

# De los patrones de organización e interacción al diseño de interfaces de usuario colaborativas

P. González<sup>1</sup>, F. T. Granollers<sup>2</sup>, F. L. Gutiérrez<sup>3</sup>, J. L. Isla<sup>4</sup>, y F. Montero<sup>1</sup>

<sup>1</sup>Grupo LoUISE  
Universidad de Castilla-La Mancha  
Campus Universitario s/n  
02071, Albacete, España  
{pgonzalez, fmontero}@dsi.uclm.es

<sup>2</sup>Grupo GRIHO  
Universitat de Lleida  
Jaume II, 69  
25001, Lleida, España  
tonig@diei.udl.cat

<sup>3</sup>Dpto. de Lenguajes y  
Sistemas Informáticos  
Universidad de Granada  
C/Periodista Daniel  
Saucedo Aranda s/n  
18071. Granada, España  
fgutierr@ugr.es

<sup>4</sup>Dpto. de Lenguajes y  
Sistemas Informáticos  
Universidad de Cádiz  
C/ Ancha, 16  
11001, Cádiz, España  
joseluis.isla@uca.es

## Resumen

El desarrollo de productos software para dar soporte a las actividades tradicionalmente llevadas a cabo en grupo es una labor compleja. Para abordar el mencionado proceso deben facilitarse mecanismos con los que cubrir los necesarios requisitos de comunicación, coordinación y compartición de información de los que hacen gala los productos groupware.

Se han propuesto diferentes metodologías para el desarrollo de productos groupware [3], pero pocas incluyen la consideración de la experiencia documentada y disponible en diferentes ámbitos que son afines.

En este artículo se propone la utilización conjunta de experiencia en forma de patrones, concretamente la disponible con los patrones de organización y de interacción, para contribuir al desarrollo sistemático de interfaces de usuario destinadas a dar soporte al trabajo en grupo, siguiendo un desarrollo dirigido por modelos.

## 1. Introducción

El desarrollo de aplicaciones groupware impone unos requisitos adicionales al desarrollo del software que podríamos entender como tradicional o *monousuario*. Estos requisitos tienen que ver con la propia dinámica del trabajo en grupo. Al trabajar en grupo la calidad en uso precisa contemplar requisitos adicionales relacionados con la necesaria comunicación, coordinación y compartición de recursos, además de facilitar la mera actividad individual de los componentes del grupo de trabajo.

La principal motivación de este artículo es la de contribuir a la solución del problema que supone el desarrollo de aplicaciones groupware de calidad y, en particular, al modelado y diseño de IU adaptadas a la interacción en grupo.

Para lograr nuestro objetivo nos cuestionaremos la utilidad de la experiencia documentada y disponible en distintos ámbitos, especialmente aquella disponible en forma de patrones. El cuestionamiento estará relacionado con la limitación que, a nuestro modo de ver, supone una documentación *pasiva* de los patrones, es decir, una documentación que impide su puesta en práctica inmediata. Por ello, en este artículo, analizaremos qué notaciones podrían ser las más adecuadas para documentar distintos tipos de patrones. Nos vamos a centrar en los patrones de organización y de interacción, así como en su interrelación, ya que pensamos que uno de los elementos más importantes y diferenciadores del trabajo en grupo es la estructura organizativa que es necesaria para acometer el trabajo en grupo y que esta estructura se va a ver reflejada de forma directa en el diseño que se deba de realizar del IU.

Propondremos, en este artículo, el uso de diferentes notaciones para documentar los patrones considerados, porque entendemos que el Lenguaje Unificado de Modelado (UML) no ofrece facilidades directas para documentarlos.

Este artículo está organizado en diferentes secciones. En los primeros apartados se describen distintos tipos de patrones. Posteriormente, sugerimos la utilización conjunta de la experiencia disponible en esos dos tipos de patrones, a través de la integración de la experiencia que documentan utilizando diferentes notaciones que permitan su modelado. La propuesta recogida en este artículo se verá complementada con algún caso de

estudio con el que queremos demostrar la viabilidad. Finalmente, las conclusiones y los trabajos futuros pondrán fin a nuestro artículo.

