

Diseño de interfaces de usuario para aplicaciones colaborativas a partir de modelos independientes de la computación

M^a Luisa Rodríguez Dept. de Lenguajes y Sistemas Informáticos ETS I. Informática Univ. de Granada 18071 Granada mlra@ugr.es	José Luís Garrido Dept. de Lenguajes y Sistemas Informáticos ETS I. Informática Univ. de Granada 18071 Granada jgarrido@ugr.es	Manuel Noguera Dept. de Lenguajes y Sistemas Informáticos ETS I. Informática Univ. de Granada 18071 Granada mnoguera@ugr.es	M^a Visitación Hurtado Dept. de Lenguajes y Sistemas Informáticos ETS I. Informática Univ. de Granada 18071 Granada mhurtado@ugr.es	José Ramón Polo Dept. de Lenguajes y Sistemas Informáticos ETS I. Informática Univ. de Granada 18071 Granada jrp1@correo.ugr.es
--	--	---	---	--

Resumen

El rápido desarrollo de la tecnología permite a las organizaciones operar en entornos de interacción donde el trabajo se organiza y asigna a grupos de personas que trabajan cooperativamente para alcanzar los objetivos que se proponen. En este tipo de sistemas groupware la interfaz de usuario es fundamental ya que debe facilitar el proceso de compartir información entre un grupo de personas que realiza una tarea común. El diseño de la interfaz de usuario es un proceso complejo que requiere comprender las tareas que el usuario debe realizar sobre el sistema y las características de los diferentes usuarios que lo manejan. Esto implica que la actividad de diseño de interfaces de usuario es crucial en el proceso de desarrollo de las aplicaciones colaborativas. Este artículo propone una aproximación al diseño de interfaces de usuario para aplicaciones colaborativas a partir de modelos independientes de la computación.

1. Introducción

Tradicionalmente el trabajo de una organización se realiza en grupos, donde sus miembros interaccionan para lograr una mayor productividad y rendimiento. El usuario forma parte de una comunidad conectada en red que permite la cooperación de sus miembros para desarrollar sus tareas. La disciplina CSCW (*Computer Supported Cooperative Work*) [1] estudia y analiza mecanismos de coordinación

para lograr una colaboración y una comunicación humana efectiva, así como los sistemas que la soportan. *Groupware* ha sido definido [5] como “*un sistema basado en computadoras que da soporte a grupos de personas implicadas en tareas (o metas) comunes y que proporciona una interfaz a un entorno compartido*”.

Los requisitos que de forma más específica se asocian con sistemas *groupware* [5] están fundamentalmente relacionados con las principales características que estos sistemas exhiben: comunicación, colaboración y coordinación.

La complejidad inherente de los sistemas CSCW implica una gran cantidad de esfuerzo en especificación y desarrollo [1]. El desarrollo de sistemas *groupware* es más difícil que el de una aplicación con un único usuario, ya que hay que tener en cuenta los protocolos sociales y las actividades de grupo para obtener un diseño satisfactorio [9]. Por tanto, deben ser aplicadas metodologías y técnicas de implementación que tengan en cuenta las actividades colaborativas, así como las interacciones persona-ordenador que pueden implicar estas actividades.

La interfaz de usuario de estos sistemas debe soportar que un grupo de usuarios realicen sus actividades colaborativamente. Esto no es solamente un problema de implementación sino también de diseño, ya que no hay métodos y notaciones para expresar el dinamismo de esta clase de interfaces.

Además, la interfaz gráfica de usuario es un aspecto clave para la usabilidad de los sistemas software, y en general de las aplicaciones

colaborativas ya que permite mejorar la comunicación, colaboración y coordinación entre los usuarios que interactúan con el sistema. Esto implica que la actividad de diseño de la interfaz gráfica de usuario es crucial en el proceso de desarrollo de los sistemas colaborativos. En este sentido es necesario contar con modelos independientes de la computación, que guíen todo el proceso de desarrollo.

HCI (Human Computer Interaction) [2] es una disciplina que se dedica al diseño, evaluación e implementación de sistemas informáticos para uso de los seres humanos y al estudio de todos los fenómenos relacionados con la comunicación. En este campo se han propuesto diferentes metodologías y entornos de desarrollo basados en modelos que permiten desarrollar la interfaz de usuario.

