraciones entre múltiples usuarios, así como una notación compacta, comprensible y representativa. El éxito clave de esta notación es la capacidad de proveer mucha información, por lo que las tareas a realizar por los usuarios serán muy intuitivas. De este modo, la notación CTT puede ser utilizada por personas que no tengan grandes conocimientos en informática como por ejemplo los usuarios de la propia aplicación.

La realización de las tareas se indica mediante la utilización de iconos gráficos. En función del actor que las lleve a cabo es posible clasificarlas en cuatro tipos.

Tareas del usuario:

Son las tareas realizadas completamente por el usuario. Pueden ser tareas cognitivas o físicas que no interactúan con el sistema. Describen procesos realizados por el usuario usando la información que recibe del entorno, por ejemplo, seleccionar dentro de un conjunto de información la que se necesita en un instante determinado para la realización de otra tarea.

- Tareas de Planificación: Secuencia de actividades a realizar por el usuario.
- Tareas de Comparación: Cuando el usuario tiene que evaluar una determinada información, por ejemplo, comparar cantidades o identificar el máximo valor de una lista de valores.

- Tareas de Solución del problema: Si el usuario tiene que encontrar la solución a un problema, por ejemplo, si hay un conflicto entre la ruta de dos vuelos y el regulador tiene que encontrar modos de evitar una colisión.

Tareas de la aplicación:

Son tareas activadas y realizadas por la propia aplicación. Permiten obtener información interna del sistema o producir información para el usuario. Por ejemplo, una tarea que presente los resultados obtenidos de una consulta en una base de datos.

- Descripción: Las aplicaciones muestran un resumen del juego de datos que no proveen los datos elementales.
- Comparación: Su propósito es ayudar al usuario a comparar valores de datos.
- Localización: Las aplicaciones dan información detallada sobre los datos, lo que permite a los usuarios encontrar rápidamente la información deseada.

Tareas de interacción:

Son tareas que realiza el usuario interactuando con la aplicación por medio de alguna técnica de interacción.

- Tareas de Selección: Son muy comunes en las aplicaciones. Por ejemplo, tareas donde el usuario debe seleccionar uno o más artículos de entre varios. La tarea se puede clasificar dependiendo de si se selecciona un único artículo o varios.
- Tareas de Edición: Tareas que permiten a los usuarios especificar datos de entrada y esta información puede ser modificada antes de ser definitivamente enviada a la aplicación.
- Tareas de Control: El propósito de estas tareas es generar un evento indicando cuando algo debe ocurrir.

Tareas abstractas:

Modelan tareas cuya única finalidad es servir de agrupación de otras tareas y se pueden descomponer en otras más sencillas.

Tareas cooperativas:

Son tareas que requieren la participación y cooperación de distintos usuarios para realizar actividades.

Como se ha expuesto anteriormente el conjunto de operadores de la notación CTT es una ampliación del utilizado por la notación LOTOS. Esta notación es concurrente y se utiliza para poder especificar interfaces de usuario, pues permite describir comportamientos manejados por eventos y modificaciones de estado.

A continuación se detalla una tabla que describe los operadores temporales en CTT.

Operador temporal	Notación	Descripción
Entrelazado (Concurrencia independiente)	$T1 \parallel T2$	Las acciones de las dos tareas pueden realizarse en cualquier orden.
Elección	$T1 [] T2$	Selección alternativa entre dos tareas. Una vez que se esta realizando una de ellas la otra no esta disponible hasta que termine la que esta activa.
Sincronización (Concurrencia con intercambio de información)	$T1 [] T2$	Las dos tareas tienen que sincronizarse en alguna de sus acciones para intercambiar información.
Desactivación	$T1 [> T2$	Desactivar. La primera tarea es desactivada cuando comienza la ejecución de la segunda.
Activar (enabling)	$T1 >> T2$	Cuando termina la T1 se activa la T2. Las dos tareas se realizan de forma secuencial.
Activar con paso de información	$T1 [] >> T2$	Cuando termina T1 genera algún valor que se pasa a T2 antes de ser activada.
Iteración	$T1 *$	La tarea T1 se realiza de forma repetitiva. Se estará realizando hasta que otra tarea la desactive.
Independencia de orden	$T1 = T2$	Ambas tareas pueden ser realizadas, pero una vez comenzada una debe de finalizar antes de comenzar con la otra.
Suspend / Resumen	$T1 > T2$	T2 tiene la posibilidad de interrumpir a T1 que podrá ser retomada cuando aquella finalice.
Tarea opcional	$[T1]$	La realización de la tarea es opcional.

Tabla 2.5. Operadores temporales definidos en la notación CTT

2.2. CONSIDERACIONES SOBRE LOS DISTINTOS MODELOS

En este capítulo detallamos las particularidades más importantes de los modelos descritos en la sección 2.2.1, las características más relevantes del modelo AMENITIES (sección 2.2.1.1), modelo TOUCHE (sección 2.2.1.2), modelo CIAM (sección 2.2.1.3), modelo RBAC (sección 2.2.1.4), modelo SEM-HP (sección 2.2.1.5), modelo KAOS (sección 2.2.1.6), modelo CTT (sección 2.2.1.7), un cuadro comparativo por modelos y características (sección 2.3) y las valoraciones sobre el cuadro comparativo (sección 2.4).

2.2.1. CARACTERISTICAS CONSIDERADAS EN LOS MODELOS

2.2.1.1. AMENITIES

La arquitectura de este modelo se basa en componentes, funcionalidad, comportamiento y despliegue, mediante vistas especificadas haciendo uso del lenguaje UML:

- Los diagramas de componentes permiten representar la descomposición de un sistema en subsistemas más pequeños.
- Los diagramas de clases e interfaces asociados con subsistemas permiten representar la estructura funcional (vista estática) de los subsistemas dentro de un diseño arquitectónico, definiendo conexiones entre ellos en base a relaciones de uso.
- Los diagramas de colaboración permiten describir el comportamiento (vista dinámica) de los subsistemas que colaboran en base a la arquitectura funcional, descrita anteriormente.
- Los diagramas de componentes y despliegue permiten describir la distribución de los subsistemas en los nodos del sistema distribuido.

Con este diseño arquitectónico es posible incrementar la independencia, modularidad, integración y reusabilidad del software. También permite proporcionar una plataforma específica de servicios de alto nivel para la gestión de metainformación (registro de información acerca de usuarios, uso de mecanismos para reforzar la conciencia del grupo, etc.) en esta clase de sistemas.

KnowCat es un sistema distribuido que, sin necesidad de supervisión, permite la creación incremental de conocimiento estructurado (figura 2.13).

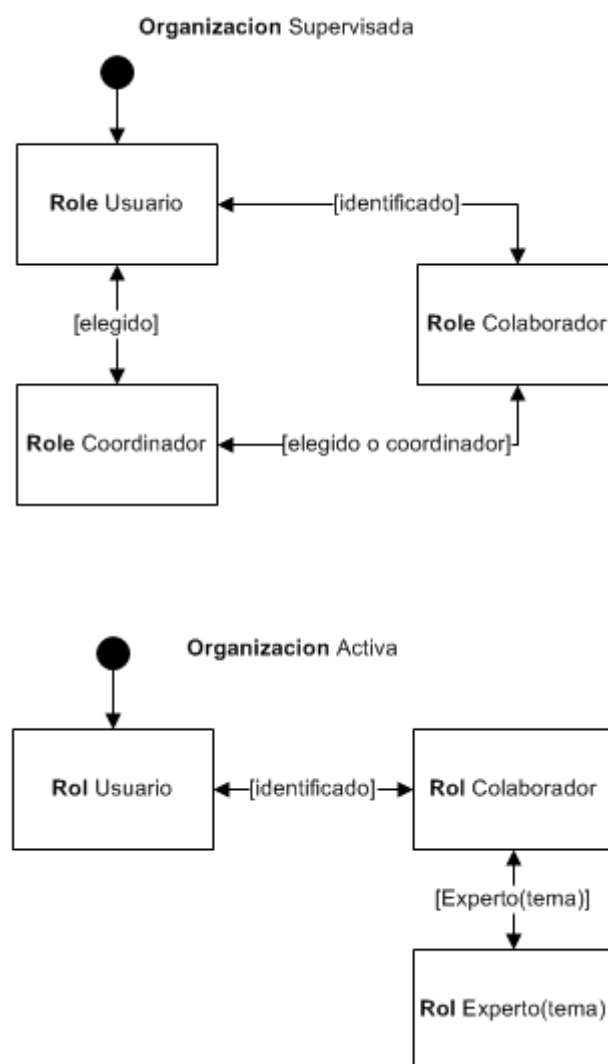


Figura 2.13 Detalle de la vista de la organización en KnowCat; modo supervisado y modo activo

Las dos organizaciones varían en cuanto a los roles y estructura. La organización del trabajo se realiza de modo diferente en función del objetivo a lograr. En el primer

caso, se busca una organización coherente del árbol de conocimiento y la obtención de una masa crítica de participantes. En el segundo caso, se pretende obtener aportaciones de contenidos de calidad y contrastadas sobre los temas del árbol de conocimiento.

2.2.1.2. TOUCHÉ

Este modelo de proceso y una metodología para el desarrollo de interfaces de usuario para 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 realizó un estudio de los sistemas CSCW que, ha dado como resultado un método para la clasificación de estas aplicaciones.

Este modelo muestra las diferentes etapas:

- **Requisitos**, donde se identifica la estructura organizativa del sistema y los principales actores. Identificando también en esta etapa los requisitos funcionales más importantes.
- **Análisis**, se realiza la identificación y descripción de roles. Mediante la introducción de los actores y la introducción de sus roles la herramienta con la que se trabaja genera automáticamente en esta etapa un diagrama con la estructura organizativa del sistema.
- **Diseño**, en esta etapa se trabajó más la parte de la interfaz de usuario. Obteniendo los diagramas de navegación (ACID, diagrama de interacción de contenedores abstractos) y los diagramas de presentación (AUID, diagrama de interfaces de usuario abstractas).

Este modelo enriquece el modelo conceptual para dar soporte a estas nuevas características propias de las aplicaciones groupware (figura 2.14):

- **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.

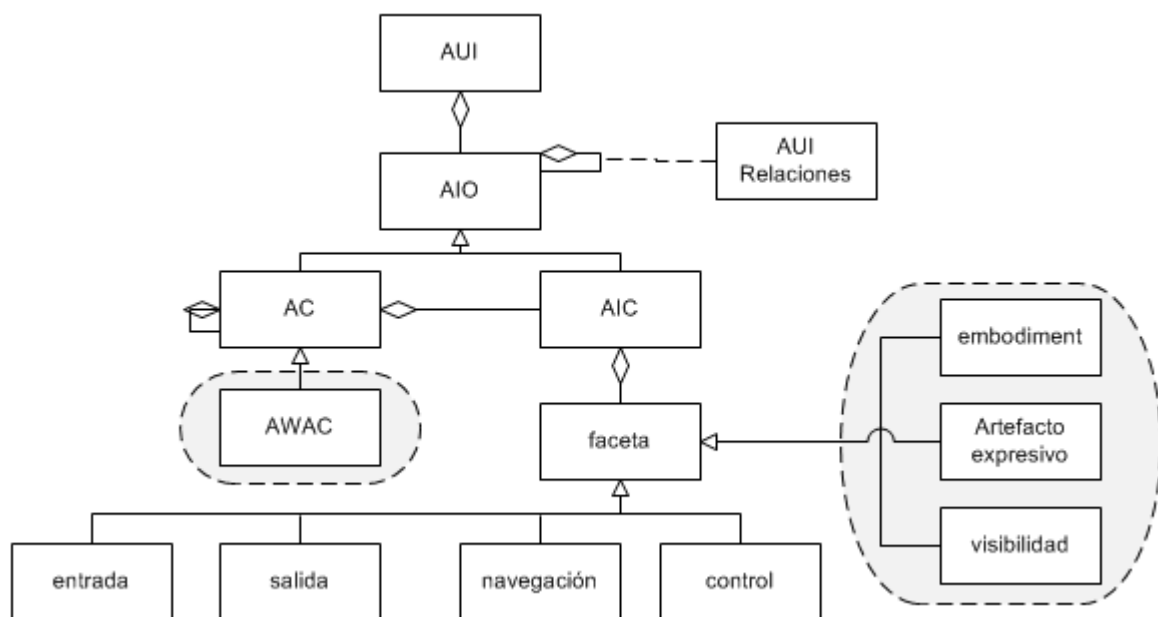


Figura 2.14 Modelo conceptual basado en el modelo conceptual para interfaces de usuarios abstractas

Los componentes propuestos en la API siguen el modelo propuesto en TOUCHE, ya que se han desarrollado CIOs en base a los AIOs caracterizados con las facetas embodiment, expressive artifact y visibility. Los CIOs son replicados permitiendo que varios usuarios estén interactuando con el componente de forma simultánea manteniendo un estado consistente global. Un componente replicado normalmente mantendrá un estado común con cada una de las instancias de dicho componente que

se estén ejecutando en un determinado momento. Para ello, esos componentes que mantienen un estado común, deberán sincronizar su estado con el de los demás componentes cuando se inician.

Se ha creado un conjunto de componentes, como extensiones de componentes estándares, con un comportamiento distribuido. A la hora de inicializar el CIOs DILista existe la posibilidad de elegir entre modo replicado o local. En el modo replicado añadir y eliminar elementos de la lista es un comportamiento distribuido, mientras que en el modo local no se notifica a ninguna instancia del componente.

2.2.1.3. CIAM

Este modelo permite el desarrollo de interfaces de usuario en aplicaciones groupware mediante la adopción de distintos puntos de vista a la hora de abordar la creación de modelos conceptuales para este tipo de sistemas. Las primeras etapas abordan un modelado centrado en el grupo, pasando en fases posteriores a un modelo más centrado en el proceso (cooperativo, colaborativo y de coordinación).

Acercándose, a medida que bajan en el nivel de abstracción, hacia un modelado centrado en el usuario, modelando el diálogo que se da entre un usuario individual y la aplicación.

Este modelo adopta diferentes puntos de vista para crear modelos de sistemas groupware interactivos, y propone una notación específica denominada CIAM, la cual promueve el modelado de la colaboración, la comunicación y la coordinación. CIAM soporta adecuadamente el modelado de la colaboración, pero no permite modelar atributos o cualidades del sistema como son la funcionalidad, del modo en que lo hace UML. Por otro lado, ni UML ni RUP están pensados para un diseño de interfaces de sistemas interactivos con características de usabilidad. Cuando en el desarrollo de software se tienen en cuenta varios aspectos, tales como la interacción con el usuario, la colaboración y la funcionalidad, no es fácil identificar la relación existente entre clases u objetos contenidos en los modelos que especifican dichos

tareas que se desempeñan. En el estudio de la estructura, además de los objetos del sistema se tiene en cuenta quienes son y de qué manera participan, como se organizan y que roles desempeñan. El estudio del comportamiento se aborda describiendo la funcionalidad desde un punto de vista CSCW, abordando sus características típicas, así como sus características espaciotemporales, muy importante en estos sistemas.

2.2.1.4. RBAC

Este modelo asigna roles a los usuarios, incluye el concepto de sesión que permite la activación y desactivación selectiva de roles, posibilitando que un usuario ejerza varios roles al mismo tiempo. Este modelo define cinco conjuntos de elementos básicos: Usuarios, roles, objetos, operaciones y permisos (figura 2.16).

El propósito de cualquier mecanismo de control de acceso es proteger a los recursos del sistema. Para que un usuario pueda ejercer los permisos para los permisos que está autorizado es necesario que este active los roles. Este modelo especifica los requerimientos funcionales de los comandos utilizados para interactuar con el sistema.

Este estándar contempla dos tipos de jerarquías de roles, generales y limitadas. Define los requerimientos funcionales de los comandos de gestión e inspección de la jerarquía y también especifica los cambios en los requerimientos de algunos comandos de Core RBAC que deben contemplar la presencia de las mismas.

Su notación estándar (\wedge , \vee , \rightarrow , \neg , \exists), permite heredar los roles formando la relación jerárquica en un orden parcial sobre el conjunto de roles del sistema.