## 2. Los patrones de organización

Cuando nos enfrentamos al desarrollo de un sistema cooperativo es fundamental estudiar cómo se organizan los distintos actores que cooperan dentro del proceso de negocio. Este estudio es necesario para poder obtener las actividades que realmente se llevarán a cabo dentro del sistema y, por ende, el diseño de los elementos a implementar.

Una buena forma de expresar este conocimiento es mediante la utilización de modelos que representen adecuadamente dicha estructura y comportamiento.

Si se analiza cómo trabajan las organizaciones se pueden identificar diferentes estructuras organizativas que, por su eficacia, a menudo rigen el contexto organizativo de un sistema. Este es el caso de estilos organizativos como los de estructura-en-5, joint venture, integración vertical, pirámide, etc., los cuales dotan a parte de la organización de una estructura que facilita la distribución

de sus integrantes (unidades organizativas o individuos) para conseguir objetivos globales.

Con objeto de favorecer la reutilización, así como facilitar la especificación y comprensión del contexto organizativo de un sistema, planteamos la utilización de patrones de organización [4] como un mecanismo idóneo para encapsular el conocimiento relacionado con dichas estructuras organizativas. Para modelar dichos patrones, proponemos la utilización conjunta de la notación COMO-UML, propia de la metodología AMENITIES [3] y basada en UML, junto a un profile que extiende UML para el modelado de patrones de software en general. Para describir los patrones de organización utilizamos el esquema recogido en [4].

### 2.1. El patrón JOINT VENTURE

A modo de ejemplo, vamos a modelar un patrón específico de organización, llamado “Joint Venture” (Figura 1), el cual representa una estructura organizacional típica procedente del ámbito de las alianzas estratégicas entre empresas.

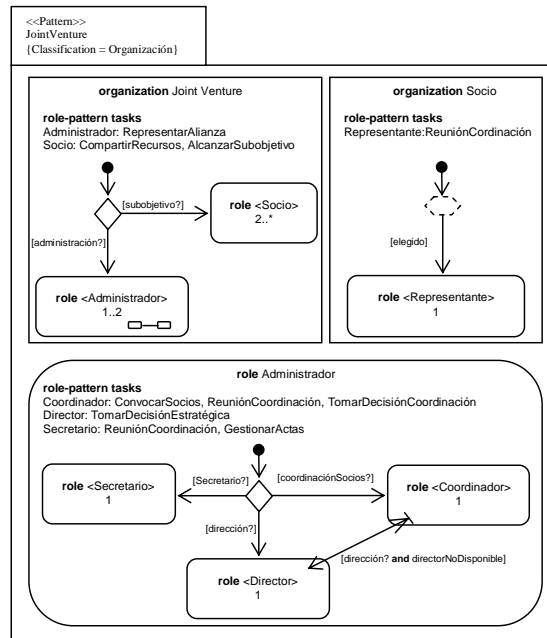


Figura 1. Estructura de la solución aportada por el patrón JOINT VENTURE

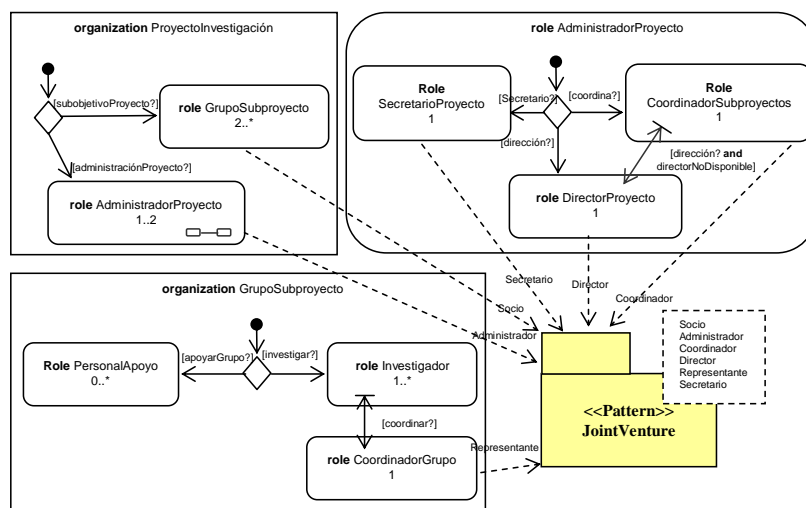


Figura 2. Utilización del patrón JOINT VENTURE en un ejemplo

Junto al patrón, presentamos su aplicación concreta en el modelado de la organización de un proyecto de investigación que incluye diversos grupos coordinados (Figura 2). Esta aplicación servirá de ejemplo para la sección 5.