Existen algunas metodologías que han aunado los aspectos interactivos con los colaborativos pero sin cubrir todo el proceso de desarrollo de un sistema groupware, tanto de la propia aplicación como de la interfaz, de forma integrada. Este trabajo de investigación se centra en este problema analizando y proponiendo una aproximación al diseño de interfaces gráficas de usuario para aplicaciones colaborativas.

Este artículo está organizado en 5 secciones. La sección 2 realiza una breve revisión de algunas propuestas metodológicas relacionadas con interacción y sistemas colaborativos y se indican las motivaciones que han llevado a la realización de este trabajo de investigación. La sección 3 presenta nuestra propuesta para el diseño de interfaces gráficas de usuario para aplicaciones colaborativas. En la sección 4 se describen las características principales de una plataforma para el diseño e implementación de interfaces groupware. La sección 5 muestra un ejemplo donde se aplica nuestra propuesta. La última sección proporciona un breve resumen de las principales aportaciones del trabajo propuesto.

2. Estado del arte

La interfaz de usuario *“es la parte del sistema informático que permite al usuario acceder a las facilidades del ordenador”*[2]. Determina en gran medida la percepción e impresión que el

usuario posee de la aplicación, y de su diseño y características depende que el usuario acepte la aplicación.

A la hora de desarrollar las interfaces de usuario una de las técnicas más utilizadas es el diseño basado en modelos, cuyo propósito es identificar modelos de alto nivel que permitan a los diseñadores especificar y analizar los sistemas software desde un nivel orientado a la semántica más que comenzar inmediatamente con el nivel de implementación [16].

Toda la información necesaria para desarrollar la interfaz de usuario está explícitamente representada en modelos declarativos. Por modelo se entiende una especificación de la estructura y comportamiento de cualquier elemento software. Un modelo es declarativo porque no contienen código sino descripciones a alto nivel de abstracción [19].

La ventaja principal de este método es la expresividad que pueden aportar estos modelos. Algunos de estos modelos declarativos [20] son el modelo de información, modelo de usuario, modelo de tareas, modelo de diálogo, modelo de presentación, modelo de aplicación, modelo espacio de trabajo, etc.

Existen distintas aproximaciones al desarrollo de interfaces de usuario provenientes del campo de la interacción persona-ordenador, como SUIP (Scenarios-based User Interface Prototyping) [6], IDEAS (Interface Development Environment within OASIS) [14] y UMLi [18].

Ninguna de las anteriores aproximaciones considera los sistemas colaborativos aunque se han desarrollado otras propuestas que aúnan los aspectos interactivos con los sistemas colaborativos. En [13] se utilizan escenarios, requerimientos y máquinas de estado para verificar la interacción de varios actores. En [4] se propone un método para especificar interacciones colaborativas entre los usuarios a partir de los escenarios, y diagramas de secuencia para modelar las interacciones. Trabaja con casos de uso, escenarios y diagramas de colaboración. CEM (CSCW Environments Methodology) [17] es una metodología que se fundamenta en MDA (Model-driven Architecture) como iniciativa para el desarrollo de sistemas mediante modelos, AOSD (Aspect Oriented Software

Development), AMENITIES (A Methodology for Analysis and Design of Cooperative Systems), IDEAS, CTT (ConcurTaskTrees), patrones y UML. Por su parte, CIAM (Collaborative Interactive Applications Methodology) [15] 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. Para especificar la parte interactiva utilizan la notación CTT. Otra metodología que aborda de forma sistemática el análisis y diseño de un sistema colaborativo facilitando el desarrollo posterior del software es AMENITIES [7]. Se basa en modelos de comportamiento y tareas para el análisis y diseño de éstos sistemas. Permite realizar un modelado conceptual del sistema colaborativo; centrándose en el grupo y cubriendo aspectos relevantes de su comportamiento (dinamismo, evolución, etc.) y estructura (organización, leyes, etc.). La especificación obtenida contiene información relevante (componentes, tareas de usuario, elementos de dominio, diálogos persona-máquina y persona-persona, presentaciones, etc.) para la creación de la interfaz de usuario, sin embargo actualmente no incluye ninguna propuesta para obtener la interfaz de usuario. Este aspecto es considerado de gran interés, por ello en el siguiente apartado se propone una aproximación a la construcción de la interfaz de usuario para un sistema colaborativo en base a la obtención de un conjunto de modelos.