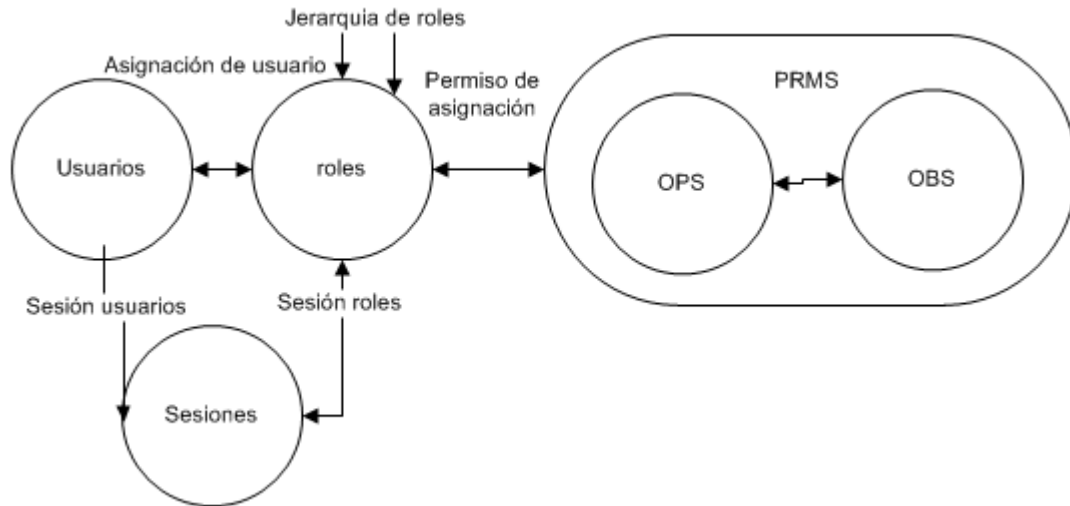


Figura 2.16 Elementos del modelo RBAC

Para simplificar la manipulación de esta jerarquía el estado del sistema compuesto por dos relaciones: Una de herencia inmediata o directa y su respectiva clausura reflexo-transitiva definida inductivamente por los siguientes constructores

(\ll) : $\text{Role} \rightarrow \text{Role} \rightarrow \text{Prop}$

$\text{rt_refl} = (/ x \leq x)$

$\text{rt_step} = (x \ll y / x \leq y)$

$\text{rt_trans} = ((x \leq y \quad y \leq z) / x \leq z)$

Cada sesión es un mapeo entre un usuario y sus roles activos. El estado de las sesiones del sistema está dividido en dos partes. La primera es un conjunto de identificadores válidos de la sesión llamado *session*. La segunda parte es una función del identificador de sesión al usuario dueño de la misma y sus roles activos, en términos formales.

La semántica de los comandos esta especificada mediante pre y pos condiciones. Para cada comando su precondition es un predicado sobre el estado y su postcondition es un predicado que relaciona el estado anterior y el posterior a la ejecución del mismo y la respuesta correspondiente.

La jerarquía de roles (figura 2.17) está definida como un orden parcial \leq en el conjunto de roles. Es natural utilizar dicha construcción ya que la transitividad y

antisimetría modelan de manera precisa lo que uno espera de una relación de herencia; no es necesario hacer explícita cada una de las relaciones.

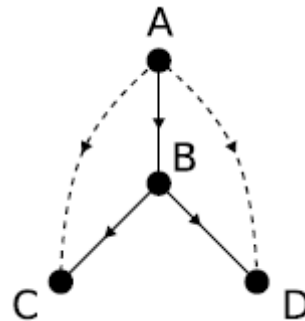


Figura 2.17 Jerarquía de ejemplo.

El estándar de NSIT persigue un modelo RBAC de referencia que unifique las diferentes implementaciones comerciales y de investigación. Este utiliza el lenguaje de especificación Z para formalizar los requisitos funcionales de las operaciones administrativas, de las consultas y el manejo de sesiones a nivel de sistema.

Los requerimientos funcionales del modelo describen la semántica de los comandos de manera parcial, es decir solo especifican el comportamiento del sistema cuando se cumplen las precondiciones de los mismos. En el cálculo de construcciones inductivas todas las funciones describen cómputos que terminan, o en otros términos, son totales. Por lo tanto fue necesario completar la especificación, indicando que sucede cuando las precondiciones no se cumplen. Para ello se definen códigos de error y una relación que los vincula con el estado del sistema.

Contar con un modelo en el CCI permitió razonar de manera abstracta sobre su corrección. Por un lado se verificó que la semántica de los comandos conserva invariante la validez del estado, lo cual garantiza que el sistema no puede transicionar a un estado que no represente un posible estado RBAC.

2.2.1.5. SEM-HP

La arquitectura del modelo establece una doble separación: A nivel vertical para distinguir cuatro aspectos funcionales y a nivel horizontal para distinguir dos niveles de abstracción.

La división vertical estructura de un SHA en cuatro sistemas interrelacionados entre si y en la interacción: Memorización, presentación, navegación y aprendizaje.

- El sistema de memorización (MS) define semánticamente, estructura y mantiene los conceptos y la información que ofrece el SHA.
- El sistema de presentación (PS) selecciona subconjuntos de los elementos ofrecidos por el MS, a los que se llama presentaciones.
- El sistema de navegación (NS) define las rutas o caminos que el usuario puede seguir cuando navega por cada presentación.
- El sistema de aprendizaje (LS) adapta la presentación y la navegación de los contenidos a las características específicas de cada usuario (adaptación al usuario). También analiza el compartimiento navegacional del grupo de usuarios con objeto de sugerir ajustes evolutivos en los otros subsistemas.

En Parents 1995, la división horizontal de la arquitectura de este modelo distingue dos niveles de abstracción: Sistema y metasistema. El metasistema proporciona al autor un conjunto de acciones evolutivas para que este pueda realizar sobre cada sistema los cambios que considere oportunos, garantizándose que tras estos se mantiene la consistencia del sistema global. También propone un proceso de desarrollo con cuatro fases iterativas, cada una de las cuales corresponde con uno de los cuatro sistemas mencionados.

JSEM-HP es una herramienta de autor para el desarrollo de sistemas hipermedia. Ha sido implementada en Java por sus portabilidad, la disponibilidad bibliotecas y por las facilidades para ser usado en Internet.

La implementación de la herramienta JSEM-HP nos ha permitido constatar los beneficios que conlleva un modelo que se basa en el software evolutivo y hasta qué

punto es viable su construcción y que mecanismos son clave en un enfoque evolutivo. Para ello dos mecanismos fundamentales han sido:

- Una representación semántica explícita
- Una implementación basada en la separación de aspectos.

Durante la navegación (figura 2.18), la estructura conceptual que se ofrece al usuario se adapta de manera dinámica al conocimiento del usuario, por lo que no tendrá acceso a información que sea incapaz de comprender, y tendrá conciencia en todo momento de sus procesos de aprendizaje (modo de navegación por conocimiento). Aunque si lo desea también puede elegir otros modos de navegación con dosis inferiores de adaptación (modo de navegación por relación conceptual y modo de navegación por conceptos), e incluso un modo libre de navegación (modo de navegación tradicional) que sería similar a navegar en la Web, pero usando una red semántica para seleccionar los items de información en lugar de únicamente enlaces embebidos en el texto.

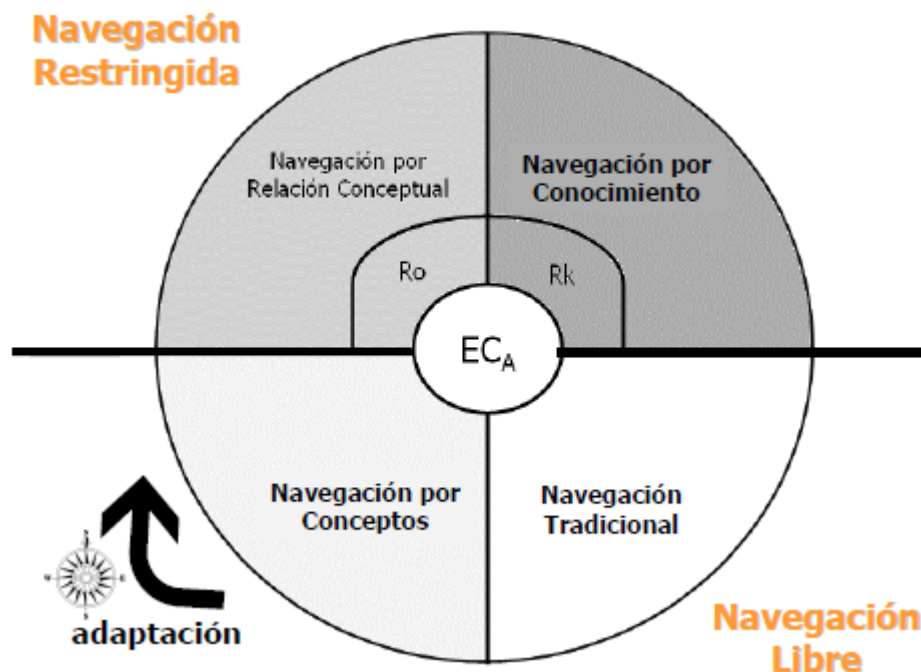


Figura 2.18 Modos de navegación permitidos en SEM-HP

2.2.1.6. KAOS

Este modelo da soporte a los procesos de elicitación de requisitos desde los objetivos a alto nivel (llevados a cabo por el sistema compuesto) hasta los objetivos operacionales. Kaos es influenciada por GRAM. Este método presenta un asesoramiento para la identificación de heurísticas y un conjunto de cuestiones a cumplimentar para poder guiar todo este proceso.

KAOS permite construir modelos de requisitos a partir de las metas organizacionales. Este modelo contribuye a la demostración de los requisitos que corresponden con las metas del futuro sistema. El principal inconveniente es que no proporciona ningún mecanismo para representar la estructura de la organizacional y como consecuencia de ello no permite efectuar un análisis de reingeniería de procesos. Trata de resolver el problema de determinar el conocimiento relevante sobre el dominio, las necesidades del cliente y su operacionalización en especificaciones de servicios y restricciones. Para ello intenta aprovechar el conocimiento sobre la captura de requisitos y emplear técnicas que automaticen en lo posible el proceso.

Los requisitos se adquieren como instancias específicas para el dominio, de un metamodelo conceptual. Su proceso de captura de requisitos involucra tres niveles de modelado. El metanivel se refiere a abstracciones independientes del dominio. El nivel de dominio trabaja con conceptos específicos del dominio de aplicación. Finalmente, el nivel de instancia emplea instancias específicas de los conceptos del nivel de dominio. La información de estos modelos se describe mediante una notación gráfica y una lógica para tiempo real. La notación gráfica representa un subconjunto del lenguaje del formalismo.

Los requisitos del sistema se obtienen gradualmente y se expresan en un lenguaje de adquisición que refleja fielmente la estructura del metamodelo. Los conceptos en KAOS tienen una estructura genérica con dos niveles: Una capa semántica externa que declara el concepto, sus atributos y sus relaciones con otros conceptos; una capa interna para su definición formal. El nivel declarativo se usa para el modelado

conceptual mediante un lenguaje visual, la trazabilidad de los requisitos mediante la navegación de la red semántica de conceptos; y la reutilización de especificaciones. El nivel de aserción es opcional y se usa para el razonamiento formal.

2.2.1.7. CTT

ConcurTaskTrees es un sistema que permite realizar el modelado de tareas. Es un modelo ya existente y muy utilizado que se puede integrar en el ciclo de vida de esta metodología para modelar las tareas de los usuarios. Ya contempla alguna idea de colaboración entre usuarios.

2.3. CUADRO COMPARATIVO

Estudio Comparativo de Formalismos para Modelado de Espacios Virtuales

		Formalismos						
		AMENITIES	TOUCHÉ	CIAM	RBAC	SEM –HP	KAOS	CTT
Características	roles	Proporciona a los diferentes roles trabajar en una tarea colaborativa	En la etapa de análisis de identifican los roles	Agrega responsabilidades compartidas a cada rol	Es posible modelar una jerarquía de roles para el control de acceso	Filtra acciones designadas por roles		
	arquitectura	Arquitectura orientada a servicios (SOA)	Se basa en una arquitectura CE4Web		Su arquitectura es X-RBAC en donde se puede observar la existencia de un repositorio XML	Su arquitectura separa el diseño y la navegación HiperMedia		
	patrones	Proporciona patrones para informar elementos	Utiliza patrones para representar interfaces	Modelan patrones tipo awareness	Ofrece una rigurosa definición de los conjuntos y sus relaciones. La especificación funcional define los requerimientos para operaciones administrativas de creación y mantenimiento de los conjuntos y sus relaciones			Reutiliza partes de la especificación para la creación de patrones de tareas y árboles de tareas concurrentes

		Formalismos						
		AMENITIES	TOUCHÉ	CIAM	RBAC	SEM –HP	KAOS	CTT
Características	actividad	Casos distintos de actividades que pueden ser construidas a partir de patrones	Utiliza un modelo iterativo que consta de las actividades de elicitación, análisis y validación	Pueden asociarse varios roles que trabajan juntos en algunas actividades	Describir los tipos de acceso a los recursos del sistema, distinguiendo entre entidades activas (roles de usuario o ejecución de un proceso)	Utiliza Reglas de Orden (Ro) obtenidas de una Estructura Conceptual (Ec) que define las condiciones para ejecutar cada actividad		Este método de diseño de la interfaz actúa como un flujo de actividades que integra otros procesos de desarrollo que requieren o proveen información útil para el modelado de la actividad o, directamente, para la interfaz de usuario
	sesión		Utiliza sesiones a partir de la elicitación		Cada sesión está asociada a un único usuario y cada usuario tiene una o más sesiones. No considera las sesiones establecidas por más de un usuario			

		Formalismos						
		AMENITIES	TOUCHÉ	CIAM	RBAC	SEM –HP	KAOS	CTT
Características	elicitación	Permite llevar a cabo la elicitación de requisitos, el modelado del sistema y la negociación de dichos requisitos	Utiliza la etapa de elicitación de Requisitos: El conocimiento del dominio, requisitos, objetivos y estructura organizativa		Integra la elicitación de requisitos		Se centra en dar soporte a los procesos de elicitación de requisitos desde los objetivos a alto nivel	
	diagrama	Utiliza diagramas de tareas en notación COMO-UML	Utiliza diagramas de casos de usos, tareas, colaboración y estructura organizativa UML	Emplea una notación similar a los diagramas de clases de UML		Este modelo utiliza la notación COMO-UML. Sus mecanismos evolutivos constituyen un soporte automático que garantiza el cambio de alguna tarea ya definida	Los diagramas emplean esquemas preconceptuales	Cada caso de uso de negocio tiene asociado un diagrama de actividad y un diagrama de objetos de negocio
	artefactos		Propone una serie de artefactos (diagramas, plantillas, etc.)	Propone artefactos, objetos para representar el contexto compartido				

		Formalismos						
		AMENITIES	TOUCHÉ	CIAM	RBAC	SEM –HP	KAOS	CTT
Características	Awareness	Awareness se encuentra en la etapa de tareas de Artefactos, objetos de información. También en la etapa de Interfaz de usuario, en los artefactos de la comunicación. Permite la inclusión de conciencia grupal (Awareness) en los requerimientos en interfaces de usuarios	Awareness se encuentra en la etapa de requisitos, en las plantillas de requerimientos. También en la etapa de objetivos, en las plantillas de objetivos. También en la etapa de Interfaz de usuario, en Widgets especiales	Awareness se encuentra en le etapa de tareas colaborativas de un Contexto compartido. También en la etapa de Interfaz de usuario, en Widgets especiales				
	objetos de Interacción Concretos		Análisis (a través de diagramas), diseño (AIO son usados para diseñar interfaces de usuarios abstractas), implementación (CIOs Objetos concretos de interacción)					

		Formalismos						
		AMENITIES	TOUCHÉ	CIAM	RBAC	SEM –HP	KAOS	CTT
Características	notación		Utiliza la notación CTT, para modelar tareas	Modelado de la Interacción, utiliza un árbol de Descomposición de Tareas Interactivas (notación CTT ampliada con iconos para representar áreas de visualización en tareas colaborativas)				Notación fácil de usar y aplicable a proyectos reales de mediana y gran envergadura y con especificaciones de cierta complejidad
	etapas		Utiliza cuatro etapas este modelo: Elicitación de requisitos, análisis, diseño, implementación	Utiliza 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				Se definen cuatro etapas para la presentación de la interfaz de usuario

2.4. VALORACIONES SOBRE EL CUADRO COMPARATIVO

La valoración de los modelos de espacios virtuales se basa en la comparación de cada una de sus características:

Los roles se encuentran en cada uno de los modelos analizados en este documento como Amenities, RBAC, TOUCHE, CIAM, SEM-HP los mismos permiten establecer una jerarquía de roles, mientras que CTT no.

La arquitectura difiere en los modelos, Amenities aplica el lenguaje UML y algunas de las directrices utilizan el Proceso Unificado de Desarrollo Software para especificar la arquitectura del software de un sistema distribuido. En RBAC utiliza un repositorio XML, su arquitectura es X-RBAC. Para Touché se basa en una arquitectura CE4Web. Para SEM-HP en cambio separa el diseño y navegación de HiperMedia.

Los patrones son utilizados en los modelos de Amenities, proporcionando patrones que informan sus elementos. En RBAC es definido funcionalmente los requerimientos para establecer sus relaciones. En TOUCHÉ se lo utiliza para representar interfaces. En CIAM se lo modelan de manera awareness. En CTT utiliza lo especificado en las tareas y árboles concurrentes.

La actividad, en Amenities es construida a partir de patrones, en RBAC necesita que sea descrito los diferentes tipos de accesos. En TOUCHÉ utiliza un modelo interactivo detallando así las actividades. En CIAM son asociados a varios roles que trabajan juntos algunas actividades. En SEM-HP utiliza reglas de Orden obtenidas a partir de una estructura conceptual. En CTT a partir de la interface modela la actividad necesaria para el usuario.

La sesión para R-BAC está asociada a un único usuario. En TOUCHÉ lo establece en la elicitación.

La elicitación es utilizada en los modelos: Amenities, RBAC, TOUCHÉ y KAOS en donde es llevada a cabo en la solicitud de requerimientos.