### 3. Los patrones de interacción

Los patrones de interacción o de usabilidad no dejan de ser, al igual que los patrones de organización presentados en el apartado anterior, simplemente experiencia. Esta experiencia identificada y documentada está disponible para los diferentes individuos involucrados en el desarrollo de un producto software, resultando especialmente útil para los diseñadores y los usuarios finales.

Son numerosos los artículos, páginas web y libros dedicados a recopilar patrones de interacción. Habitualmente, los patrones de interacción se han documentado utilizando descripciones en lenguaje natural. Sin embargo, esta forma de documentarlos tiene sus ventajas e inconvenientes.

La principal ventaja es que permite a un mayor número de colectivos, especialmente a los usuarios finales o potenciales, conocerlos y expresarse en términos del nombre asociado a cada patrón, facilitando con ello su participación en el proceso de desarrollo de software y posibilitando la puesta en práctica de un verdadero diseño centrado en el usuario.

No obstante, en diferentes reuniones de trabajo relacionadas con el concepto de patrón de interacción, ha quedado de manifiesto la posibilidad de asociar a su descripción tradicional otras vistas del mismo. Estas otras perspectivas de un patrón de interacción, concretamente las relacionadas con su implementación, estructura de la solución o modelado de la misma, pueden verse como un complemento que facilita al equipo de desarrollo su puesta en práctica. Específicamente en el terreno de los patrones de interacción, nuestra experiencia demuestra que en muchos casos éstos documentan la necesidad de ofrecer al usuario *facilidades adicionales* que le permitan la realización de las tareas asociadas con los requisitos funcionales identificados. Estas facilidades adicionales suelen pasar por ser tareas o elementos de interfaz que contribuyen a la calidad de la interacción entre el hombre y la máquina, de ahí el nombre dual de estos patrones.

Una de las notaciones que venimos utilizando con éxito para asociar un modelo a la solución documentada en un patrón de interacción es la notación ConcurTaskTrees [2].

Seguidamente, se recoge un ejemplo de patrón de interacción, en concreto, el patrón LOGIN (disponible en <http://www.welie.com>). En este tipo de patrones, para capturar la estructura o modelo de la solución aportada, normalmente, se utilizan capturas de pantalla que muestran ejemplos de uso de la solución aportada. Además, pueden utilizar-

se notaciones relacionadas con el modelado de tareas y con el modelado de la presentación a diferentes niveles de abstracción. Dada la relación que puede identificarse entre una y otra notación,

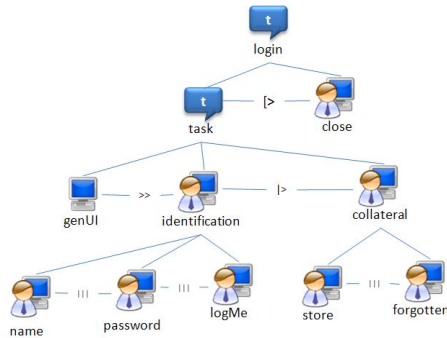
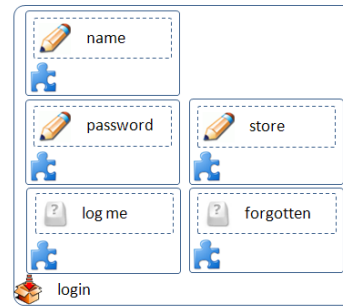


Figura 3. Diferentes especificaciones de la solución propuesta por el patrón LOGIN

podríamos derivar la presentación abstracta a partir de un modelo de tareas asociado con cada patrón [1] (véase Fig. 3).



#### 4. Integración de la experiencia en el desarrollo de interfaces de usuario colaborativas

En este artículo proponemos la utilización conjunta de la información disponible en los diferentes tipos de patrones comentados. Consideramos que dicha integración facilita el desarrollo de groupware y la calidad de la interfaz de usuario asociada a estas aplicaciones.