3. Una aproximación al diseño y desarrollo de interfaces de usuario para aplicaciones colaborativas.

En esta sección proponemos un nuevo enfoque para el diseño de la interfaz de usuario para aplicaciones colaborativas. Primero vamos a describir los fundamentos de nuestra propuesta,

después estableceremos las conexiones entre los modelos cooperativos y los modelos de interfaz de usuario y, por último esbozaremos una descripción general del proceso de diseño, como observamos en la Figura 1.

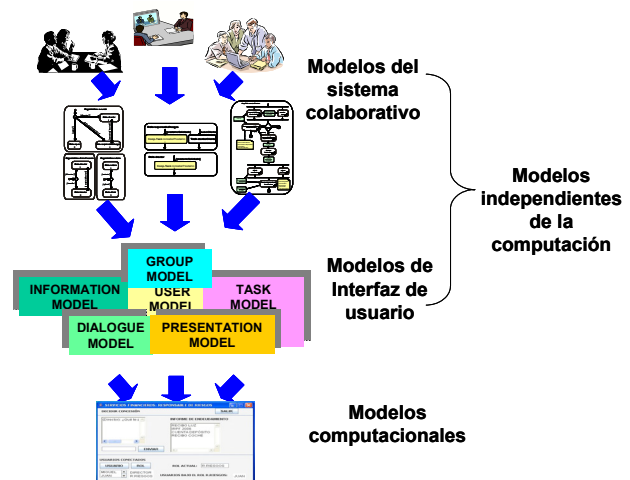


Figura 1. Proceso de obtención de la interfaz

3.1. Fundamentos

La aproximación que se propone, parte de la descripción de un sistema colaborativo que se realiza en la metodología AMENITIES. Dicha descripción consta de dos grupos de modelos [8]:

1. Modelos de requisitos: El proceso de elicitación de requisitos se realiza principalmente con la técnica de etnografía aplicada y los diagramas de casos de uso, aunque pueden ser utilizadas otras técnicas complementarias, como escenarios y *brainstorming*.
 - Etnografía: Permite reunir y documentar informalmente los requisitos de la interfaz de usuario desde el punto de vista del usuario.
 - Diagramas de casos de uso: Estructuran y especifican los requisitos funcionales, roles y actores del sistema.

2. Modelo cooperativo: Es un modelo conceptual que describe y representa la estructura y funcionamiento de un sistema colaborativo. Para su construcción de forma jerárquica, en base a modelos específicos que proporcionan diferentes vistas para un mismo sistema cooperativo, se aplica un método que consta de cuatro etapas:

- Especificación de la organización: Se definen las organizaciones que intervienen en el sistema en base a los roles de cada organización, leyes que especifican los cambios dinámicos entre roles y capacidades de los actores. También se puede especificar eventos y acciones asociadas a los roles.
- Definición de roles: Se definen las tareas que están asociadas a cada rol, distinguiendo entre tarea individual y cooperativa. Además, para cada tarea se especifica: conjunto de tareas que pueden interrumpir su realización, eventos que la disparan, leyes que habilitan su realización por miembros del grupo, acciones en las transiciones de entrada y salida de la tarea, objetos de información requeridos y multiplicidad necesaria para desarrollar la tarea.
- Definición de tareas: Se describen cada una de las tareas identificadas anteriormente, indicando: los roles/actores que colaboran en su realización (sólo para tareas cooperativas), los objetos de información utilizados, los artefactos necesarios para realizar la tarea, los protocolos de interacción entre los participantes, número de veces que se puede repetir, restricciones temporales, su descomposición en subactividades, tareas mixtas o internas.
- Especificación de protocolos de interacción: Se definen las características de los protocolos, las subactividades asociadas, los roles participantes y las leyes que tienen que cumplir.

