

DESARROLLO DE APLICACIONES DE TRABAJO COOPERATIVO MEDIANTE JAVA Y LA WWW

Carlos Ley, Alvaro Suárez

*GAC (Grupo de Arquitectura y Concurrency)
Departamento de Ingeniería Telemática, Universidad de Las Palmas de Gran Canaria
E-mail: {alvaro, cley}@cic.teleco.ulpgc.es*

En los últimos años la potencia de proceso de los ordenadores personales y las estaciones de trabajo ha experimentado un considerable incremento, que ha venido además acompañado de un aumento - en cantidad y calidad - de las redes de comunicación de datos utilizadas para la interconexión de los mismos. Todo ello ha traído consigo la aparición de un nuevo tipo de aplicaciones, diseñadas para facilitar la cooperación entre usuarios que colaboren en una tarea con el fin de alcanzar un objetivo común. Por otro lado, el continuo auge que la red Internet viene experimentando ha ocasionado aparición de nuevas tecnologías y aplicaciones, entre las que brilla con luz propia la World Wide Web. En consecuencia, en la actualidad se viene planteando cada vez más la realización de aplicaciones para trabajo cooperativo basadas en la WWW, dadas las ventajas que ésta puede aportar. En este artículo, presentamos los principales aspectos de un modelo de aplicación de trabajo cooperativo basada en la WWW, haciendo énfasis en dos aspectos fundamentales en el mismo: el control de concurrencia y la forma de acceso a la información compartida. Finalizamos el artículo presentando una aplicación para edición cooperativa de documentos de texto, desarrollada para evaluar y verificar en la práctica las ideas en que se basa nuestro modelo.

1. INTRODUCCIÓN

El término «Trabajo Cooperativo Asistido por Computador» viene utilizándose desde hace ya algunos años para designar a las aplicaciones que facilitan la realización de un trabajo en común a un grupo de usuarios, que comparten un mismo objetivo. A este tipo de aplicaciones se las conoce normalmente por las siglas CSCW, acrónimo de la expresión inglesa Computer Supported Cooperative Work [Kup98].

También se suele utilizar el término Groupware, aplicado a los sistemas basados en computador que proporcionan - a un grupo de usuarios - una interfaz de acceso a un entorno compartido. En general, el término Groupware se utiliza más para definir a los propios sistemas basados en computador, mientras que el término CSCW engloba tanto el estudio de las aplicaciones Groupware como las implicaciones psicológicas, sociales y de organización que se desprenden de su uso.

Como ejemplos ilustrativos de aplicaciones CSCW podemos citar a los sistemas de videoconferencia para múltiples participantes, los foros de discusión, los juegos para múltiples jugadores, y las aplicaciones cooperativas que permiten la edición de gráficos o documentos de forma compartida. Como puede verse, el campo abarcado por estas aplicaciones es muy amplio, lo cual se refleja en el gran número de ellas que han sido o están siendo desarrolladas.

En los últimos años, muchas de las tendencias experimentadas por la tecnología de los computadores y las formas de organización empresarial han potenciado el auge de los sistemas CSCW. En primer lugar, los sistemas de computación vienen experimentando un cambio en su estructura, que está evolucionando de un modelo puramente centralizado - basado en unidades centrales de gran capacidad de proceso - a un modelo más distribuido, basado en grupos interconectados de estaciones de trabajo y/o computadores personales.

Se está produciendo una diseminación de la capacidad de procesamiento en los sistemas actuales, cuyos elementos aparecen interconectados mediante redes de computadores, que pueden abarcar desde un área geográfica local (redes LAN) hasta áreas de amplia extensión (redes WAN).

Por otro lado, las actuales tendencias en las formas de organización empresarial representan también un factor importante en el auge de los sistemas CSCW. En la actualidad, las organizaciones empresariales se encuentran cada vez más distribuidas, y no sólo en cuanto a diversidad en ubicaciones geográficas, sino en sus formas de interrelación y colaboración con otros individuos y organizaciones.

Si analizamos las características de los sistemas CSCW nos encontraremos con una gran diversidad de aspectos a tener en cuenta. Así, podemos considerar cuestiones relativas a la forma de interacción entre los usuarios, como son la conciencia de grupo, la comunicación, y la forma de coordinación entre los mismos durante

la realización del trabajo en común. Pero no menos importantes son los aspectos propios del diseño de las aplicaciones, como el control de concurrencia, y el acceso a la información que se maneja de forma compartida entre los usuarios.

Habitualmente se suele establecer una clasificación de los sistemas CSCW, según cómo se materialice la cooperación entre sus usuarios en el tiempo o en el espacio. Así, podemos establecer cuatro grandes categorías de clasificación: sistemas síncronos (que permiten la cooperación simultánea) asíncronos (en los que la cooperación se realiza en momentos diferentes) cara a cara (cooperación en el mismo lugar) y distribuidos (en lugares diferentes).

Una característica habitual en muchos de los sistemas CSCW desarrollados hasta la fecha es el hecho de ser concebidos como prototipos de laboratorio, con un ámbito de utilización más bien restringido. Esta característica puede representar un inconveniente si pensamos en la aplicación de tales sistemas para la colaboración de usuarios geográficamente muy dispersos. En tales sistemas, supone una clara ventaja el hecho de que sean desarrollados utilizando tecnologías y plataformas que disfruten de una amplia implantación.