El uso efectivo de los patrones de organización e interacción surge en las primeras fases del proceso de desarrollo de un producto software. En estas primeras fases el equipo de desarrollo debe abordar labores de análisis y de diseño del producto software. Centrándonos en las labores propias del diseñador de interfaces de usuario, en esta fase inicial del proceso de desarrollo éste debe elaborar un prototipo de la interfaz a partir de la identificación de los actores y de los casos de uso. En estas labores el diseñador de interfaces cuenta con la ayuda de su experiencia y de las guías de estilo para la elaboración de interfaces de usuario.

Como se comentó en el apartado destinado a describir los patrones de organización, cuando un usuario trabaja en grupo tiende a desempeñar un conjunto de roles fuertemente relacionados con sus capacidades y competencias. Esa información, que debe especificarse para cada proyecto, en muchos casos es documentada por medio de patrones de organización que capturan la estructu-

ra y comportamiento de los distintos roles junto a sus responsabilidades. En función de esta información el diseñador de interfaces de usuario cuenta con un primer apoyo para desempeñar sus tareas, que según RUP son las mostradas en la Figura 4.

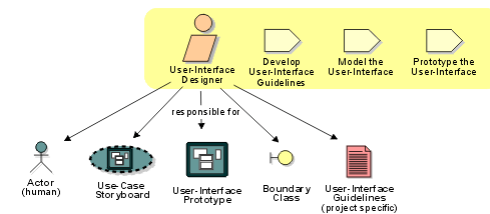


Figura 4. Tareas propias del diseñador de interfaces de usuario en RUP

Para abordar el diseño y prototipado de la interfaz de usuario, el diseñador de interfaces de usuario cuenta con otra información igualmente valiosa. Se trata de la documentada en forma de patrones de interacción. Con ella, y según se ha comentado, dispone de información relacionada con cómo abordar la realización de situaciones habituales y qué opciones de interfaz deben estar disponibles en esas ocasiones.

La principal motivación que este artículo persigue es la de contribuir al desarrollo de interfaces de usuario de calidad para aplicaciones groupware. En este sentido hay que recalcar que existen diferentes tipos de aplicaciones groupware que pueden clasificarse atendiendo a muy diferentes

critérios. Uno de los más recurrentes en la literatura es la consideración del lugar y el tiempo en el que realizan las tareas en grupo los integrantes del mismo. La Tabla 1 resume los diferentes tipos de aplicaciones groupware que pueden identificarse atendiendo a estos criterios, en ella se identifican cuatro tipos de aplicaciones:

- SC (síncrono-centralizado): aquellas aplicaciones que facilitan que todos los integrantes del grupo trabajen simultáneamente y en el mismo lugar.
- AC (asíncrono-centralizado): aquellas aplicaciones que facilitan que todos los integrantes del grupo trabajen en el mismo lugar, aunque en momentos diferentes.
- SD (síncrono-distribuido): aquellas aplicaciones que facilitan que todos los integrantes del grupo trabajen simultáneamente aunque estén en sitios diferentes.
- AD (asíncrono-distribuido): aquellas aplicaciones que facilitan que todos los integrantes del grupo trabajen conjuntamente aunque no puedan compartir ni tiempo ni espacio.

En términos generales, las herramientas groupware deben ofrecer facilidades para que un grupo de trabajo u organización desarrolle su labor de forma satisfactoria e independientemente de dónde se encuentren los recursos materiales o

personales de los que disponga el grupo. En este sentido, la interacción en grupo que surge debido a esta actividad conjunta viene determinada por la disponibilidad de facilidades relacionadas con el soporte a la comunicación, compartición y coordinación de actividades (véase Tabla 1).

Una vez comentados los tipos de interacción en grupo que pueden identificarse en función de la sincronía y localización de los integrantes del mismo y de asociarles diferentes soluciones tecnológicas a cada una de las posibilidades anteriores, el paso siguiente consiste en establecer una asociación entre esas mismas soluciones y una especificación basada en las tareas que debe realizar cada role en función de sus competencias y capacidades.