A partir de la información descrita en los dos modelos anteriores, se derivarán los modelos básicos sobre los que se fundamentará la construcción de la interfaz de usuario para las aplicaciones colaborativas a desarrollar. Las relaciones entre los distintos modelos facilitan el proceso de obtención del interfaz de usuario a

través de un proceso de transformación de modelos [8], como se puede observar en la Figura 1. De este modo, se define un conjunto de modelos encargados de recoger las características intrínsecas de estos sistemas, es decir, tanto el aspecto del trabajo individual como en grupo. El proceso de trabajo en grupo debería ser tan natural como cuando se trabaja individualmente, aunque en la práctica los procesos de trabajo en grupo son más complejos que los individuales ya que se ha de considerar el comportamiento del grupo en base a las dinámicas presentes en las relaciones sociales, como entorno, implicación del participante, procedimiento de trabajo en grupo, conciencia de grupo y cognición distribuida.

3.2. Modelos para el diseño de la interfaz de usuario

Según el marco de trabajo de partida, los principales modelos utilizados para la construcción de la interfaz incluirán la siguiente información:

1. Modelo de información: Define el dominio de la información sobre la que se va a trabajar.
 - Objetos de información de la aplicación.
 - Acciones que se realizan sobre ellos.
 - Objetos de entrada a las tareas.
 - Objetos que generan las tareas.
2. Modelo de Usuario: Describe las características del usuario final de la aplicación.
 - Tareas individuales asociadas a cada rol.
 - Capacidades de los actores.
3. Modelo de Grupo: Describe las características de los grupos de usuarios de la aplicación.
 - Tareas cooperativas asociadas a cada rol.
 - Contexto y entorno para cada usuario: grupo al que pertenece, artefactos compartidos, usuarios que están conectados en cada momento, rol desempeñado por cada usuario, usuarios con el mismo rol, roles que puede desempeñar, etc.

4. Modelo de Tareas: Define todas las tareas específicas que los usuarios necesitan realizar en la aplicación.
 - Tipos de tareas (individuales, cooperativas, mixtas, externas e internas).
 - Roles asociados a las tareas.
 - Interrupciones entre tareas.
 - Objetos de información utilizados.
 - Orden temporal de realización de las tareas.
 - Número de veces que se ejecutan las tareas.
5. Modelo de diálogo: Especifica la estructura sintáctica de la interacción persona-ordenador.
 - Acciones sobre el sistema interactivo.
 - Protocolos de interacción.
6. Modelo de Presentación: Determina los componentes que pueden aparecer en la interfaz de usuario, sus características de diseño y las dependencias visuales existentes entre ellas.
 - Componentes necesarios para la interfaz y sus relaciones.

En la Tabla 1 se indican las relaciones entre los modelos que describen el sistema colaborativo y los que permiten diseñar la interfaz de usuario.

3.3. Proceso de diseño

El objetivo del proceso de diseño es obtener como resultado modelos de computación específicos que son de especial interés en la construcción de las interfaces de usuario. Las relaciones entre los modelos que hemos descrito en la Tabla 1, nos permiten identificar los principales elementos de los sistemas interactivos. A su vez, estos elementos y sus relaciones se transformarán en elementos de un modelo computacional. Por ejemplo, en los diagramas de organización se especifican los cambios dinámicos entre los roles de cada organización. Esta relación implica un mecanismo en la interfaz de usuario para reflejar el cambio dinámico, como puede ser un componente gráfico que permita seleccionar el nuevo rol de una lista disponibles de roles. Además, habrá que adaptar la interfaz de usuario al nuevo estado incorporando las nuevas tareas de ese rol (por ejemplo, con botones para

seleccionarlas). Es decir, la interfaz de usuario cambia su estructura y comportamiento según los requerimientos del sistema cooperativo.

Por último, se construye una interfaz de usuario a partir de un conjunto de componentes estándar y específicos de una plataforma para sistemas groupware, la cual describiremos en la siguiente sección.

4. Plataforma de soporte al desarrollo de GUI colaborativas

Las interfaces de los sistemas groupware deben permitir que un grupo de personas realice su trabajo de forma simple y natural. Deben por tanto, permitir la interacción:

- Entre el usuario y el sistema de forma que se puedan efectuar tareas individuales y cooperativas.
- Entre los miembros del mismo grupo que realizan tareas cooperativas.