Los diagramas utilizan del tipo UML para los modelos: Amenities, TOUCHÉ, CIAM, SEM-HP, CTT y KAOS.

Los artefactos en TOUCHÉ y CIAM son utilizados en diagramas para representar el contexto.

Awareness es usado en el modelo Amenities encontrándose en las etapas de tareas, artefactos. En TOUCHÉ y CIAM se encuentra en la elicitation, interfaz de usuario, widgets.

Los OIC para TOUCHÉ son usados en el diseño de interfaces de usuario.

La notación, para TOUCHÉ utiliza la notación CTT modelando sus tareas. En CIAM utiliza la notación CTT con iconos para representar áreas de visualización de tareas colaborativas.

Las etapas, para TOUCHÉ y CTT es utilizado en cada una de sus fases: Elicitación de requisitos, análisis, diseño, implementación. En CIAM es utilizado en el modelado de procesos.

3. DESCRIPCIÓN DEL PROBLEMA

En este capítulo se presenta una descripción del problema que se intenta resolver en este trabajo de tesis. En tal sentido, se enumeran las falencias de las características en los modelos estudiados (sección 3.1), se formulan consideraciones sobre la integración de los modelos en una propuesta (sección 3.2) y la unificación de dichos modelos para una futura solución (sección 3.3).

3.1. CARACTERÍSTICAS O CONCEPTOS NO EXISTENTES EN LOS MODELOS ESTUDIADOS

En esta sección se describen las características faltantes de los modelos estudiados, los cuales se ilustran en la figura 3.1.

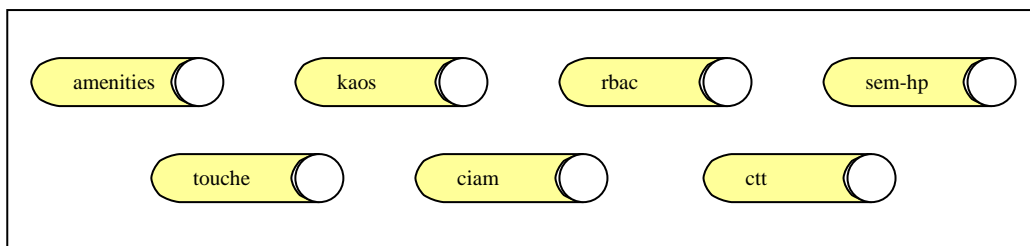


Figura 3.1. Modelos estudiados con características no existentes.

A continuación se citan dichas características:

- I. TOUCHE.** S. F. Zumaquero, 2010, menciona que a este modelo se le agregaría una nueva fase de mantenimiento para realizar las pruebas con el software en producción y así encontrar las fallas en la funcionalidad y/o mejoras. También debería añadirse nuevas iteraciones para poder volver a la fase de elicitación y de análisis ya que si se detecta alguna falla en el análisis, se deberá volver a esa fase sin necesariamente volver a elicitar requisitos. Hay muy poca especificación en cuanto al mecanismo para la creación de prototipos y pruebas de actividades por parte de los usuarios finales.

- II. RBAC.** Jisbo, 2007, dice que la naturaleza de este modelo en cuanto a los roles puede ser denominada estática, ya que carecen de flexibilidad y sensibilidad para el entorno en el cual es usado. Este modelo soporta roles activos para un usuario en un concepto de sesión, obteniendo así un conjunto de permisos disponibles para un usuario, pero no considera las sesiones establecidas por otros usuarios en el sistema, es decir no engloba todo el contexto asociado con el sistema. No especifica un control de grano fino sobre usuarios individuales en ciertos roles y sobre instancias de objetos individuales, aquí se observa la necesidad de establecer permisos comunes a grupos de usuarios. Es decir se debería crear un rol específico y asignando de forma individual este rol a cada usuario perteneciente a este grupo. Si se crean estos grupos de carácter temporal, provocaría que el control de acceso sea mas difícil de controlar.
- III. CIAM.** M.P.Velasco, dice que a este modelo se le agregaría una validación incorporando procesos de generación automática de IU (Interfaz de Usuario) y una metodología de diseño no sistemática, evaluando las respectivas interfaces de usuario generadas. Como resultado de esta evaluación se obtendría que las interfaces generadas con CIAM soportan todos los requisitos de interacción colaborativa y que el grado de satisfacción del usuario es alto en cuanto a trazabilidad. J.F.Martinez, 2012, menciona que CIAM soporta awareness está basado en Workspace awareness, pero no provee mecanismos a nivel desarrollador para soportar otros tipos de awareness y no está clara su trazabilidad desde dichos componentes de la interfaz de usuario hasta las tareas que la utilizan. Tampoco provee una forma para delimitar o restringir el tipo de información de Awareness proporcionada, por lo que queda a nivel de implementación el controlar dichos procesos.

IV. SEM-HP. Los Sistemas Hipermedia Adaptativos, SHA, desarrollados conforme a este modelo no son dinámicos, ni adaptables, ni tienen en cuenta información de contexto, aunque las dos últimas propiedades son fácilmente incorporables, ya que el modelo de usuario es capaz de integrar nueva información siempre que pueda ser calculada a partir de otra o sea actualizada directamente por el usuario.

V. KAOS. C. M. Zapata y L. A. Lezcano, 2009, plantean algunos problemas que se presentan en los diagramas de objetivos de KAOS es difícil automatizar el proceso porque los analistas suelen construir el diagrama de objetivos de forma subjetiva, tomando como base la información que suministran los interesados y se suele presentar una confusión entre los verbos que denotan objetivos y aquellos que expresan operaciones del dominio.

Cabe destacar la existencia de otras características de carácter más general a los modelos estudiados que enriquecerán este trabajo de tesis, las cuales se detallan a continuación:

- Extensión del modelo conceptual presentado de manera que considere aspectos dinámicos en sistemas CSCW, tal como el cambio de rol de los usuarios, que adapte y amplíe el modelo de sesión adoptado a las necesidades concretas.
- Introducción de las fases de análisis y la validación de los requisitos en la primera etapa del modelo de proceso, como es la elicitación de requisitos. Adicionalmente, y para ello, habría que estudiar cómo afectan las modificaciones introducidas en esta metodología a la metodología de [Durán, 2000]. Así como también, tener presente 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.
- Adaptación y ampliación de los criterios de accesibilidad y usabilidad a aplicaciones CSCW que permitan el uso de las aplicaciones al mayor número de personas posibles (que por su discapacidad no puedan hacerlo) y de la forma más sencilla posible.
- 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 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.

Se estima necesario realizar una propuesta integradora de dichos modelos que contemplen las mejores características de cada uno de ellos.

3.2 DISCUSIÓN DE LA INTEGRACIÓN DE LOS MODELOS EN UNA PROPUESTA

En función de los autores citados en la sección 3.1 que mencionan las carencias de los modelos estudiados, se identifica la falta de conexión o vinculación entre las diferentes características. En virtud de lo expuesto se estima generar un Modelo Integrador.

Se define la integración como la manera de conectar, ya sea de manera física o funcional, diversas aplicaciones con distintas capacidades para llegar a un único sistema con capacidades propias más amplias que las de cada una de las partes que lo conforman. En el contexto del software groupware, la integración permite llevar a cabo tareas a un nivel de abstracción mayor, aunque usualmente esto implica un mayor esfuerzo y complejidad de desarrollo.

La comunicación, la coordinación y la colaboración son actividades humanas inherentes a los sistemas CSCW. Estos requisitos funcionales implican que la interoperatividad debe resolver problemas tales como permitir el intercambio de una misma información entre usuarios que utilizan aplicaciones diferentes para comunicarse, por ejemplo, debido a motivos de usabilidad (existencia de diferentes perfiles de usuario, roles que desempeñar, etc.) o por las características propias de los dispositivos que se utilizan (ordenadores, dispositivos móviles, etc.).

3.3 UNIFICACION DE LOS MODELOS ESTUDIADOS EN UNA PROPUESTA

En virtud de lo expuesto, se considera generar un Modelo Integrador que concentre las características más significativas de los modelos descritos.

El aspecto clave consiste en la integración de dichas características en un único modelo, de forma transparente con gran capacidad de extensión y adaptación a los

usuarios y al contexto. Con este modelo se pretende facilitar el trabajo colaborativo en grupo.

4. SOLUCIÓN PROPUESTA

En este capítulo se presenta una propuesta de unificación de los distintos modelos estudiados en base a las mejores características de los mismos, y que confluyen en el desarrollo de un modelo integrador.

Se propone un Modelo Integrador (sección 4.1), se define una propuesta para la representación de modelos de tareas usando redes hipermedia (sección 4.2), se realiza una reseña de patrones de organización Joint Venture y los diferentes modelos (sección 4.3), se propone un método para la construcción del modelo cooperativo en base a patrones (sección 4.3.1), y se desarrolla un caso de estudio: Modelado de un proceso de aprendizaje colaborativo basado en la estrategia JigSaw (sección 4.3.2).

4.1. MODELO INTEGRADOR

Los modelos surgen como herramientas para integrar a los grupos de trabajos que no se encuentran físicamente juntos (figura 4.1), por consiguiente este trabajo de tesis intenta realizar un aporte que tiende a subsanar los modelos estudiados. A tal efecto es preciso tomar las mejores características de cada modelo para lograr un adecuado modelo integrador.

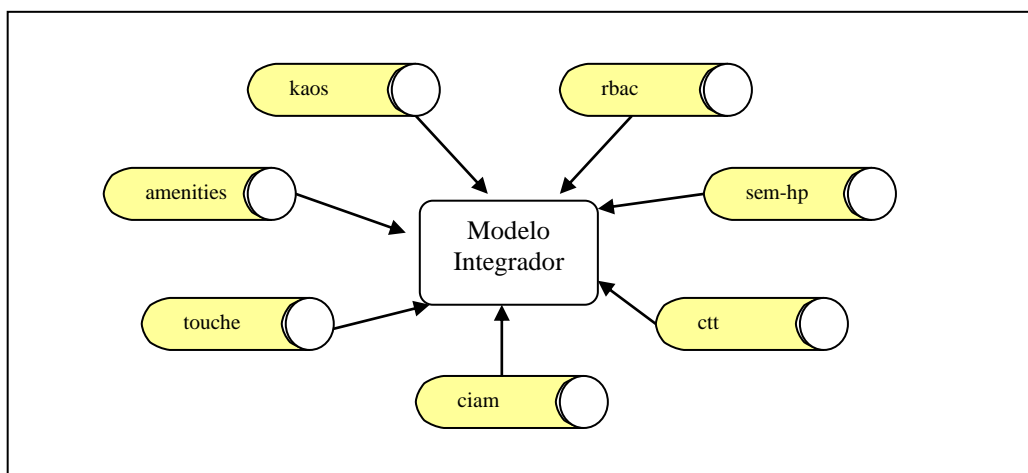


Figura 4.1. Modelos estudiados generando un modelo Integrador

A continuación se citarán las mejores características de los distintos modelos:

- I. El modelo que aborda la complejidad de los entornos colaborativos es AMENTITIES, centrándose en el concepto de grupo, cubriendo aspectos relevantes de su comportamiento y estructura. Además permite llevar a cabo la elicitación de requisitos (utilizando como notación COMO-UML), el modelado del sistema y la negociación de dichos requisitos en sistemas basados en computadoras, ofreciendo una especificación de sistema independiente de su implementación, proporcionando una mejor comprensión del dominio del problema. Este modelo colabora con un sistema para la gestión del conocimiento compartido llamado KnowCat, entre los puntos que sobresalen son:
 - a. La notación nos permite reflejar tanto la organización de la propia comunidad como la del conocimiento. Se pueden analizar por separado cada una de estas facetas porque se representan en diferentes vistas. Además la metodología nos permite también estudiar su interrelación mediante las tareas colaborativas de generación y opinión sobre las distintas unidades de conocimiento que son manejadas por el sistema.
 - b. Esta representación permite recoger las estrategias y la política que adopta la comunidad para gestionar el conocimiento. Por ejemplo, todas las decisiones se deben tomar democráticamente en votación, sin embargo cada decisión tiene sus propias características (síncrona/asíncrona, requiere mayoría absoluta, mayoría o unanimidad) y éstas son claramente recogidas con la notación que nos ofrece la vista interacción de la metodología.
 - c. La organización facilita la participación democrática activa de los miembros de la comunidad para generar conocimiento colectivo (a través de los colaboradores).

- d. La conciencia de grupo se obtiene a través de la participación dentro de un rol (sobre todo coordinadores y expertos) ya que poseen actividades conjuntas que refuerzan su identidad corporativa (aprobar, decidir, etc.).
 - e. En la cristalización del conocimiento participan de forma indirecta toda la comunidad a través de sus acciones (votar, consultar...), si bien esta labor es computada por un agente en un instante del tiempo.
- II. Para el tema de control de acceso sugerimos al mejor candidato el modelo RBAC por que cubre bien esa característica, en la que simplifica la administración de autorización. Puede ser reutilizado por cualquier aplicación o entorno que tenga como contexto un framework. Con este modelo se pueden utilizar dos servicios Web, uno como manejador de tareas para dar soporte a aspectos de coordinación y otro de sesión para representar el estado de cada usuario / subsistema en el sistema.
- III. Para los sistemas Hipermedia SEM-HP ofrecen mecanismos que respaldan todo su proceso de cambios por medio de un conjunto de mecanismos evolutivos (construcción, navegación y mantenimiento), facilitándose esta tarea. También proporciona un entorno grafico en el que es posible la visualización de actividades involucradas en cada una de las tareas, permitiendo un filtrado de las mismas en función de los roles y capacidades de los actores, asimismo cuenta con mecanismos que guían al actor a través de la secuencia correcta de ejecución para dicha tarea. La representación de SEM-HP consta de dos componentes interrelacionadas: Una Estructura Conceptual y un conjunto de reglas de orden. En este caso la Estructura Conceptual (EC) se transforma en una representación ontológica de las actividades consideradas y las relaciones existentes entre éstas. Las reglas de orden (Ro), por su parte, son las reglas lógicas obtenidas parcialmente a partir de la EC encargadas de definir las condiciones necesarias para la ejecución de cada actividad.

- IV. Para las características como la representación jerárquica y la complejidad de los modelos (trazabilidad) se obtienen mejores resultados con KAOS, da soporte a los procesos de elicitación de requisitos desde los objetivos a alto nivel (llevados a cabo por el sistema compuesto) hasta los objetivos operacionales. KAOS obtiene un mejor resultado para las características menos valoradas, como la representación jerárquica y la complejidad de los modelos.
- V. Awareness es considerado a nivel de especificación en el modelo TOUCHE, también el modelo aporta planillas para guiar las distintas fases del proceso y su trazabilidad mediante la herramienta de TOUCHE CASE Tool que permite automatizar la definición de los distintos artefactos involucrados y hacer un seguimiento guiado del modelo de proceso, reutilizando los elementos durante todo el proceso para asegurar y mantener la trazabilidad desde las etapas iniciales hasta el final. Los métodos de evaluación de usabilidad pueden apoyarse en los modelos de informes ya existentes en normas internacionales de calidad en la evaluación de usabilidad.
- VI. CIAM da un soporte más completo y con capacidad para representar una mayor semántica que el resto de los modelos que tratan el diseño de la capa de presentación en CSCW. Esta metodología fue extendida para soportar el modelado de conceptos específicos del Awareness de grupo a través de un meta modelo desarrollado para incluir estos conceptos y a la vez proporcionar una base flexible para incluir otros conceptos relacionados con Awareness. A nivel de implementación cada uno de los conceptos del awareness modelados puede traducirse en componentes de la interfaz de usuario gráfica (widget). Dichos widgets están basados en los proporcionados por Groupkit y relacionados con el soporte del Workspace Awareness.

4.2 PROPUESTA PARA LA REPRESENTACIÓN DE MODELOS DE TAREAS USANDO REDES HIPERMEDIA

La propuesta se basa en la consideración de tres aspectos relevantes dentro de los modelos de tareas: Estructura de la organización (roles), tareas a realizar por los miembros de la organización y recursos para llevar a cabo las tareas. Utilizando los diagramas del modelo AMENITIES.

Un proceso para realizar una representación de modelos de tareas usando redes hipermedia (SEM-HP).

Para el caso de transiciones entre roles no aditivas (el rol origen se abandona y no podrá ser recuperado posteriormente) basta con representar un único enlace entre el rol origen y el rol destino, teniendo como paso intermedio el nodo que representa la acción de cambio de rol (R-BAC).

Sin embargo, para la representación de transiciones entre roles aditivas (el rol origen no se abandona y podrá ser recuperado más tarde cuando se complete el trabajo del rol destino) es necesario incluir dos enlaces en sentidos opuestos entre el rol origen y el rol destino, con dos nodos intermedios representando respectivamente: Los cambios que provocan el adquirir el rol destino; y los cambios que provocan la recuperación del rol origen y la pérdida o abandono del rol destino.

Los modelos de tareas son detallados como descripciones lógicas de las actividades a realizar para alcanzar los objetivos del usuario. La siguiente figura muestra un modelo conceptual para la representación de las tareas que se realizan en una organización.

Este modelo de tareas es clasificado en tres tipos: Abstractas, individuales y de equipo.