En este trabajo utilizamos fundamentalmente dos notaciones: ConcurTaskTrees [1] para el modelado de tareas, y usiXML [5] para el modelado de la presentación. Con ellas en el siguiente apartado veremos cómo es posible modelar y desarrollar prototipos de interfaz de usuario, utilizando la notación presentada en este apartado (Tabla 2) y la experiencia disponible, para una aplicación groupware destinada a dar soporte a la realización de reuniones de trabajo de una organización.

		Interacción en grupo		
		comunicación	compartición	coordinación
Interacción en el tiempo	síncrono	Chat, videoconferencia, teléfono SD	Editores multiusuario, SC, SD	MUD, SD
	asíncrono	email, AD	Acceso distribuido a bases de datos, AC, AD	Workflow, AC, AD

Tabla 1. Ejemplo de distintas herramientas para abordar el trabajo en grupo

Tareas	Elementos de presentación
<i>abstracta</i>	Independientemente de su <i>Container</i>
<i>interacción</i>	- Puede ser un <i>Container</i> cuando se puede descomponer en otras tareas - Puede ser un <i>Componente</i> cuando es nodo hoja en las especificación
<i>apliación</i>	- Puede ser un <i>Container</i> cuando se puede descomponer en otras tareas - Puede ser un <i>Componente</i> cuando es nodo hoja en las especificación
<i>usuario</i>	No tiene asignado ninguna presentación
<i>cooperación</i>	SD, AC, AD
<i>colaboración</i>	SD
<i>sesión ausente de interacción</i>	SC

Tabla 2. Equivalencias identificadas entre modelado de tareas y de presentación

## 5. Caso de estudio: La reunión

En esta sección queremos poner en práctica todo lo recogido hasta el momento en este artículo. Se trataría de comprobar empíricamente la utilidad y posibilidad de utilizar la experiencia disponible y documentada en forma de patrones. Los patrones a los que hemos prestado especial atención en este artículo son los patrones de organización y de interacción.

El supuesto práctico que utilizaremos sería el de una aplicación que deseamos de soporte al trabajo en grupo desarrollado por un conjunto de individuos que disfrutan de un proyecto de investigación conjunto. Este grupo de individuos deben mantener una interacción de grupo que posibilite alcanzar los objetivos del proyecto de investigación y para ello deben poder realizar sesiones de trabajo independientemente de su localización.

El grupo de trabajo, integrado por diferentes miembros que pueden pertenecer a diferentes grupos de investigación (subproyectos), desarrollará una actividad colaborativa y su organización

puede responder a la recogida en el patrón JOINT VENTURE (véase Figura 1 y 2). A modo de ejemplo vamos a estudiar la actividad encargada de realizar una reunión de coordinación entre los coordinadores de los diferentes subproyectos.

El supuesto práctico que utilizaremos sería el de una aplicación que deseamos de soporte al trabajo en grupo desarrollado por un conjunto de individuos que disfrutan de un proyecto de investigación conjunto. Este grupo de individuos deben mantener una interacción de grupo que posibilite alcanzar los objetivos del proyecto de investigación y para ello deben poder realizar sesiones de trabajo independientemente de su localización.

El grupo de trabajo, integrado por diferentes miembros que pueden pertenecer a diferentes grupos de investigación (subproyectos), desarrollará una actividad colaborativa y su organización puede responder a la recogida en el patrón JOINT VENTURE (véase Figura 1 y 2). A modo de ejemplo vamos a estudiar la actividad encargada de realizar una reunión de coordinación entre los coordinadores de los diferentes subproyectos.