Además, deben dar soporte a la conciencia de grupo (*group awareness*), que se define como “*el entendimiento y conocimiento de las actividades de los otros miembros del grupo que proporciona el contexto para llevar a cabo la propia actividad*” [3]. Es decir, incluir información sobre quién está utilizando el sistema, dónde están trabajando y qué están haciendo. También es importante que las acciones de un usuario se muestren a los otros usuarios que colaboran en la misma tarea (*feedthrough*) [2]. Por ejemplo, observar a otra persona navegar a través de los elementos de un menú suministra indicios sobre lo que está haciendo o quiere hacer.

Estos requisitos resaltan la importancia de proporcionar el soporte adecuado a la construcción de interfaces para aplicaciones colaborativas. Para ello, se ha desarrollado una plataforma de componentes gráficos (multi-user widgets) accesibles a través de una API [12], con la intención de simplificar el desarrollo de estas interfaces en base a extensiones de componentes estándares (botones, menús, campos de texto, etc.) y de componentes específicos para sistemas groupware (telepunteros, chat, lista de usuarios conectados, etc.). Los componentes que se proponen son replicados permitiendo que varios usuarios estén interactuando con el componente de forma

MODELOS DE AMENITIES	MODELOS PARA LA INTERFAZ DE USUARIO					
	Información	Usuario	Grupo	Tareas	Diálogo	Presentación
Etnografía aplicada	X	X	X	X	X	X
D. Casos de uso		X		X		X
Especificación organización		X	X			X
Definición de roles	X	X	X	X		X
Definición de tareas	X	X	X	X	X	X
Protocolos de interacción			X	X	X	X

Tabla 1. Relación entre la descripción del sistema y los modelos de interfaz

simultánea manteniendo un estado consistente global. En la Tabla 2 se describen los componentes creados en la plataforma. La API también consta de un módulo de metainformación dinámico que permite gestionar los cambios producidos en la aplicación. Trabajos similares se encuentran en MAUI (Multi-user Awareness [11]

que proporciona un toolkit de widgets para sistemas groupware con componentes estándares y específicos, como telepunteros y lista de participantes. En [10] se centran en estudiar la navegación en el espacio de trabajo, la interacción con los componentes y las distintas vistas de trabajo individual y en grupo.

Nombre	Funcionalidad	Propiedades semánticas	Comentarios
<i>DIList</i>	Selección, adición y sustracción de elementos de la lista	No visible, lectura, escritura	Replicado o local
<i>DIComboBox</i>	Apertura y cierre de la lista. Resaltado y selección de un elemento	No visible, lectura, escritura	Replicado
<i>DIButton</i>	Pulsar y soltar el botón	No visible, lectura, escritura	Replicado
<i>DIToggleButton</i>	Pulsar y soltar el botón	No visible, lectura, escritura	Replicado
<i>DICheckBox</i>	Pulsar y soltar la checkbox	No visible, lectura, escritura	Replicado
<i>DITextField</i>	Inserción y eliminación de texto	No visible, lectura, escritura	Replicado
<i>DITree</i>	Abrir y cerrar los niveles del árbol. Selección de un elemento	Lectura, escritura	Replicado
<i>DJMenu</i>	Indica el usuario que actúa sobre el menú	No visible, lectura, escritura	Replicado
<i>DIChat</i>	Chatear con los usuarios conectados	No visible, lectura, escritura	Distribuido
<i>DIRoleChange</i>	Indica el rol actual y permite cambiar a otro rol	Lectura, escritura	Replicado
<i>DICurrentRole</i>	Rol desempeñado en cada momento	Lectura, escritura	Replicado
<i>DIOnlineUserList</i>	Usuarios conectados en cada momento	Lectura, escritura	Replicado
<i>DIUserListRolePlayed</i>	Usuarios conectados en cada momento y el rol desempeñado	Lectura, escritura	Replicado
<i>DIUserListSameRole</i>	Usuarios conectados con el mismo rol	Lectura, escritura	Replicado
<i>Telepointer</i>	Realimentación de los movimientos de otros usuarios	Lectura, escritura	Local (escritura) Replicado (lectura)