Las tareas abstractas modelan tareas cuyo fin es servir de agrupación de otras tareas, las tareas de equipo son una o varias tareas realizadas por un grupo de usuarios del sistema para alcanzar objetivos comunes.

Las tareas abstractas y las de equipo están compuestas en última instancia por tareas individuales (figura 4.2).

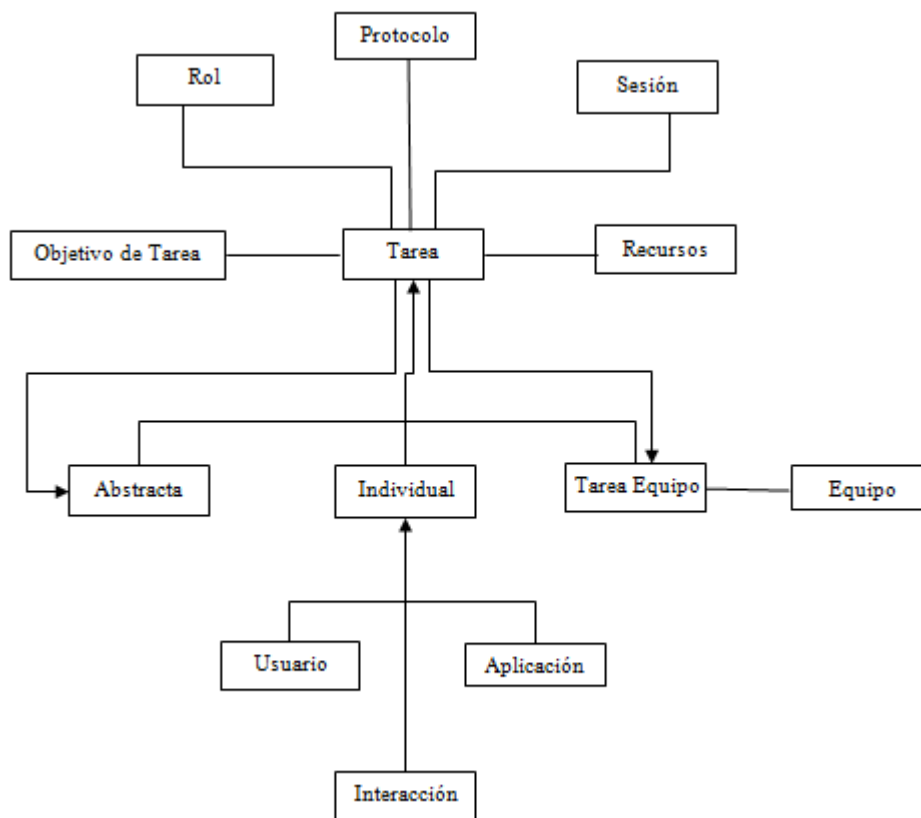


Figura 4.2. Modelo de tareas

Para las tareas de equipo, estas tareas individuales serán realizadas por los diferentes miembros que componen el equipo, pero bajo la influencia y la dinámica de trabajo en equipo y teniendo una responsabilidad compartida y no asignada. A su vez, las tareas individuales son caracterizadas como tareas de aplicación, de interacción o de usuario. Para establecer esta clasificación de tareas se ha partido de la propuesta de Paterno en su modelo para modelar tareas colaborativas denominado CONCURTASKTREE.

Del análisis de éste modelo conceptual se pueden extraer varias consideraciones que han de ser tenidas en cuenta para una representación hipermedia posterior. Así pues, hay que considerar que la razón de que un proceso tenga éxito o no, dependerá no solo de la lógica de aplicación de éste, sino también del cumplimiento de los objetivos asociados a cada una de las tareas.

También se ha de tener presente que una tarea no está directamente relacionada con los actores de un sistema, lo está indirectamente, puesto que dichos actores tienen

sentido en las estructuras organizativas por medio de los roles que desempeñan, y son precisamente los roles los que indican las responsabilidades que tienen dentro del marco de las tareas a realizar. Por último, y no menos importante, otro aspecto a considerar en la coordinación de las tareas de cualquier proceso colaborativo es todo lo relativo al control y presencia de los recursos asociados con las distintas acciones de una tarea.

Las ventajas de una representación hipermedia: Estructuración de la información, facilidades de navegación, y uso simultáneo de los recursos, entre otras, permiten organizar la información de los modelos de tareas de una forma flexible y extensible, de modo que dichos modelos sean susceptibles de ser manejados en entornos Web.

Así, partiendo de las observaciones expuestas anteriormente, para poder llevar a cabo una representación hipermedia de los modelos de tareas, es necesario considerar y representar tres aspectos relevantes: La estructura de la organización (roles), las tareas a realizar por los miembros de la organización y los recursos para llevar a cabo las tareas.

Para cada uno de estos aspectos establecemos una estructura hipermedia particular, y la unión de todas estas estructuras particulares da como resultado una estructura hipermedia multinivel (Roles, Roles-Tareas, Tareas y Tareas-Subactividades) que conforman la red hipermedia global.

El ejemplo que se ilustra representa la versión Web del popular juego de mesa conocido como Pictionary.

La particularidad de este ejemplo radica en el desarrollo de la trama propia del juego como una tarea colaborativa en la que participan distintos usuarios y en las que da lugar a cambiar de rol conforme evoluciona la traza del mismo.

La metodología empleada para representar los modelos iniciales que describen el sistema es AMENITIES, método estructurado para la construcción de modelos cooperativos.

El primer paso, en el proceso de obtención de la representación hipermedia, es analizar el aspecto relativo a la representación del comportamiento de los actores en el sistema: Especificación de roles y cambios de roles dentro de una organización.

En la figura 4.3 se muestra la especificación de la organización del sistema por medio de un diagrama de organización. El sistema queda estructurado como una organización de carácter general, organización Pictionay, la cual a su vez contiene una suborganización de carácter más específico, Equipo.

Para representar la información mostrada en un diagrama de la organización mediante una estructura hipermedia es necesaria una representación ontológica de los roles y de las transiciones que capacitan los cambios de rol.

Teniendo en cuenta esto, en nuestra estructura hipermedia se consideran dos tipos de nodos: Nodos que representan roles y nodos que representan las capacidades y/o leyes que provocan los cambios de rol.

Para el caso de transiciones entre roles no aditivas (el rol origen se abandona y no podrá ser recuperado posteriormente) basta con representar un único enlace entre el rol origen y el rol destino, teniendo como paso intermedio el nodo que representa la acción de cambio de rol.

Sin embargo, para la representación de transiciones entre roles aditivas (el rol origen no se abandona y podrá ser recuperado más tarde cuando se complete el trabajo del rol destino) es necesario incluir dos enlaces en sentidos opuestos entre el rol origen y el rol destino, con dos nodos intermedios representando respectivamente: Los cambios que provocan el adquirir el rol destino; y los cambios que provocan la recuperación del rol origen y la pérdida o abandono del rol destino en ese orden.

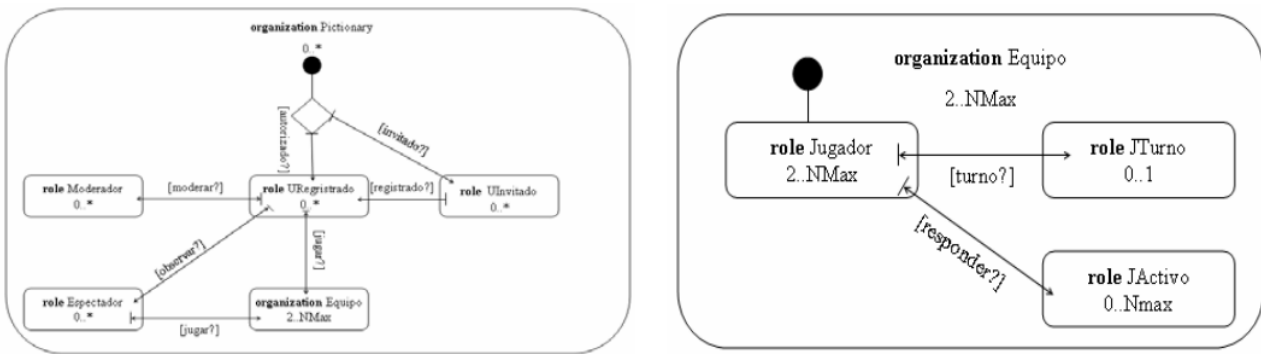


Figura 4.3. Diagrama de la organización Pictionary y Equipo

En el caso de organizaciones en las que no existe un único estado de partida (en el ejemplo existen inicialmente dos formas de acceso posibles según la capacidad que se haya adquirido: URegistrado o UInvitado) hay que incluir un nodo adicional (ver figura 4.3 representado mediante un nodo de doble elipse) para poder representar las transiciones hacia los cambios de rol de partida.

La figura 4.4 muestra la estructura hipermedia concreta que se obtiene para el diagrama de organización de la figura 4.3. Los nodos sombreados son nodos de cambio de rol.

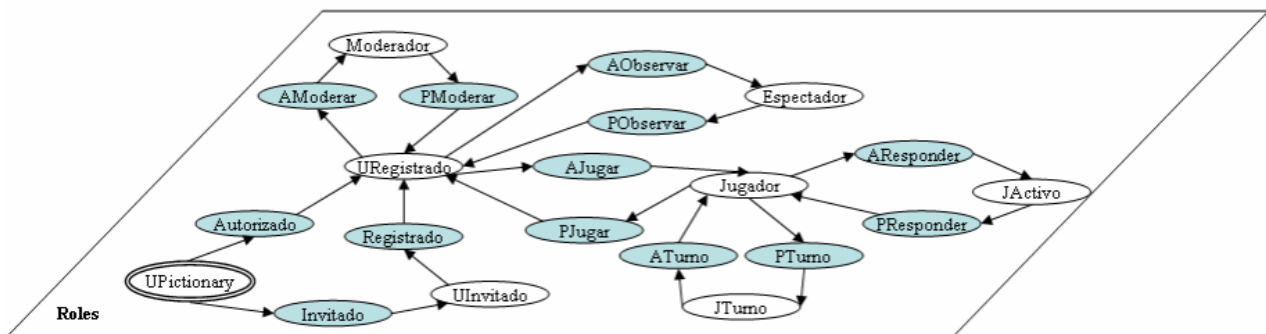


Figura 4.4. Red hipermedia (nodos sombreados, cambio de rol)

El segundo paso en el proceso de obtención de la representación hipermedia es analizar el aspecto relativo a la representación de la gestión de tareas en el sistema: Tareas desempeñadas por un rol, concurrencia de tareas, disparo de eventos, etc.

En AMENITIES la definición de roles por medio de diagramas de rol sirve de enlace entre la especificación de la estructura de la organización y de las tareas a desempeñar por los miembros de la organización.

En la figura 4.4 se muestra una asociación de tareas a roles para el caso particular del rol *JTurno*.

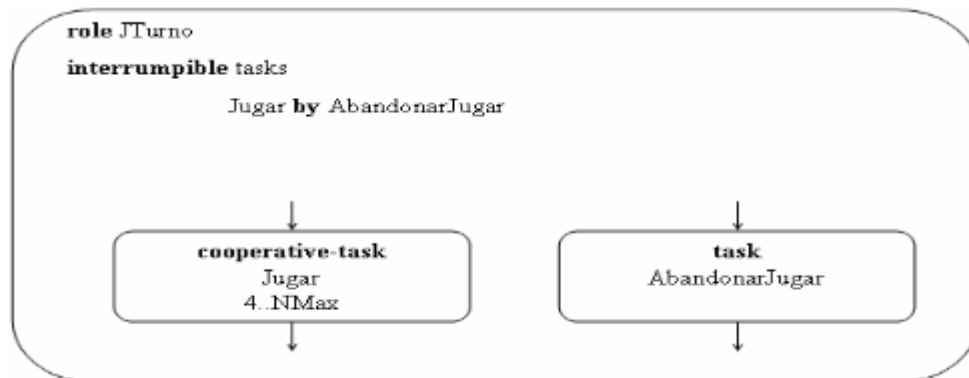


Figura 4.5. Diagrama de rol JTurno.

La representación de este diagrama produce como resultado una nueva estructura (nivel Roles-Tareas ver figura 4.6) que será el nexo de unión entre la estructura hipermedia de la organización y la estructura de las tareas a desempeñar por los miembros de la organización.

Cada una de las distintas tareas es representada por medio de un nodo en la estructura hipermedia. Los enlaces de la estructura sirven para representar dos conceptos: Las asociaciones de cada tarea con el rol que lo ejecuta (enlace desde el nodo de rol al nodo tarea); así como, la posibilidad de interrupción de la ejecución de una tarea por parte de otra u otras tareas (enlaces desde el nodo de la tarea que es interrumpida hacia el nodo o nodos de tarea que la interrumpe).

Una vez obtenida la representación hipermedia de la asociación de los roles con las tareas, el siguiente paso es completar el nivel jerárquico de Tareas considerando la especificación detallada de la ejecución de cada una de dichas tareas. En la figura 4.5 se muestra la especificación AMENITIES de la tarea cooperativa Jugar mediante un diagrama de tareas y de la subactividad TurnoJugada (perteneciente a la tarea Jugar) mediante un diagrama de subactividad.

Para representar la información mostrada en un diagrama de tareas y de subactividad en una estructura hipermedia es necesaria una representación ontológica de las tareas y subactividades así como de las transiciones que reflejen la relación entre ellas.

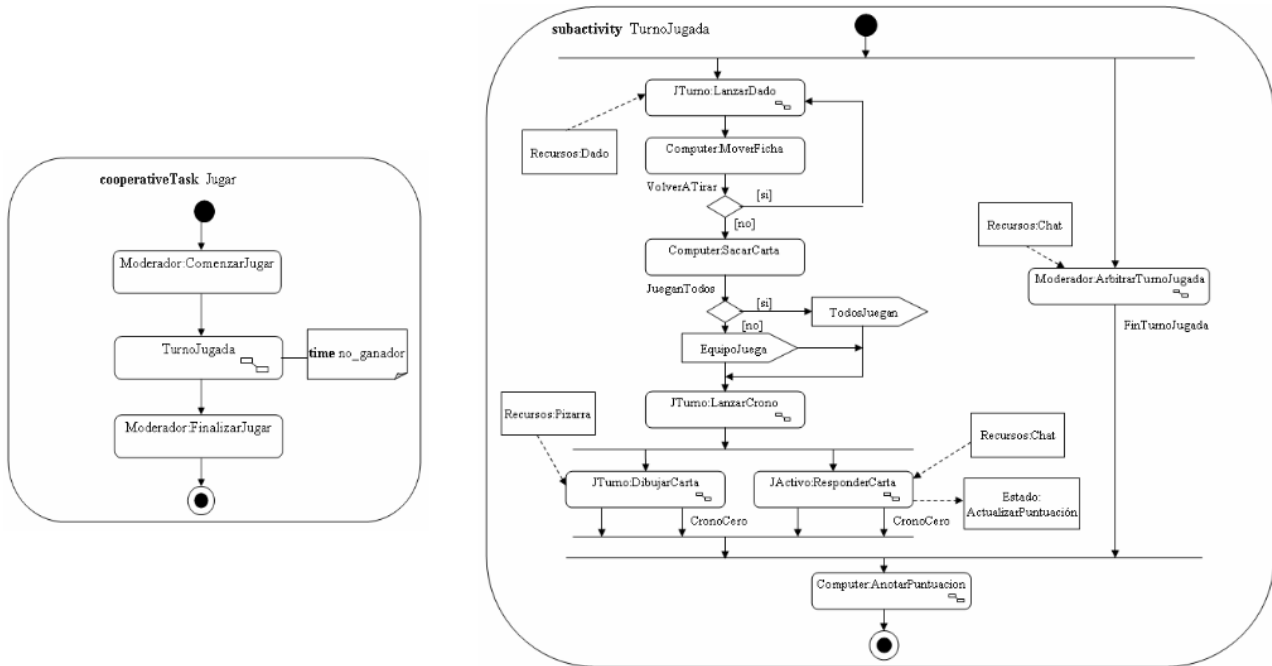


Figura 4.6. Diagrama de tarea Jugar y diagrama de subActividad TurnoJugada

En este caso diferenciaremos tres tipos de nodos: Nodos representando a cada una de las subactividades de la tarea; nodos representando el envío de señales o eventos; y nodos que representan el objetivo global asociado a la tarea.

El hecho de que un flujo de trabajo tenga éxito o no, depende principalmente de la lógica de aplicación de éste, aunque se pueda considerar en última instancia el ‘objetivo’ del proceso, por lo tanto, habrá situaciones en las que la ejecución de una tarea no sea alcanzada en base a una ejecución rígida de las tareas que componen su flujo, sino que bastará con que se alcance el cumplimiento de los objetivos/subobjetivos asociados a las tareas y subactividades con independencia del modo de ejecución de éstas. Los objetivos/subobjetivos serán por tanto nodos que conformen un camino alternativo a la ejecución de tareas.

Esta situación es muy habitual en procesos colaborativos en los que no se tiene clara la secuencia de acciones que realizarán los actores que colaboran pero si cual es el objetivo que se han planteado.