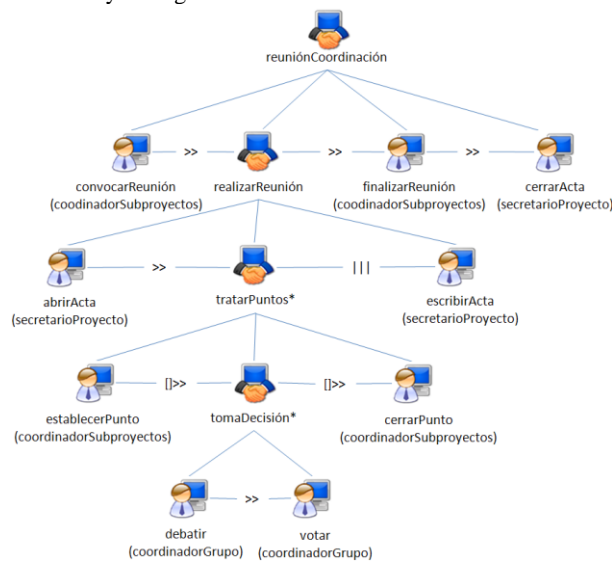


Figura 5. Especificación de las actividades realizadas conjuntamente por los distintos actores

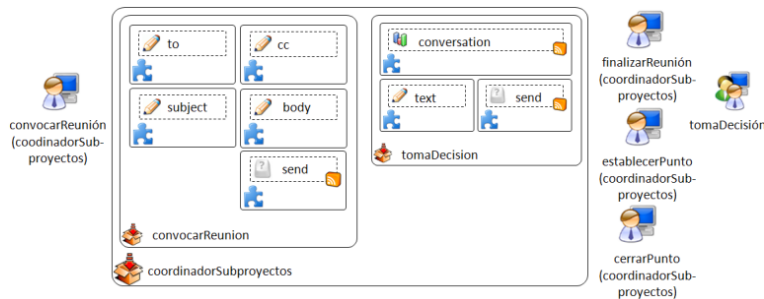


Figura 6. Modelo de presentación asociado al rol coordinadorSubproyectos

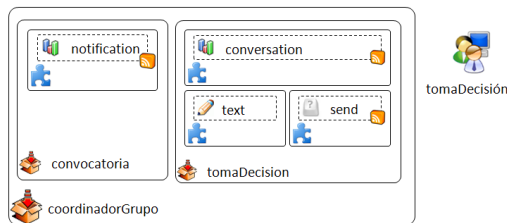


Figura 7. Especificación de la interfaz asociada al rol coordinadorGrupo

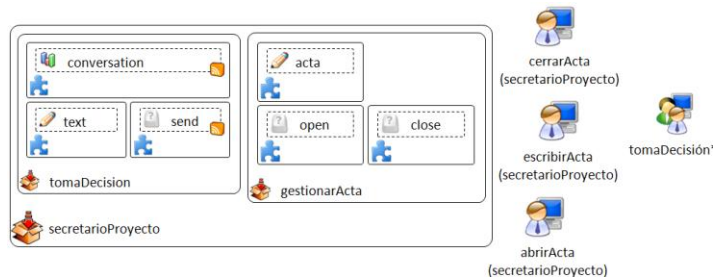


Figura 8. Modelo de presentación abstracto elaborado para el rol de secretarioProyecto

La dinámica de trabajo asociada a esta actividad hace que los roles que participan sean tres: el coordinador del proyecto (CoordinadorSubproyectos), el secretario (SecretarioProyecto) y los coordinadores de los diferentes grupos (CoordinadorGrupo).

En la Figura 5 se muestra un modelo de tareas asociado a la actividad en estudio (realizar una reunión para abordar una serie de puntos que se deben tratar y tomar decisiones en consecuencia), desarrollada conjuntamente por los roles presentados con anterioridad. A partir del modelo de tareas y de las equivalencias establecidas en la Tabla se pueden identificar necesidades a las que una aplicación groupware debería dar soporte. El coordinador del proyecto debe poder convocar reuniones y seguir el desarrollo de las mismas y

para ello hay diferentes posibilidades (p.e.: chat o videoconferencia). Del mismo modo, los coordinadores de cada grupo deberán poder interactuar entre ellos independientemente de su localización geográfica. Finalmente, la interfaz ofrecida al secretario deberá permitir levantar acta de las sesiones de trabajo, recoger los acuerdos alcanzados y las decisiones tomadas.