Tabla 2. Descripción de los componentes gráficos implementados en la plataforma

5. Caso de estudio

De cara a aplicar la propuesta, consideraremos un caso de estudio basado en un sistema de ayuda a la decisión para operaciones de riesgo de entidades financieras. El primer paso en la actividad de concesión de un préstamo hipotecario consiste en realizar un estudio de viabilidad de cada propuesta creando un expediente que recoja toda la información. En los préstamos hipotecarios, a la vista de los ingresos, gastos y deudas se calcula el nivel de endeudamiento. Para realizar el estudio de viabilidad se necesitan los datos personales del cliente (Nombre, Apellidos, DNI), el patrimonio, rentas, efectos bancarios impagados y cuentas. Como resultado de este estudio, la entidad opta por aprobar, denegar o suspender la operación. En caso de aprobación de la operación es necesario contar con una cuenta para vincular el préstamo.

La construcción del modelo colaborativo de este ejemplo sigue el método que consta de cuatro etapas que se describen a continuación.

5.1. Especificación de la organización

El caso de estudio incluye tres organizaciones: sucursal, oficina de tasación y notaría. El diagrama de las tres organizaciones se muestra en la Figura 2.

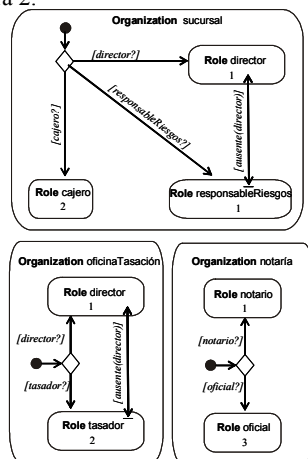


Figura 2. Diagrama de organización

La organización *sucursal* está compuesta por tres roles: *Director*, *Responsable de riesgos* y *Cajero* según su especialización o categoría profesional, que viene especificado por las capacidades (*director?*, *responsableRiesgos?* y *cajero?*). Además, algunos miembros de la Oficina pueden cambiar dinámicamente de rol debido a una serie de circunstancias. Por ejemplo, la organización establece la regla que en ausencia del Director el Responsable de riesgos puede sustituirlo, los que especificamos como la transacción aditiva *[ausente(director)]*. Lo que se traduce en la interfaz de usuario al componente *DIRoleChange* (1) de la Figura 5.

5.2. Definición de roles

Una vez descrita la organización tenemos que especificar cada uno de los roles que la constituyen, como podemos observar en la Figura 3.

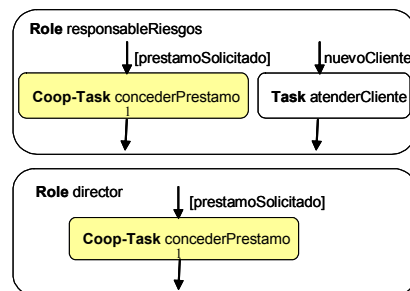


Figura 3. Diagramas de roles

El diagrama de roles para el Responsable de riesgos especifica que puede realizar dos tareas: la tarea individual *atender un cliente* y la tarea cooperativa *conceder un préstamo hipotecario*.

5.3. Definición de tareas

Vamos a describir la tarea cooperativa *conceder un préstamo hipotecario* a través del diagrama de tareas de la Figura 4. La tarea se ha dividido en subactividades modelando así los pasos a seguir para su realización. Especificamos las restricciones temporales de realización de las subactividades con flechas (secuencialidad) y

fork-join (conurrencia). También indicamos qué rol tiene que desempeñar cada una de las subactividades (por ejemplo, la subactividad *Decidir concesión* la realizan el Director y el Responsable de riesgos).