Los enlaces de la estructura hipermedia representan, ahora, el flujo entre las subactividades de la tarea teniendo en cuenta las diferentes leyes de aplicación:

- La ejecución secuencial de instancias de una misma subactividad es representada con un enlace cíclico al nodo que la representa. Ver TurnoJugada en la figura 4.6.
- Las subactividades iniciales en un flujo concurrente se representan usando un arco que unifique sus respectivos enlaces con la subactividad precedente al bloque concurrente, de este modo se indica que cualesquiera de los enlaces es de obligada ejecución (es necesario ver todos los conceptos asociados con los enlaces). Ver el nodo ComenzarTurno en la figura 4.7

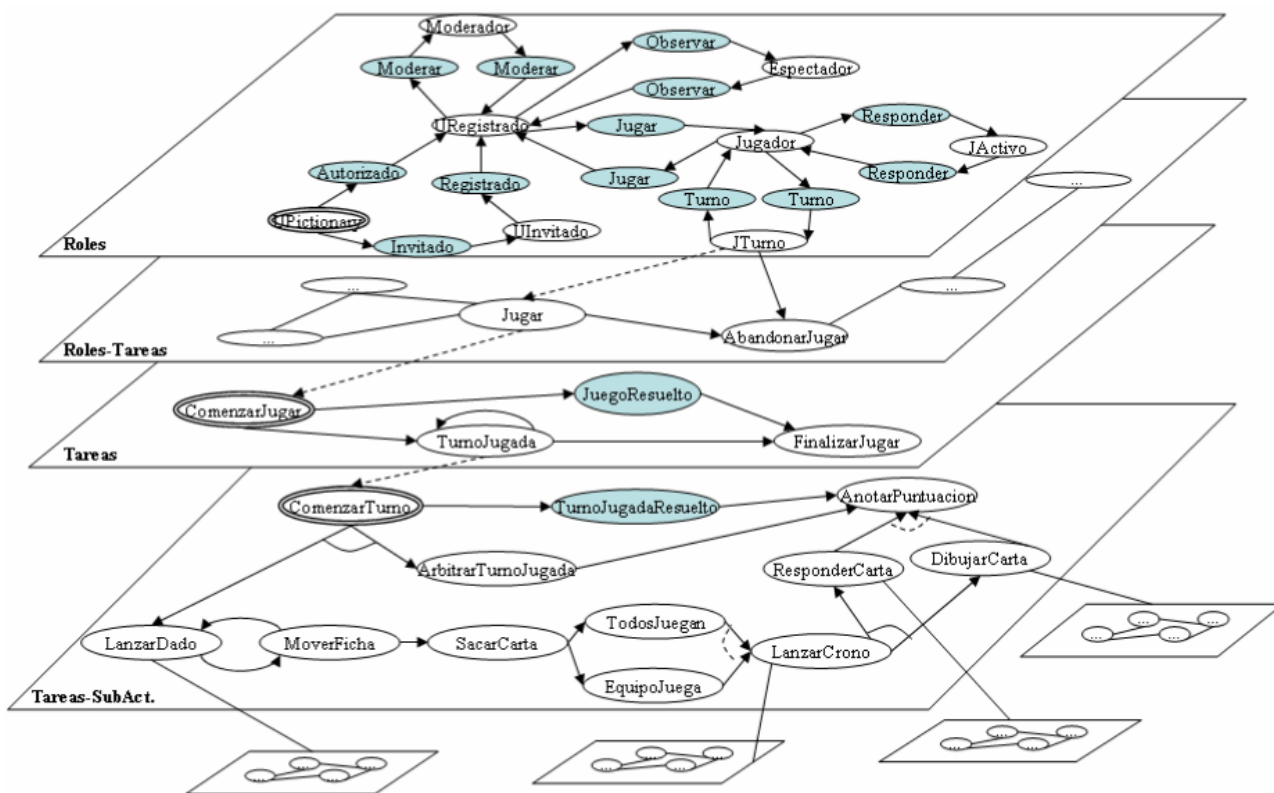


Figura 4.7. Red hipermedia conexión rol-tarea JugadorTurno-Jugar y conexión tareas-subactividadesJugar-TurnoJugada

- En las transacciones de flujos condicionales la subactividad origen es la subactividad precedente a la condición en el diagrama, y los destinos cada una de las subactividades posibles tras la evaluación de la condición. Ver los nodos SacarCarta (origen) y TodosJuegan y EquipoJuega (destinos).
- Por último, cuando existe más de una transición para alcanzar una misma subactividad se usa un arco punteado que permita unir las transiciones desde

todos los orígenes posibles, indicando que esa subactividad es alcanzable desde cualquiera de los enlaces que le llegan. Ver el nodo LanzarCrono o AnotarPuntuación

Los nodos sombreados a nivel de Tareas y Tareas-Subactividades representan ahora los objetivos asociados a las tareas. Los nodos iniciales para cada nivel son representados mediante una elipse doble. Con trazo discontinuo se representan enlaces entre nodos a diferente nivel.

Para el caso de tareas en las que no exista una única subactividad de partida (en el ejemplo, la subactividad TurnoJugada se inicia con un bloque concurrente) hay que incluir un nodo adicional (en el ejemplo, nodo ComenzarTurno) que marque el inicio. El tercer paso en el proceso de obtención de la representación hipermedia es analizar el aspecto relativo a la utilización de los recursos por las tareas en el sistema. Junto a la descripción jerárquica de las tareas es posible añadir una descripción paralela de los objetos manipulados por las distintas tareas bajo los diferentes roles.

En AMENITIES estos objetos son denominados artefactos: Unidad tecnológica que sirven de soporte para llevar a cabo ciertas acciones; y objetos de información: Entidad que contiene información para realizar ciertas acciones. Ambos conceptos quedan representados por medio de diagramas de clase y por su aparición en los diagrama de actividad que definen las tareas.

Para la representación hipermedia de los recursos basta con insertar a nivel de tarea/subactividad, en la estructura conceptual anterior, un nodo para cada uno de los recursos empleados y un enlace para expresar el uso de ese nodo por parte del nodo tarea que lo requiere. La red hipermedia resultante a nivel de Tareas-Subactividad es la mostrada en la figura 4.8. Los nodos sombreados representan ahora nodos recurso.

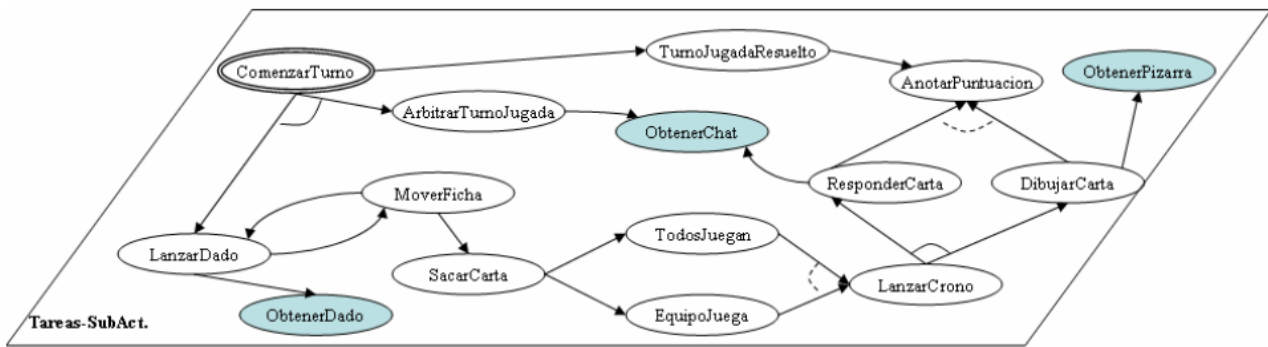


Figura 4.8. Red hipermmedia de la tarea Jugada

Todas las estructuras hipermmedia obtenidas jerárquicamente a lo largo de este proceso de representación forman finalmente una única red global que nos permite reflejar una especificación dinámica susceptible de adaptarse a los cambios del contexto.

Gracias a esta posibilidad de adaptación pueden ser reflejados cambios en el comportamiento de los actores los cuales determinan que los flujos de trabajo dentro de un sistema colaborativo no tengan que ser preestablecidos, sino que sea la propia persecución de unos objetivos condicionada por los cambios de contexto la que determine el camino a seguir en cada momento.

Los trabajos futuros estarían relacionados con la inclusión de la red hipermmedia obtenida dentro de la arquitectura orientada a servicios (WSA) para implementar sistemas colaborativos.

Dentro de esta arquitectura se propone un servicio WEB encargado de la coordinación de los flujos de trabajo que aparecen en las diferentes aplicaciones existentes en el sistema. La red hipermmedia definida, se convierte en el centro de información que el servicio WEB necesita para realizar todas sus actividades de forma flexible.

4.3 RESEÑA DE PATRONES DE ORGANIZACIÓN JOINT VENTURE Y SU MODELADO CON DIFERENTES MODELOS

Una manera de analizar y modelar un sistema consiste en describirlo como una estructura social que evoluciona dinámicamente a lo largo del tiempo y en la que cada miembro de dicha estructura se encarga de realizar una serie de actividades productivas para el negocio del sistema.

La inclusión de estos elementos en el estudio de un sistema nos permitir describir de una forma más precisa las responsabilidades de asignar a cada uno de los miembros del sistema y, por tanto, facilitar y guiar el posterior proceso de análisis y diseño del software.

Diferentes estudios realizados han propuesto estructuras organizativas que, por sus cualidades, a menudo rigen el contexto organizativo de un sistema. Este es el caso de estilos organizativos como la estructura de joint venture, integración vertical, pirámide, etc., los cuales dotan a toda la organización de una estructura que distribuye a sus integrantes (unidades organizativas o individuos) para conseguir objetivos globales. No obstante, también existen otras estructuras sociales, a priori de grano más fino, que suceden a menudo en el seno de las propias organizaciones. En la mayoría de los casos, éstas describen cómo se organizan un conjunto de actores que persiguen un objetivo particular. De este tipo son por ejemplo la estructura de broker, mediador, embajador, etc.

La intención es encapsular dichas estructuras organizativas en forma de “patrones de organización”, con el objetivo de favorecer su reutilización en diferentes situaciones y facilitar la especificación y comprensión del contexto organizativo de un sistema.

En la siguiente tabla se comentan los elementos necesarios para poder describir este tipo de patrones y que complementan la notación utilizada en los diagramas de organización.

Al mostrar uniformemente los aspectos que definen un patrón, su descripción se suele estructurar en secciones que facilitan su aprendizaje, comparación y uso.

Entre los formatos más utilizados se encuentra el llamado formato Alejandrino, también conocido como formato canónico, empleado por Alexander en sus trabajos (tabla 4.1).

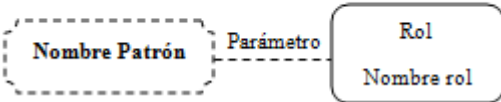
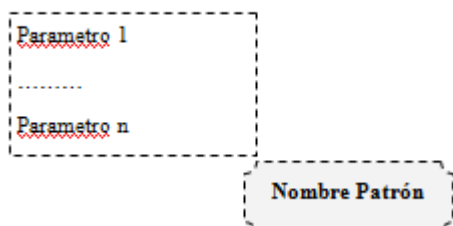
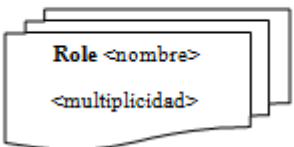

Símbolo	Semántica
<p style="text-align: center;">Ligadura</p> 	<p>Los actores que desempeñan el rol “NombreRol”, dentro de un sistema particular, asumen el rol “Parámetro” en el contexto de la estructura organizativa del patrón “NombrePatrón”.</p> <p>Las responsabilidades características del rol “Parámetro” tendrán que estar contenidas, posiblemente junto a otras necesarias para otros cometidos, en el rol “NombreRol”.</p>
<p style="text-align: center;">Vista externa de un patrón</p> 	<p>Permite representar una instancia del patrón “NombrePatrón” dentro de un modelo. Sus parámetros especifican los roles (u otros elementos) del patrón que deberán ligarse a elementos concretos de un sistema.</p>
<p style="text-align: center;">Rol múltiple</p> 	<p>Define un rol que puede ligarse a una cantidad indeterminada de roles concretos. En este caso, la expresión de multiplicidad indica la cantidad de actores que podrán desempeñar cada uno de los roles ligados.</p>
<p style="text-align: center;">Rol destacado</p> 	<p>Destaca un rol esencial para el patrón ignorando parte del diagrama. El patrón no indica las transiciones y estados intermedios para poder alcanzar el rol “NombreRol”.</p>

Tabla 4.1 Notación para el modelado de patrones de organización

Otro formato ampliamente usado es el denominado formato del GoF (Group of Four, haciendo alusión a los cuatro autores del famoso libro). No obstante, tal y como sostiene Vlissides, los autores deberían adaptar el formato de descripción al dominio concreto de aplicación. Siguiendo su consejo, se pensó que la siguiente plantilla puede ser útil para describir los patrones que se pretende modelar (tabla 4.2).

Nombre	Debe ser significativo y reflejar la esencia del patrón en pocas palabras.
---------------	--

Alias	Otro nombre por el cual es conocido.
Clasificación	Según alguna taxonomía previamente establecida.
Problema	¿Cuál es el escenario que pretendemos describir?
Contexto	Responde a preguntas tales como, ¿en qué situaciones se puede aplicar?, ¿cómo reconocer dichos escenarios? Muestra las precondiciones bajo las cuales el problema y su solución pueden ocurrir.
Participantes	Descripción de los elementos participantes y sus responsabilidades.
Solución	Mediante la notación COMO-UML, propia de la metodología AMENITIES, se especifica el modelo que describe los elementos participantes, su estructura y comportamiento. Puede incluir variantes.
Explicación	Descripción de la solución que se propone.
Ejemplo	Aparición del patrón durante el modelado de algún sistema real.
Patrones relacionados	Otros patrones que forman parte del mismo catálogo y con los cuales se relaciona. Por ejemplo, patrones que han podido aplicarse antes o que podrían aplicarse después, aquellos que son alternativos, etc.

Tabla 4.2 Plantilla de la descripción

Nombre	Alianza entre Empresas (Joint Venture)
Alias	
Clasificación	Organización
Problema	Describir la organización de un grupo de actores (socios), cada uno especializado en la realización de una tarea concreta, que unen sus capacidades y recursos para alcanzar

	objetivos más ambiciosos y así obtener una serie de ventajas colectivamente (inversión parcial, costos de mantenimiento más bajos, mayores beneficios, recursos compartidos, etc.).
Contexto	<p>El objetivo común se descompone en varios subobjetivos.</p> <ul style="list-style-type: none"> • Cada socio está especializado y es responsable de llevar a cabo alguno de los subobjetivos. • Existe un actor que se encarga de dirigir y representar la asociación. • Existe un actor que se ocupa de coordinar las actividades entre los distintos socios.
Participantes	<p>Rol Socio</p> <ul style="list-style-type: none"> • Ejecuta las tareas necesarias para lograr alguno de los subobjetivos marcados (tarea AlcanzarSubobjetivo) • Comparte sus recursos con los demás socios (tarea CompartirRecurso) • Rol Administrador • Se encarga de las relaciones con el exterior (tarea RepresentarAlianza) • Rol Administrador::Director • Decide la estrategia de la alianza (tarea TomarDecisiónEstratégica) • Rol Administrador::Coordinador • Convoca reuniones con los socios de la alianza (tarea ConvocarSocios) • Realiza reuniones de coordinación con los socios (tarea ReuniónCoordinación) • Toma decisiones de coordinación (tarea CoordinarSocios) • Rol Socio::Representante • Se reúne con el coordinador de la alianza cuando es convocado (tarea ReuniónCoordinación)
Solución	

	<p>organization Joint Venture role-pattern tasks Administrador: RepresentarAlianza Socio: CompartirRecurso, AlcanzarSubobjetivo</p> <p>organization Socio role-pattern tasks Representante: ReuniónCoordinación</p> <p>organization Administrador role-pattern tasks Coordinador: ConvocarSocios, ReuniónCoordinación, CoordinarSocios Director: TomarDecisiónEstratégica</p>
<p>Explicación</p>	<p>Quando un actor posee la capacidad para poder realizar una de las actividades necesarias para conseguir el objetivo común ([RealizaActividad?]), por ejemplo fabricar una de las piezas de un avión cuando el objetivo es construir un avión, éste puede desempeñar el papel de socio (obsérvese que el cardinal del rol indica que debería haber al menos un par de socios). Dentro de este estado un socio puede desempeñar además otros roles cuando realiza tareas determinadas.</p> <p>En el diagrama se destaca el rol de representante para indicar que este rol es imprescindible dentro del patrón, ya que una tarea clave dentro de esta organización va a</p>

consistir en la realización de reuniones entre los representantes y el coordinador de los socios.

La sección role-pattern tasks del patrón especifica las tareas esenciales que cada uno de los roles debería desempeñar bajo el contexto del patrón. Por supuesto, un rol podrá desarrollar además otros tipos de actividades, pero como mínimo debería realizar esas. Tal y como se puede observar, el rol socio tiene que realizar al menos las tareas CompartirRecurso (los socios deben poder compartir recursos entre ellos) y AlcanzarSubobjetivo (cada socio debe cubrir una parte específica del objetivo global de la organización, por ejemplo construir alguno de los elementos que componen el producto que se va a fabricar).