Suponiendo que la herramienta groupware que se pretende desarrollar quiera dar soporte la realización de reuniones virtuales de los miembros del grupo de investigación y que dichos miembros están en diferentes situaciones geográficas, el coordinador del proyecto sería el rol encargado de convocar a los miembros implicados en cada reunión. Dicha labor se supone que la realiza, al margen de otras posibilidades, utilizando el correo

electrónico. Para ello dispone de aquellos elementos de interfaz que caracterizan este mecanismo asíncrono de comunicación. Las otras tareas, que recaen sobre el rol coordinador del proyecto, es la asistencia a la reunión, la participación en la toma de decisiones y el estableciendo los puntos a tratar, cerrándolos cuando se supone que ya ha habido una razonable discusión sobre los mismos o cuando se considere cerrado su tratamiento. También pondrá fin a la reunión cuando no haya más puntos que tratar. Las actividades mencionadas se llevarán a cabo utilizando una especie de chat, que de manera simplificada aparece simbolizado en la parte derecha del *container* representado en la Fig. 6.

De forma semejante a la comentada para el rol coordinador del proyecto, los otros roles identificados (CoordinadorGrupo y SecretarioProyecto), así como las funciones que le son propias, que han sido identificadas gracias a la disponibilidad del patrón de organización JOINT VENTURE, pueden modelarse utilizando una notación basada en CTT. Desde esta notación se pueden derivar las especificaciones de interfaz que aparecen reflejadas en las figuras 7 y 8.

## 6. Conclusiones y trabajo futuro

El objetivo último de este artículo es contribuir al desarrollo de groupware de calidad siguiendo una metodología de desarrollo dirigida por modelos. Con ello en mente, el artículo ha resaltado las bondades de dos tipos de patrones: los de organización y los de interacción, para contribuir al desarrollo de las citadas interfaces. También se ha comprobado cómo es posible asociar a los patrones especificaciones que, más tarde, pueden utilizarse para modelar soluciones groupware.

Las contribuciones de los patrones de organización están relacionadas con la identificación de roles y de actividades ligadas a la necesaria dinámica de grupo que desempeñan dentro de la organización. Por otro lado, los patrones de interacción tienen la facultad de documentar facilidades funcionales y no explícitamente funcionales que la interfaz de usuario debería ofrecer para facilitar las tareas que debe realizar un usuario cuando utiliza un producto software.

Por la actividad presentada, con este artículo se dan los primeros pasos en la utilización conjunta y sistemática de la experiencia en el desarrollo

de interfaces de usuario para soportar el trabajo en grupo de estructuras organizacionales. Los próximos pasos deberán abordar la necesaria generación de interfaces de usuario finales a partir de las especificaciones abstractas elaboradas y la evaluación de calidad de las mismas.

En este sentido, tomamos en consideración el estándar ISO/IEC 9126 (que define la calidad interna y externa de una aplicación software) creemos oportuno evaluar la usabilidad como atributo fundamental que garantiza la calidad de uso de dicho sistema. Para ello se enlazará con el trabajo previo realizado en [6] que definen el procedimiento a seguir para evaluar la usabilidad de un tipo particular de sistemas colaborativos como son los sistemas Shared-Knowledge Awareness (SKA).

## Agradecimientos

La realización de este artículo está parcialmente financiados por el proyecto CICYT TIN2004-08000-C03-01.

## Referencias

- [1] Montero, F., López, V. IdealXML: an interaction design tool. Proceedings of the Sixth International Conference on Computer Aided Design of User Interfaces. 2006.
- [2] Paternò, F. (1999). Model-based design and evaluation of interactive applications. Applied Computing. Springer-Verlag London Limited.
- [3] Garrido, J.L. AMENITIES: Una metodología para el desarrollo de sistemas cooperativos basada en modelos de comportamiento y tareas, Tesis doctoral. Universidad de Granada. 2003.
- [4] Isla, J.L., Gutiérrez, F. Gea, M. Supporting Social Organization Modelling in Cooperative Work Using Patterns. In Shen, W. et al. (eds.), CSCW in Design II, LNCS 3865, Springer, pp. 112-121, 2006
- [5] Limbourg, Q., Vanderdonckt, J.: USIXML: A User Interface Description Language Supporting Multiple Levels of Independence. ICWE Workshops 2004: 325-338
- [6] González, M.P., Collazos, C., Granollers, T. "Guidelines and usability principles to design and test Shared-Knowledge Awareness for a CSCL interface". Proc. of 12th CRIWG. 2006.