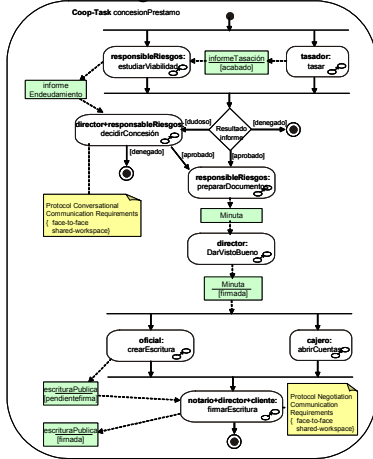


Figura 4. Diagrama de tareas

La tarea de concesión de un préstamo es una tarea colaborativa y, por tanto, todos los usuarios que participan deben conocer quién está trabajando en ese momento. La interfaz, Figura 5, tiene que recoger la información necesaria sobre el contexto de trabajo de esta tarea, lo que hacemos con los componentes replicados desarrollados, lista usuarios conectados y su rol *DIUserListRolePlayed* (2), lista usuarios conectados con el mismo rol *DIUserListSameRole* (3) y rol actual *DICurrentRole* (4).

Figura 5: Interfaz con conciencia de grupo

En el diagrama de tareas de la subactividad Estudio de viabilidad, Figura 6, incluimos los objetos de información que utilizan las subactividades como datos de entrada (renta, impagados, cuentas e Informe de tasación) y los documentos de salida que genera (Informe de endeudamiento).

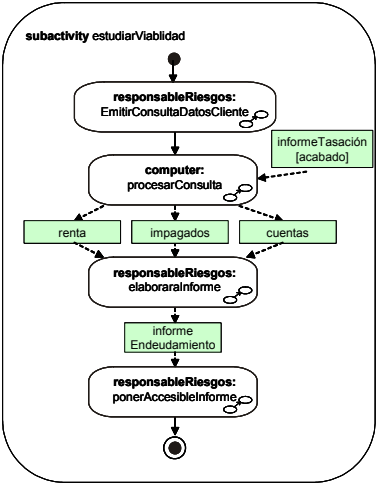


Figura 6. Diagrama de Estudio de viabilidad

La interfaz de usuario para esta subactividad la podemos observar en la Figura 7:

Figura 7. Interfaz de Estudiar viabilidad

5.4. Especificación de protocolos de interacción

En la Figura 8 podemos observar la interfaz de usuario de subactividad *Decidir concesión* que refleja un espacio compartido de trabajo (el

MODELO COOPERATIVO	MODELOS DE INTERFAZ DE USUARIO						MODELO COMPUTACIONAL
	Información	Usuario	Grupo	Tareas	Diálogo	Presentación	
Diagrama de organización							
Cambio de rol		X	X	X		X	<i>DIRoleChange</i>
Diagrama de tareas							
Tareas cooperativas			X	X	X	X	<i>DIUserListRolePlayed</i> <i>DIUserListsameRole</i> <i>DICurrentRole</i>
Restricciones temporales				X	X	X	Componentes deshabilitados
Subactividades				X	X	X	Componentes de selección (Botones)
Datos de entrada	X			X	X	X	Componentes de edición Componentes de selección
Datos de salida	X			X	X	X	Componentes de listado
Protocolos de interacción							
Tipos				X	X	X	<i>DIChat</i> Video conferencia
Espacio compartido de trabajo				X	X	X	<i>Telepointer</i> Texto compartido

Tabla 3. Transformación entre modelos para el caso de estudio

Informe de evaluación) y un componente “*DIChat* (1)” para implementar la interacción entre el Responsable de riesgos y el Director. Además, también se observa un componente “*Telepointer* (2)” sobre el Informe de endeudamiento que corresponde a la acción del Director.



Figura 8. Interfaz de Decidir concesión

La Tabla 3 muestra algunas de las conexiones entre los modelos, las cuales hemos descrito en el caso de estudio.

6. Conclusión

Este trabajo presenta una propuesta para el diseño de interfaces de usuario para aplicaciones colaborativas, que integra el proceso de obtención de la interfaz con el desarrollo del sistema groupware. La interfaz de usuario se obtiene a partir de la especificación del sistema colaborativo en un proceso de transformación de un conjunto de modelos independientes de la computación. Esta propuesta nos permite obtener interfaces de usuario para aplicaciones colaborativas que soporten y potencien la colaboración humana. Las interfaces adaptan su estructura y comportamiento a los requerimientos de estos sistemas. Además, hemos presentado una plataforma de soporte al desarrollo de interfaces de usuario colaborativas que facilita y simplifica su obtención.