Por otro lado, tal y como se puede advertir en el diagrama de organización del rol socio, debe haber un actor que desempeñe el rol de representante si éste resulta elegido por el resto de socios. El representante tiene la obligación de reunirse con el coordinador cuando sea preciso (ReuniónCoordinación).

Esta actividad será colaborativa, es común para el coordinador y el representante.

Cuando un actor tenga la capacidad necesaria para poder administrar el Joint Venture podrá participar como administrador (obsérvese que solamente puede haber uno o dos actores que actúen como tales). En ese estado, un actor debería desempeñar al menos la tarea RepresentarAlianza, encargándose de las relaciones externas de la alianza. Si además tiene capacidad para coordinar los socios, podrá desempeñar el rol de coordinador (éste podrá desempeñarlo un único actor) y, por consiguiente, tendrá que reunirse con los representantes de los socios cuando sea necesario (tareas ConvocarSocios y ReuniónCoordinación) además de realizar labores de coordinación de éstos (tarea CoordinarSocios).

Si tiene capacidad para dirigir la estrategia de la alianza podrá desempeñar el rol de director (obsérvese que podrá ser desempeñado también por un único actor) cuya principal misión es la de tomar decisiones estratégicas para la alianza (tarea TomarDecisiónEstratégica). En este mismo diagrama se describe, mediante una transición aditiva, bajo qué ley un actor que desempeña el rol de coordinador podrá asumir además el rol de director. Tal y como se puede advertir, esto podrá suceder cuando el coordinador tenga capacidad de dirección y la unidad de dirección no esté disponible (el coordinador actúa en sustitución del director).

Ejemplo	<p>Este patrón describe una estructura organizativa bastante habitual en el ámbito de la alianza estratégica entre empresas, las cuales se agrupan para alcanzar un objetivo común y obtener una serie de ventajas de manera colectiva (inversión parcial, costos de mantenimiento más bajos, mayores beneficios, etc.).</p> <p>Un ejemplo real de este tipo de organización es la que posee la empresa Airbus, la cual coordina las actividades entre varias empresas que se han asociado para construir y vender aeronaves: Aerospatiale (desarrolla y fabrica principalmente la cabina), DASA (el fuselaje), British Aerospace (las alas), CASA (la cola) y finalmente el ensamblaje se hace en Aerospatiale. En este caso, todas las operaciones estratégicas, venta, postventa y marketing son realizadas por Airbus Industrie.</p> <p>Otro ejemplo, a menor escala, se puede dar durante el modelado de la estructura organizativa de un proyecto de investigación que es dividido en subproyectos asignados a diferentes grupos de investigación. Las labores de coordinación de todos los grupos suelen estar realizadas por alguien que mantiene contactos con representantes de los distintos subproyectos. Además, debe existir una cabeza visible con capacidad para dirigir todo el proyecto.</p>
----------------	--

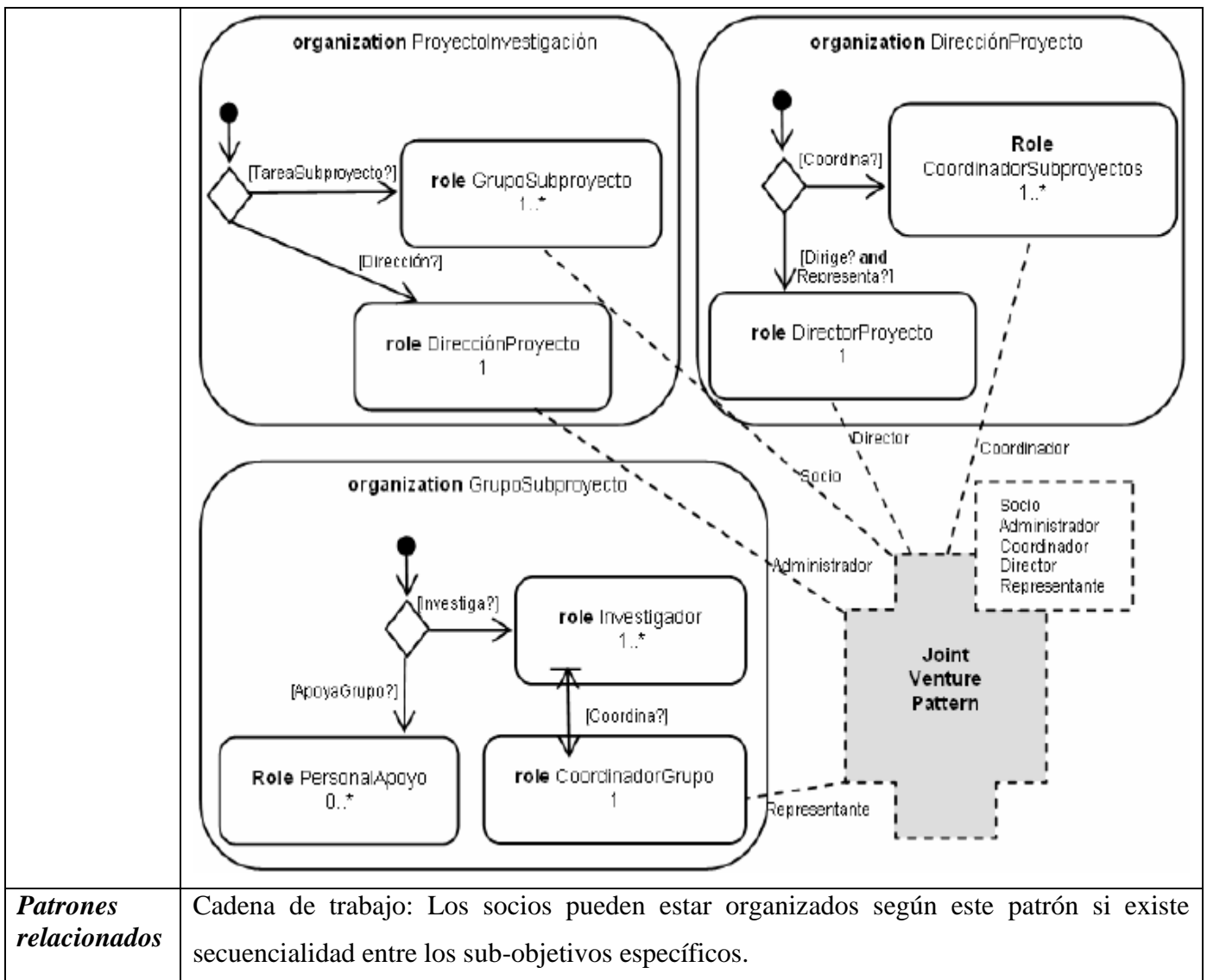


Tabla 4.3 Descripción del patrón Joint Venture

El modelo AMENITIES contempla cuatro vistas complementarias: Organizacional, cognitiva, de interacción, de información (tabla 4.4). Las fases en las que la metodología divide la construcción del Modelo Cooperativo son: Especificación de la organización, de los roles, de las tareas y protocolos de interacción y, por último, de los datos.

Considerando lo anteriormente mencionado y para facilitar la selección del patrón más adecuado en cada momento, se decide que una buena forma de organizar los patrones en el catálogo consiste en clasificarlos según la vista en la que pueden participar, la fase de especificación del Modelo Colaborativo en la que se suelen

aplicar y el aspecto concreto que abordan, donde este último define el tipo concreto de patrón cooperativo.

El modelo R-BAC se ajusta a las necesidades de que un usuario pueda pasar de desempeñar de un rol a otro. De esta manera multiplica la cantidad de actores que podrán desempeñar cada uno de los roles ligados.

El modelo CIAM propone su notación el lenguaje CIAN que promueve el modelado de la colaboración, comunicación y coordinación.

La siguiente tabla organiza según estos criterios los patrones del catálogo que hemos recopilado hasta este momento. De igual forma, se piensa que una organización puede ser adaptada de forma sencilla a otras metodologías y otros procesos de desarrollo específicos.

Vista	Fase de especificación	Tipo de patrón	Patrones
Organizacional	Organización	De organización	Joint venture Cadena de trabajo Broker Estructura en 5 Pirámide Mando-submando
	Tareas	De equipo	Circulo de calidad Equipo de dirección
Cognitiva	Roles	De rol	Coordinador Secretario
	Tareas	De actividad	Proceso de reunión Reunión Votación Negociación no moderada Convocatoria de reunión Llamamiento para propuestas Tormenta de ideas

Vista	Fase de especificación	Tipo de patrón	Patrones
Cognitiva	Tareas	De actividad	Crear plan de trabajo Enrutar formulario
	Tareas, Roles	De coordinación	Productor-consumidor simple discontinuo Productor-consumidor simple continuo Salvavidas Productor-consumidor múltiple Discontinuo Productor-consumidor múltiple continuo Turno de palabra
Interacción	Tareas, Protocolos de Interacción	De comunicación	Debate moderado Debate no moderado Petición-respuesta simple Petición-respuesta múltiple Exposición Mensajes encolados Publicación-suscripción
Información	Modelo Conceptual de Datos	De estructura	Acta de reunión Calendario de eventos Plan de trabajo
	Tareas, Roles	De acceso	Autorizado Primero en llegar-primero en servirse Orden de prioridad Turno de acceso

Tabla 4.4 Estructura del catálogo.

A continuación se muestra la red de interconexión de patrones (Figura 4.9) de acuerdo a la Tabla 4.4

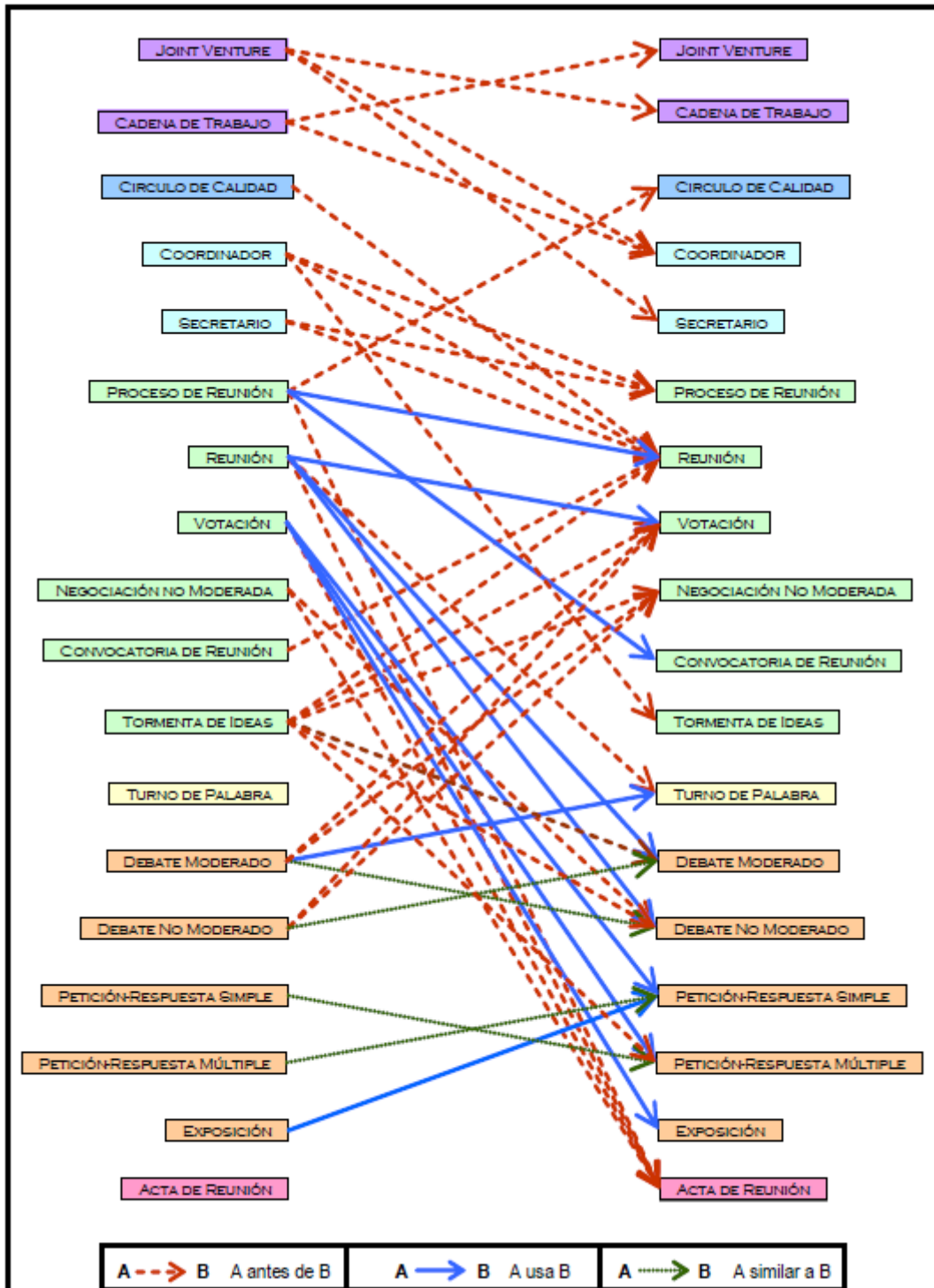


Figura 4.9. Vista parcial de la red de interconexión entre patrones.

4.3.1 UN MÉTODO PARA LA CONSTRUCCIÓN DEL MODELO COLABORATIVO EN BASE A PATRONES

No resulta ser de gran utilidad poseer un catálogo de patrones si no sabemos cómo usarlo. Durante el proceso de construcción del Modelo Colaborativo de un sistema, hemos de tomar multitud de decisiones que afectan a qué y cómo modelar. Justamente, los patrones nos asisten en la toma de esta clase de decisiones, guiando la percepción que tenemos del dominio y ayudándonos a identificar, comprender y especificar aquellos escenarios que son recurrentes.

El método que propuesto consta básicamente de dos pasos:

- Selección del patrón más adecuado para tratar el problema encontrado (modelar un determinado tipo de escenario).
- Aplicación del patrón anteriormente seleccionado dentro del contexto concreto en el que nos encontramos.

Puesto que el camino para seleccionar un determinado patrón puede variar dependiendo de la experiencia que tenga cada cual con el catálogo, la pretensión es que el método sea usado de manera flexible y que pueda adaptarse a las necesidades de cada modelador.

La selección se basa en la aplicación de tres filtros consecutivos, ordenados de grano más grueso a más fino.

El primer filtro aprovecha la clasificación de patrones presentada en la Tabla 4.4, donde los patrones son organizados según la vista en la que participan, la fase de especificación del Modelo Cooperativo y el tipo de problema que manejan. También se tienen en cuenta las relaciones de uso y las de proximidad que facilitan la búsqueda a partir de un patrón dado. La aplicación de este filtro dará lugar a un conjunto de patrones potencialmente aplicables.

El segundo filtro se basa en el examen del nombre y la intención de los patrones extraídos anteriormente.

El tercer filtro se fundamenta en el estudio minucioso de las secciones contexto, solución, explicación y ejemplo correspondientes a las plantillas que describen detalladamente los patrones obtenidos con el segundo filtro.

Si llegados a este punto hemos encontrado el patrón que se ajusta a nuestras necesidades podemos comenzar con la fase de aplicación. En caso contrario es posible que hayamos aprendido algo más sobre cómo resolver un problema. Nunca hay que forzar la aplicación de un patrón si no se ajusta realmente a nuestras necesidades, puesto que lo único que podríamos conseguir es modelar un problema diferente.

Basándonos en el modelado de patrones de catálogo, se usa PMP, la aplicación del patrón seleccionado consistiría esencialmente en crear primero la etiqueta que especifica las expresiones de ligadura que definen la instancia del patrón.

4.3.2 CASO DE ESTUDIO: MODELADO DE UN PROCESO DE APRENDIZAJE COLABORATIVO BASADO EN LA ESTRATEGIA DE JIGSAW

En esta sección se desarrolla un caso de estudio cimentado en un modelo de proceso de Aprendizaje colaborativo en base a la estrategia de Jigsaw. Se realiza un modelado de la organización (sección 4.3.2.1), se desarrolla un modelado de roles (sección 4.3.2.2), y el modelado de las actividades (sección 4.3.2.3).

Dicha estrategia de aprendizaje colaborativo que se modelará, basada en el conocido método Jigsaw, consta esencialmente de las siguientes fases:

- Se realiza una distribución de los alumnos en grupos pequeños con la misma cantidad de miembros (idealmente de 4 a 6 personas).
- Se divide el tema de trabajo (ejercicio, lectura material, etc.) en tantas partes como miembros haya en cada grupo.

- Cada miembro del grupo elegiría, o el profesor asignaría, una parte diferente del tema común de trabajo.
- Una vez que cada alumno prepare su parte, éste se reúne con los compañeros de otros grupos que tienen asignada esa misma parte, creando “grupos de expertos” temporales para discutirla, mejorarla y estudiarla en profundidad. Terminado el debate, cada alumno vuelve a su grupo de origen.
- Se proporciona tiempo suficiente para que cada cual, individualmente, actualice y prepare su fragmento.
- Cada estudiante expone a sus compañeros lo que ha aprendido sobre el tema (una de las partes del rompecabezas) y aclara todas aquellas dudas que surjan dentro del grupo.
- Al final los alumnos deberían ser evaluados.