Agradecimientos

Este trabajo está financiado por el proyecto del CICYT TIN2004-08000-C03-02.

Referencias

- [1] Beaudouin-Lafon, M. (ed). Computer Supported Cooperative Work. Université Paris.Sud, France, John Wiley&Sons, 1999.
- [2] Dix, A., Finlay, J., Abowd, G., Beale, R. Human-Computer Interaction. Prentice-Hall, 1998.
- [3] Dourish, P., Belloti, V. Awareness and Coordination in Shared Workspaces. Proceeding of the ACM Conference on Computer-Supported Cooperative Work, 1992.
- [4] Dumont, A., Pietrobon, C. A Method for Specification of Collaborative Interfaces through the Use of Scenarios. Fifth International Conference on Computer Supported Cooperative Work in Design. 2001.
- [5] Ellis, C.A., Gibbs, S.J., Rein, G.L. Groupware: Some Issues and Experiences. Communications of the ACM. Vol. 34, nº 1, 1991
- [6] Elkoutbi, M., Khriess, I., Keller, R.K. Generating User Interface Prototypes from Scenarios. Proceedings of the Fourth IEEE International Symposium on Requirements Engineering, 1999.
- [7] Garrido, J.L., Gea, M., Rodríguez, M.L. Requirements engineering in cooperative systems, in: Requirements Engineering for Sociotechnical Systems, Idea Group, Inc., USA, 2005 (Chapter XIV).
- [8] Garrido, J.L., Noguera, M., González, M., Hurtado, María V., Rodríguez, María L. Definition and use of Computacional Independent Models in an MDA-based groupware development process. Science of Computer Programming, Volume 66, Issue 1, April 12007, Pages. 25-43.
- [9] Grudin, J. Groupware and Cooperative Work: Problems and Prospects. In Baecker, R.M. (ed.) Readings in Groupware and Computer Supported Cooperative Work, San Mateo, CA, Morgan Kaufman Publishers (1993).
- [10] Gutwin, C., Greenberg, S. Design for Individuals, Design for Groups: Tradeoff between power and workspace awareness. Proceeding of ACM CSCW'98, 1998.
- [11] Hill, J.M. A Direct Manipulation Toolkit for Awareness Support in Groupware. Thesis, University of Saskatchewan, 2003.
- [12] Ibáñez Santórum, J.A. Diseño e implementación de una plataforma para el desarrollo de sistemas groupware. Proyecto fin de carrera. Dpto. Lenguajes y Sistemas Informáticos, Universidad de Granada, 2006.
- [13] López Nores, M., et al. Formal specification applied to multiuser distributed services: Experiences in collaborative t-learning. The Journal of Systems and Software, Volume 79, Issue 8, 2006.
- [14] Lozano, M. Entorno Metodológico Orientado a Objetos para la Especificación y Desarrollo de Interfaces de Usuario. Tesis Doctoral, Universidad Politécnica de Valencia, 2001.
- [15] Molina, A.I., Redondo, M.A., Ortega, M. Conceptual and Methodological Framework for Modelling Interactive Groupware Applications. Comunicación corta en 12th International Workshop on Groupware. LNCS 4154, Springer-Verlag, 2006.
- [16] Paternò, F. Model-Based Design and Evaluation of Interactive Applications. Springer, 1999.
- [17] Penichet, V., Lozano, Maria D., Gallud, J.A. Hacia una Propuesta Metodológica para Entornos Colaborativos. III Taller de Sistemas Hipermedia Colaborativos y Adaptativos, 2005.
- [18] Pinheiro da Silva, P., Paton, N.W. UMLi: The Unified Modeling Language for Interactive Applications. LNCS Vol. 1939. Springer Verlag, 2000.
- [19] Puerta, A.R. A Model-Based Interface Development Environment. IEEE Software, 1997.
- [20] Schlunbaum, E. Model-based User Interface Software Tools Current state of declarative models. Research Report, Georgia Institute of Technology, Graphics, Visualizations & Usability Center, GIT-GVU-96-30, 1996.