Para llevar a cabo este caso de estudio, en primer lugar vamos a modelar la estructura organizativa de los actores que participan, sus responsabilidades y la forma en que se coordinan por medio de esta estrategia. Como veremos, para la construcción de estos modelos a veces nos apoyamos en la reutilización de patrones conceptuales que ya tenemos definidos en nuestro catálogo.

4.3.2.1 MODELADO DE LA ORGANIZACIÓN

Aunque éstos pueden a su vez desempeñar diferentes subroles, los actores que participan en el Jigsaw pueden asumir básicamente los siguientes roles:

- Aprendiz: Entre sus objetivos principales están el estudio y la realización de las tareas que el profesor (facilitador en el Jigsaw) propone.
- Facilitador: Planifica las tareas que deben hacer los alumnos como parte del Jigsaw e interviene en ellas facilitando su realización y evaluando a los alumnos.

La metodología AMENITIES cuenta con un tipo de diagrama de estados, denominado diagrama de organización, con el que es posible representar los distintos

roles involucrados, así como la dinámica de los actores desde el punto de vista de los posibles cambios de rol que pueden suceder, dependiendo de sus capacidades o el cumplimiento de normas existentes en la organización.

El diagrama de organización que aparece en la Figura 4.10 muestra precisamente los dos tipos de roles contemplados, junto con las capacidades necesarias para que puedan ser desempeñados. En este caso, para jugar el rol de aprendiz hay que contar con los conocimientos previos necesarios para poder llevar a cabo las tareas a realizar y para el rol de facilitador hay que tener capacidad para diseñar y ejecutar el Jigsaw.

La multiplicidad que aparece junto a cada uno de los roles marca el número de actores que pueden jugar dicho rol. Como puede verse en la siguiente figura, al menos cuatro actores deberán desempeñar el rol de aprendiz (para montar al menos dos grupos de dos aprendices) y tan sólo uno el de facilitador.

Para construir este diagrama no hemos usado ningún patrón del catálogo. Aunque en otros sistemas es fácil encontrar estructuras de organización típicas como las que están definidas en nuestro catálogo a nivel de organización.

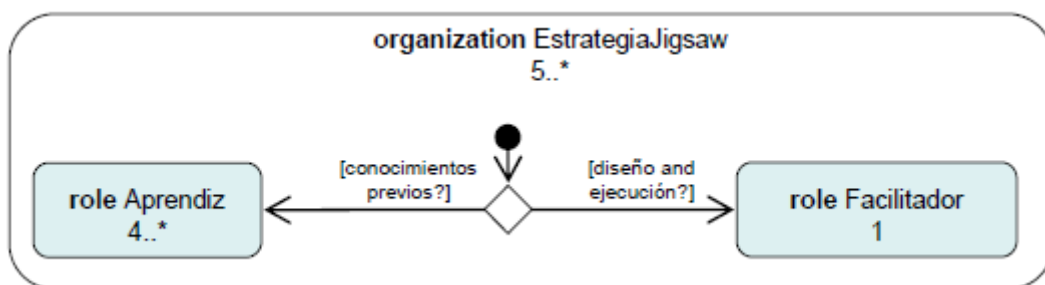


Fig. 4.10 Diagrama de organización para la estrategia Jigsaw.

4.3.2.2 MODELADO DE LOS ROLES

Formando parte de la vista cognitiva, iniciamos la especificación de los roles que participan en el escenario colaborativo que estamos abordando. Para ello usamos lo que en AMENITIES se conoce como diagramas de rol. Este tipo de diagramas permiten representar las tareas (objetivos generales) de las que son responsables cada

uno de los roles, así como sus características más relevantes (naturaleza cooperativa, mecanismos de activación y modos de sincronización, posibles interrupciones, etc.).

Los diagramas para ambos roles se muestran en la Figura 4.11

Entre todas las tareas la única que es cooperativa es EjecutarJigsaw, en la que participa un actor con el rol de Facilitador y al menos cuatro con el rol de Aprendiz.

También podemos observar que la única tarea cuya ejecución está sujeta a la aparición de un determinado evento es la de EjecutarJigsaw por parte del rol Aprendiz. Esta tarea no empieza hasta que el Facilitador lo ordena.

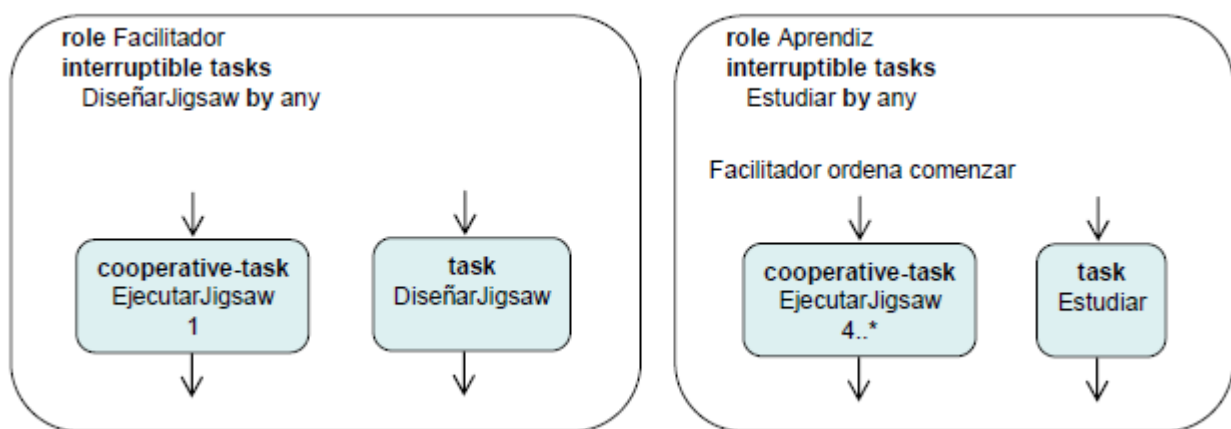


Fig. 4.11 Diagramas de rol para la estrategia Jigsaw

Respecto a la posibilidad de interrupción, la única tarea que no puede ser interrumpida por otras bajo ninguno de los roles que cooperan en su realización es EjecutarJigsaw. El resto de tareas pueden ser interrumpidas por cualquier otra.

La construcción de estos diagramas tampoco ha necesitado de patrón alguno, aunque para este artefacto se propone al modelo R-BAC.

4.3.2.3 MODELADO DE LAS ACTIVIDADES

Por simplicidad e interés, en esta fase nos vamos a centrar en el modelado de la tarea cooperativa EjecutarJigsaw, común a los roles Facilitador y Aprendiz.

Conociendo los pasos necesarios para poner en práctica esta estrategia, por medio de un diagrama de actividad de UML se modela en la Figura 4.12

Este diagrama añade elementos propios de la metodología AMENITIES y del perfil PMP que permiten expresar, respectivamente, ciertos requisitos de colaboración y la aplicación de patrones. Centrándonos en el uso de los patrones, destacamos la actividad concurrente SeleccionarSubtarea, cuya definición puede determinarse por el patrón NEGOCIACIÓN NO MODERADA (Figura. 4.13). Para su selección hemos aplicado el método expuesto en la sección 4.3.1, pudiendo comprobar que desde el primer filtro ya se arroja como resultado este patrón. Los filtros segundo y tercero lo único que han hecho es confirmar que dicho patrón es el que se ajusta a las necesidades de modelado de dicha actividad.

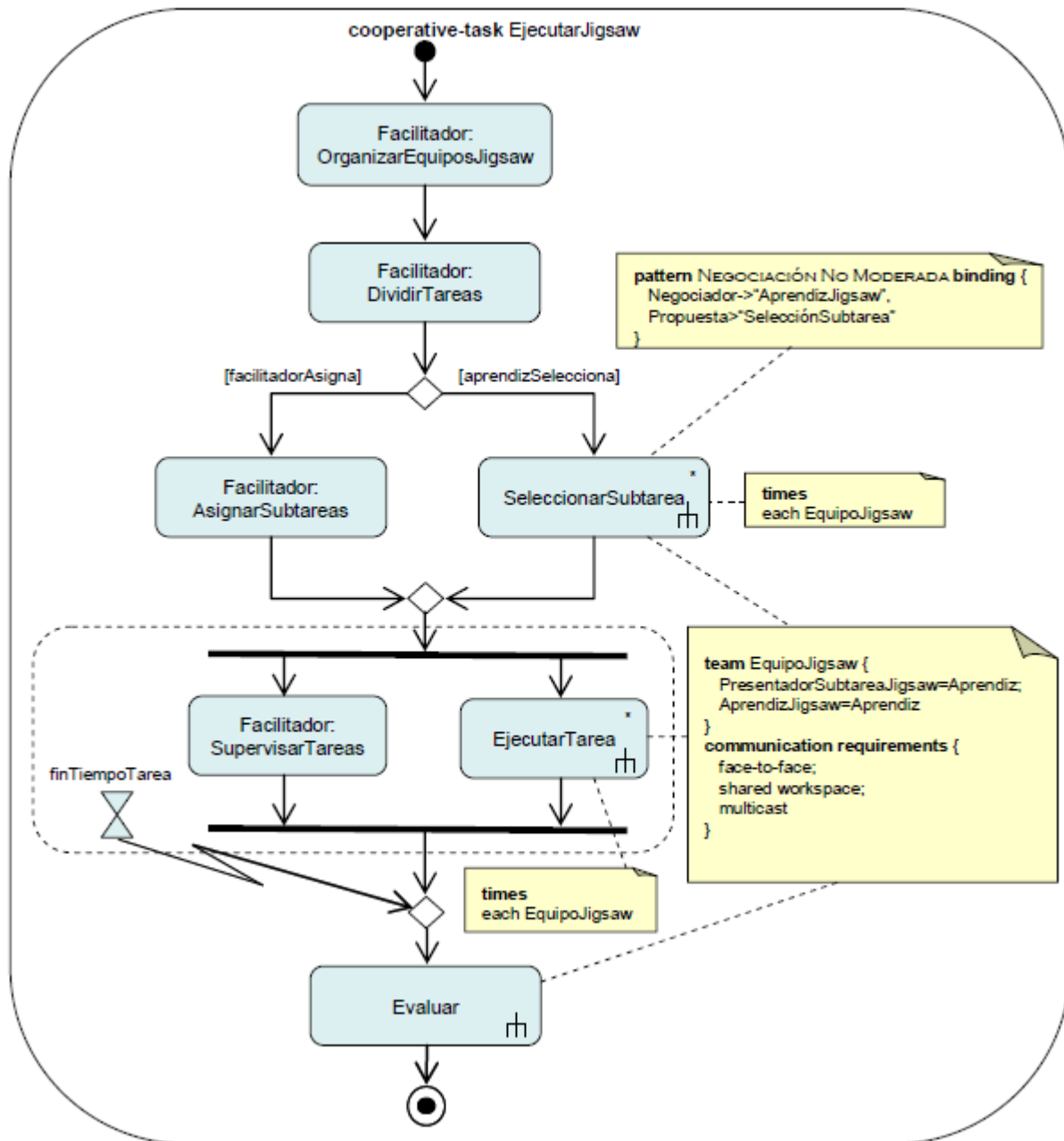


Fig. 4.12 Diagrama de actividad para modelar la tarea cooperativa EjecutarJigsaw.

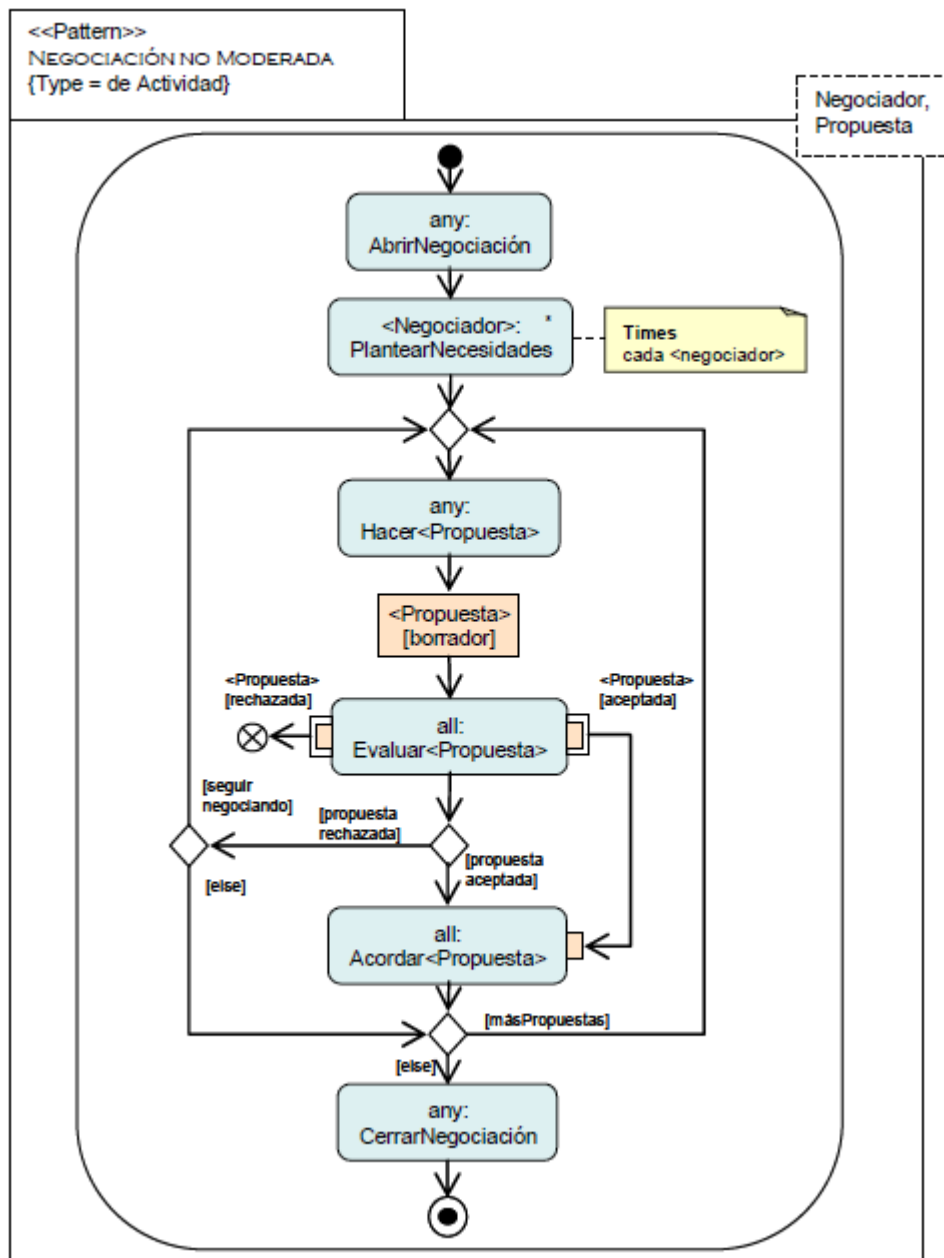


Fig. 4.13 Modelo correspondiente al patrón NEGOCIACIÓN NO MODERADA.

La especificación de ligadura contenida en la etiqueta señala que el modelo que representa dicha actividad se puede generar sustituyendo el parámetro *Negociador* por el valor “AprendizJigsaw” y el parámetro *Propuesta* por “SelecciónSubtarea”. Conociendo el patrón y su especificación de ligadura, no sería necesario añadir nada más para poder comprender y desplegar rápidamente, si fuera preciso, el modelo en cuestión. Entre las ventajas de la aplicación de patrones está la reutilización del conocimiento y el establecimiento de un vocabulario común.

La otra tarea clave que vamos a modelar para entender este proceso de aprendizaje colaborativo es la sub-actividad EjecutarTarea (Figura 4.14). Como hemos visto en la Figura 4.11, esta actividad es realizada por todos y cada uno de los equipos que intervienen en el Jigsaw y los requisitos de comunicación especificados son heredados por todas sus subactividades.

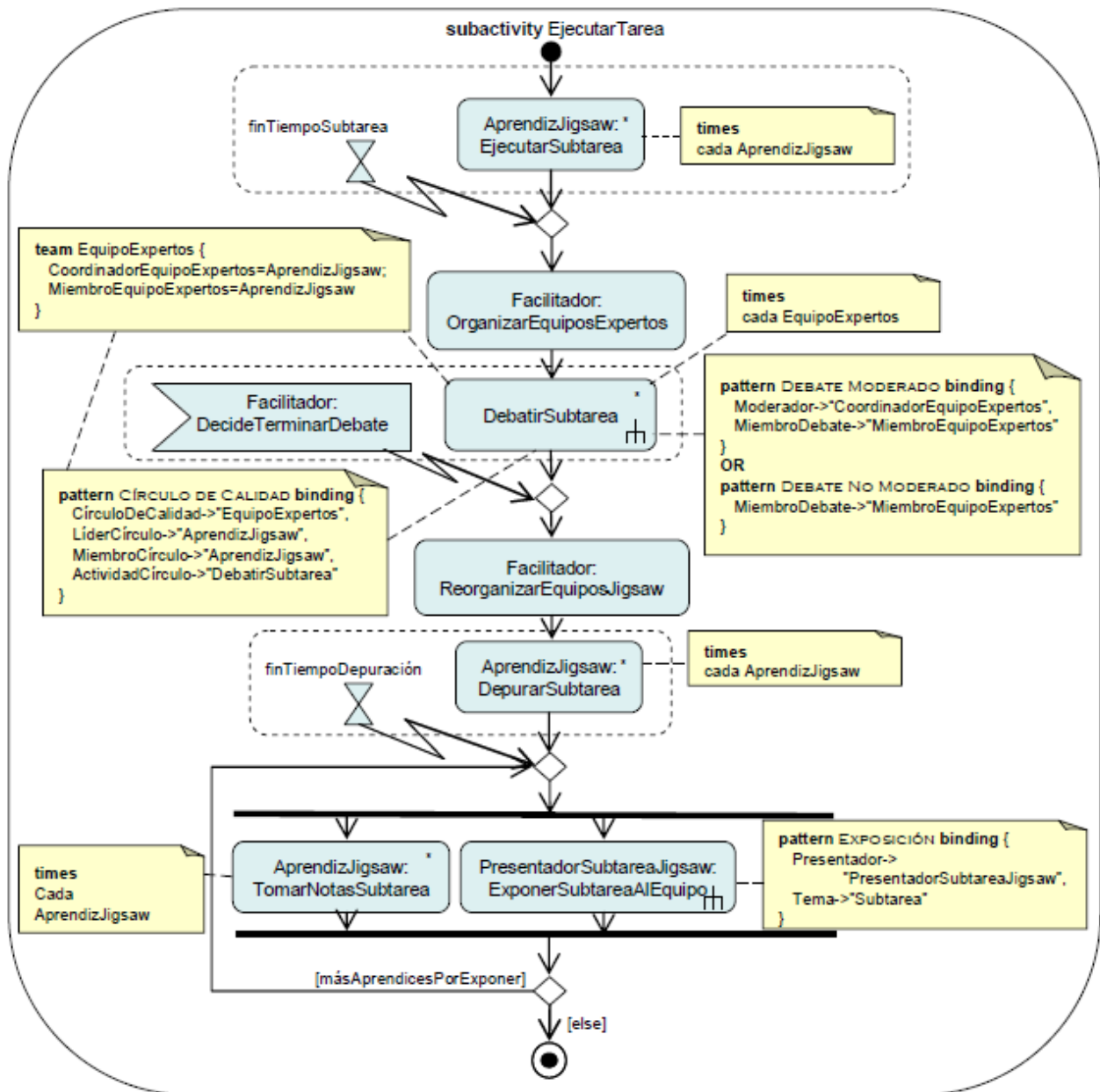


Fig. 4.14 Diagrama de actividad correspondiente a EjecutarTarea.

Igual que antes, enfocamos nuestro análisis en la selección y aplicación de patrones del catálogo que nos ayudan a modelar/describir este escenario colaborativo.

Como muestra el diagrama, el Facilitador organiza equipos compuestos por miembros de los distintos grupos que tienen asignada una misma subtarea, creando así “grupos de expertos” (team EquipoExpertos) temporales para discutirla, estudiarla y, a ser posible, mejorarla. En este caso, es el patrón CÍRCULO DE CALIDAD el que hemos seleccionado, permitiéndonos especificar de manera precisa esta clase de equipos. Los equipos de mejora o círculos de calidad se forman con el objetivo de mejorar, en la medida de lo posible, el desarrollo de una determinada tarea.

En cuanto a la actividad DebatirSubtarea, podemos ver cómo ésta se describe mediante la ligadura dinámica de los patrones de comunicación DEBATE MODERADO y DEBATE NO MODERADO. La ligadura dinámica permite reflejar que el equipo de expertos puede cambiar en cualquier momento el protocolo (patrón) de comunicación utilizado para el debate. Como vemos, en caso de un debate moderado la persona que actuaría como Moderador sería quien ocupa el rol de CoordinadorEquipoExpertos dentro del equipo.

Podemos comprobar también que hemos utilizado el patrón de comunicación PRESENTACIÓN para especificar la actividad ExponerSubtareaAlEquipo. Para su ligadura necesitamos saber simplemente quién va a actuar como Presentador, en este caso el miembro del EquipoJigsaw que tiene el rol de PresentadorSubTareaJigsaw en ese momento, y como Asunto usamos Subtarea.

El sistematizar el modelado conceptual de un sistema colaborativo tiene como base a patrones y catálogo. Se muestra una clasificación e interconexión más de una treintena de patrones en base a criterios que facilitan enormemente la selección del más adecuado en cada momento. Estos patrones son descritos mediante una plantilla homogénea que nos ayuda a estudiarlos, compararlos y aplicarlos.

El reconocimiento inmediato de abstracciones clave que son recurrentes en determinados momentos y situaciones durante el proceso de modelado conceptual de un sistema colaborativo.

Los beneficios que destacan en este documento son:

- El modelado rápido de estas abstracciones por reutilización (instanciación) del patrón seleccionado.
- La utilización de un vocabulario común que nos permite comunicar y razonar en base a dichos patrones.
- La mejora de la comprensión, comunicación y mantenimiento de los modelos, así como de la documentación en general.

5. CONCLUSIONES Y FUTUROS TRABAJOS DE INVESTIGACIÓN

En este capítulo se presenta el aporte realizado en este trabajo de tesis (sección 5.1); así como también, las futuras líneas de investigación que de él se desprende en opinión de la autora (sección 5.2).

5.1. APORTES

Los aportes más importantes que se pueden extraer del presente trabajo de tesis, se destacan los siguientes:

- Diseño de un modelo integrador que genere un abanico de posibilidades desde diferentes puntos de vista, en lo que se refiere a la ponderación de cada una de las características de los modelos estudiados a juicio del diseñador.
- Aplicación de los modelos existentes de la Ingeniería de Software y la interacción persona-ordenador al ambiente de trabajo cooperativo.
- Este trabajo presenta las bases de un modelo Integrador cimentados a una serie de técnicas y modelos que permitirán especificar y desarrollar un modelo cooperativo a partir de los modelos estudiados; a la vez que conectados, y teniendo en cuenta la interfaz de usuario, o lo que da a llamar interfaz de grupo.

A modo de síntesis se estima de interés hacer referencia a las características más importantes de los diferentes modelos colaborativos, los cuales han servido de base para el desarrollo del Modelo Integrador propuesto:

- Amenities es considerado como uno de los mejores modelos en cuanto a la notación que utiliza [COMO-UML], porque se adapta fácilmente a las políticas y estrategias de grupo.

- Para los mecanismos evolutivos como navegación, construcción y mantenimiento se encuentra el modelo Hipermedia de SEM-HP.
- KAOS se muestra notable en la trazabilidad de la representación jerárquica.
- TOUCHE CASE Tool permite generar los distintos artefactos involucrados y hacer un seguimiento guiado del modelo de proceso, reutilizando los elementos durante todo el proceso para asegurar y mantener lo diseñado desde las etapas iniciales hasta el final.
- CIAM da un soporte más completo y con capacidad para representar una mayor semántica que el resto de los modelos que tratan el diseño de la capa de presentación en CSCW.

5.2 FUTUROS TRABAJOS DE INVESTIGACIÓN

Entre las futuras líneas de investigación cabe destacar las siguientes:

- Se sugiere avanzar en un proceso metodológico que posibilite refinar este modelo a los efectos de establecer mejores alcances y capacidades en el desarrollo de la aplicación groupware; con la idea de fomentar la colaboración y potenciar el aprendizaje grupal.
- Los sistemas GroupWare disponen de algún sencillo buscador interno, pero no cuentan con servicios, que informen de aquellos contenidos de los que deberíamos estar enterados. Tampoco suelen disponer de indicadores a contenidos, aunque algunos disponen de un sistema de avisos y notificaciones de eventos -de forma generalizada- a todos los usuarios, sin distinción de funciones
- De cara al futuro, no cabe duda que irán apareciendo muchas más aplicaciones específicas para el trabajo colaborativo de libre distribución o freeware que permitan a distintos tipos de comunidades utilizar estos recursos que tan alto costo tienen actualmente.

6. REFERENCIAS

John, 2013, Groupware CSCW Computer Supported Cooperative Work. http://www.johnagredo.com/index.php?option=com_content&view=article&id=245&Itemid=33. página vigente al 16/04/2013.

Francisco Gutiérrez, 2007, Taller en Sistemas Hipermedia Colaborativos y adaptativos. <http://www.sistedes.es/ficheros/actas-talleres-JISBD/Vol-1/No-1/SHCA07.pdf>. página vigente al 16/04/2013.

Karina Terán, 2008, Sistemas colaborativos, Cátedra Tecnología Educativa. Universidad nacional experimental de las fuerzas armadas. <http://estoeseun ejemp lo.wikispaces.com/file/view/sistemas+colaborativos.pdf>. página vigente al 16/04/2013.

José Figueroa, 2012, Integración de los requerimientos de conciencia situacional y grupal al desarrollo de sistemas colaborativos y dinámicos usando un enfoque basado en modelos. Cátedra Tecnología Educativa. Universidad de Granada. <http://buhoz.net/public/jose/ugr/jfigueroa-phd201209.pdf>. página vigente al 18/04/2013.

Sebastián Romero, 2010, Diseño basado en modelos de sistemas interactivos sociales. Cátedra Tecnología Educativa. Universidad Castilla - La Mancha. <http://www.sebastianromero.net/papers/designSIS.pdf>. Página vigente al 18/04/2013.

M. Sánchez, , Modelo de control de acceso de un sistema colaborativo. Cátedra Tecnología Educativa. Universidad de Granada. <http://www.aipo.es/articulos/4/24.pdf>. Pagina vigente al 20/04/2013.

Hernan Sagastegui, 2005, Modelo de control de acceso x-rbac para un framework de aprendizaje colaborativo. Cátedra Tecnología Educativa. Universidad Privada Antenor Orrego. http://www.academia.edu/486983/MODELO_DE_CONTROL_DE_ACCESO_X-

RBAC_PARA_UN_FRAMEWORK_DE_APRENDIZAJE_COLABORATIVO. Pagina vigente al 20/04/2013

Nuria Medina, , SEM-HP: Un modelo para el desarrollo de sistemas hipermedia adaptativos. Cátedra Tecnología Educativa. Universidad de Granada. <http://lsi.ugr.es/~mgea/workshops/coline02/Articulos/nmedina.pdf>. Pagina vigente al 26/04/2013

José Luis De La Vara González, 2006, Derivación de modelos de requisitos a partir de modelos organizacionales. Cátedra Tecnología Educativa. Universidad Politécnica de Valencia. http://hci.dsic.upv.es/jdelavara/PFC_JoseLuisDeLaVara.pdf. Pagina vigente al 27/04/2013.

Blanca Arribas, 2007, CoCTT: Modelado de tareas en un entorno colaborativo. Cátedra Tecnología Educativa. Universidad De Castilla - La Mancha.

<http://www.google.com.ar/url?sa=t&rct=j&q=modelo%20%2B%20%22ctt%22%20%2B%20sistemas%20colaborativos%20groupware&source=web&cd=1&cad=rja&ved=0CCoQFjAA&url=http%3A%2F%2Fwww.usixml.org%2Fservlet%2FRepository%2Farribas-blanca07.pdf%3FID%3D1741%26saveFile%3Dtrue&ei=3W-WUeDVJo2y0QH2toEw&usq=AFQjCNHa4G6JhXKZummLHKXIK0IHPUQBDg&bvm=bv.46471029,bs.1,d.aWM>. Pagina vigente al 27/04/2013.

Garrido, 2004, Una propuesta arquitectónica para el desarrollo de aplicaciones colaborativas. Cátedra Tecnología Educativa. Universidad de Granada. <http://www.aipo.es/aipo/articulos/3/24.pdf> Página vigente al 16/06/2013.

Cobos, 2003, Modelado del sistema para trabajo colaborativo KnowCat mediante AMENITIES. Cátedra Tecnología Educativa. Universidad de Granada. http://lsi.ugr.es/~mgea/invest/articulos/cobos_interacc03.pdf. Página vigente al 20/06/2013.

Cristian, 2008, Especificación formal del modelo RBAC en el calculo de construcciones inductivas. Cátedra Tecnología Educativa. Universidad Nacional de Rosario. <http://www.fceia.unr.edu.ar/~crosa/thesis/rosa.2008.pdf>. Página vigente al 20/06/2013.

Sebastián Romero, 2010, Diseño basado en modelos de sistemas interactivos sociales. Cátedra Tecnología Educativa. Universidad de La Castilla - La Mancha. <http://www.sebastianromero.net/papers/designSIS.pdf>. Página vigente al 20/06/2013.

Victor Penichet, , Una aproximación al proceso de diseño e implementación de interfaces de usuario para aplicaciones groupware. Cátedra Tecnología Educativa. http://siteground171.com/~penichet/images/stories/profesional/investigacion/docs/publicaciones/penichet_Interaccion08-1.pdf. Página vigente al 20/06/2013.

Luisa Rodríguez, , Diseño de interfaces de usuario para aplicaciones colaborativas a partir de modelos independientes de la computación. Cátedra Tecnología Educativa. <http://www.aipo.es/aipo/articulos/1/12453.pdf>. Página vigente al 20/06/2013.

Victor Penichet, 2007, Análisis en un modelo de procesos CSCW. Organización, roles e interacción persona - ordenador - persona. Cátedra Tecnología Educativa. <http://aipo.es/articulos/1/12414.pdf>. Página vigente al 20/06/2013.

Fernando Molina, , JSEM-HP: Una herramienta para el desarrollo de sistemas hipermedia adaptativos y evolutivos. Cátedra Tecnología Educativa. <http://hera.ugr.es/tesisugr/20726533.pdf>. Página vigente al 22/06/2013.

B. Jimenez, , Inclusión de Objetivos en el modelado para la coordinación de procesos no-estructurados en sistemas colaborativos. Cátedra Tecnología Educativa. <http://lbd.udc.es/jornadas2011/actas/JISBD/JISBD/S7/Emergentes/JISBD11trabajo50postrevision.pdf>. Página vigente al 22/06/2013.

Manuel Noguera, 2009, Modelado y análisis de sistemas CSCW siguiendo un enfoque de ingeniería dirigida por ontologías. Cátedra Tecnología Educativa. <http://0-hera.ugr.es.adrastea.ugr.es/tesisugr/18014094.pdf>. Página vigente al 22/06/2013.

Victor Penichet, , Hacia una propuesta metodológica para entornos colaborativos. Cátedra Tecnología Educativa. Universidad De Castilla - La Mancha. <http://lsi.ugr.es/fguti/taller/06/penichet.pdf>. Página vigente al 22/06/2013.

Victor Penichet, 2007, Modelo de proceso para el desarrollo de interfaces en entornos CSCW centrado en los usuarios y dirigidos por tareas. Cátedra Tecnología Educativa. <https://www.google.com.ar/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CC4>

QFjAA&url=http%3A%2F%2Fwww.usixml.org%2Fservlet%2FRepository%2Fpenichet-phd2007.pdf%3FID%3D1773%26saveFile%3Dtrue&ei=uCrRUqWgCsjlsATRqYLgDA&usg=AFQjCNH5YxwUwz5SbAdIx-CIj7kzpW7oWQ&bvm=bv.59026428,d.cWc. Página vigente al 11/01/2014.

José Figueroa Martínez, 2012, Integración de los requerimientos de conciencia situacional y grupal al desarrollo de sistemas colaborativos y dinámicos usando un enfoque basado en modelos. Cátedra Tecnología Educativa. Universidad de Granada. <http://hera.ugr.es/tesisugr/21467821.pdf>. Página vigente al 16/02/2014.

Miguel Teruel Martínez, 2011, CSRML: un lenguaje para la especificacion de requisitos en sistemas colaborativos. Cátedra Tecnología Educativa. Universidad de Castilla-La Mancha. <http://ruidera.uclm.es:8080/xmlui/bitstream/handle/10578/2404/TFM%20Teruel%20Mart%C3%A9nez.pdf?sequence=3>. Página vigente al 16/02/2014.

M. Sánchez, Modelo de Control de Acceso en un sistema colaborativo. Cátedra Tecnología Educativa. Universidad de Granada. <http://aipo.es/articulos/4/24.pdf>. Página vigente al 16/02/2014.

Beatriz Jiménez, 2006, REPRESENTACION DEL MODELO DE TAREAS DE UN SISTEMA COLABORATIVO USANDO REDES HIPERMEDIA. Cátedra Tecnología Educativa. Universidad de Granada. <http://lsi.ugr.es/~fguti/taller/06/BJimenez.pdf>. Página vigente al 03/03/2014.

José Luis Isla, Descripción de Patrones de Organización y su Modelado con AMENITIES. Cátedra Tecnología Educativa. Universidad de Granada. <http://www.grise.upm.es/rearviewmirror/conferencias/jiisic04/Papers/2.pdf>. Página vigente al 29/04/2014.

José Luis Isla, Hacia un Catálogo de Patrones para el Modelado Conceptual de Sistemas Colaborativos. Cátedra Tecnología Educativa. Universidad de Granada. http://lbd.udc.es/jornadas2011/actas/JISBD/JISBD/S7/Regulares/jisbd2011_submission_26.pdf. Página vigente al 29/04/2014.