En la actualidad, es indudable que entre tales tecnologías se encuentra la WWW (World Wide Web), cuyas extensas posibilidades se ven aún más ampliadas gracias al uso del lenguaje de programación Java, con el que podemos realizar fácilmente una gran variedad de aplicaciones capaces de funcionar integradas con la WWW.

En el resto de este artículo expondremos las principales ideas que hemos desarrollado para el diseño de un modelo de aplicación CSCW, basada en la WWW [Ben97] y desarrollada en el lenguaje de programación Java

2. LA WWW COMO PLATAFORMA PARA APLICACIONES CSCW

Una característica de la WWW que la hace muy adecuada para nuestros fines es el hecho de que ésta posee una arquitectura cliente-servidor relativamente simple, con programas clientes - navegadores - disponibles para las plataformas y sistemas operativos más comunes (Windows, UNIX, Linux, etc.). Al mismo tiempo, también es destacable el hecho de que exista una amplia diversidad de programas servidores WWW para dichas plataformas.

Si a todo esto añadimos la inherente capacidad de ejecución multiplataforma de las aplicaciones realizadas en el lenguaje Java, podemos concluir que gracias a la WWW y al lenguaje Java disponemos de un entorno de desarrollo adecuado para la implantación de aplicaciones CSCW que puedan ejecutarse en sistemas heterogéneos,

interconectados a través de redes de computadores diversas.

Sin embargo, hay también aspectos de la arquitectura cliente-servidor de la WWW que pueden condicionar o limitar el funcionamiento de nuestras aplicaciones CSCW. En primer lugar, consideremos la forma en que se realiza el intercambio de datos entre el cliente y el servidor. Dicha comunicación se realiza principalmente a través del protocolo HTTP (HyperText Transfer Protocol), en el que las comunicaciones entre cliente y servidor se inician normalmente a partir de un mensaje (una solicitud HTTP) enviado por el cliente, a la cual contesta el servidor (con una respuesta HTTP).

El hecho de que el servidor en una aplicación CSCW no pueda iniciar las comunicaciones con los clientes supone una clara limitación, que incidirá especialmente en la forma en que los clientes sean informados acerca de los cambios que se realicen en la información compartida que se encuentre almacenada en el servidor.

Otro aspecto de la WWW a considerar es su arquitectura, en la que en se considera normalmente un único servidor que centraliza las peticiones de los clientes. Esto hace que las aplicaciones CSCW basadas en la WWW sean potencialmente vulnerables frente a los posibles fallos en el funcionamiento del servidor. En definitiva, al plantearnos utilizar la WWW como plataforma sobre la que desarrollar nuestras aplicaciones CSCW, hay que considerar en qué forma afectan sus diferentes características a la aplicación CSCW que pretendamos diseñar.

3. PRINCIPIOS DE DISEÑO DE UNA APLICACIÓN CSCW BASADA EN LA WWW

En este apartado expondremos las principales características que nos hemos planteado para el diseño de una aplicación CSCW, basada en la WWW y desarrollada en Java. Nos centraremos sobre todo en dos aspectos fundamentales para las aplicaciones CSCW: el control de concurrencia y la forma de acceso a la información compartida, susceptible de ser accedida concurrentemente por múltiples usuarios para consulta y/o modificación.

En nuestro modelo de aplicación CSCW, el servidor es el encargado de almacenar la información compartida, y controlar los accesos concurrentes de los clientes a la misma. La información que se maneja de forma compartida en nuestro modelo tiene la característica estar organizada jerárquicamente, de forma que el servidor se emplean estructuras de datos en árbol para el control de acceso a la misma. Como ejemplo ilustrativo, pensemos en cómo representar un feto humano mediante una estructura en forma de árbol. Podríamos representar cada una de sus partes principales (cabeza, piernas, brazos, etc.) con un nodo del árbol, situado bajo su raíz (en el nivel más alto de la jerarquía).



A un nivel inferior, pueden situarse otros nodos que representen las secciones de cada parte principal del feto (la cabeza tendrá los ojos, la nariz, la boca, etc.) y así sucesivamente. Trabajando en un entorno de programación orientado a objetos, diseñamos cada elemento (o nodo de cada árbol) como un objeto, que representa a una parte de la información compartida y con el que se supervisan los accesos concurrentes que se realizan sobre la misma.

Pero el servidor no es sólo responsable de la gestión de las estructuras de control de acceso, sino también de la propia información compartida. Para garantizar la integridad de la misma, el servidor ha de sincronizar las operaciones de lectura/escritura que se realicen sobre dicha información, como consecuencia de las peticiones de acceso a la misma que se reciben desde los clientes.

Para permitir la realización de modificaciones sobre la información compartida, nuestro modelo emplea un sistema mediante el que los usuarios pueden bloquear el acceso a determinadas partes de dicha información. Para ello, los usuarios solicitan al servidor la reserva de las partes de información compartida que desean modificar.

Las solicitudes de reserva se reciben en el servidor procedentes de los clientes, y generadas a requerimiento de los usuarios que desean modificar partes de la información compartida. Al recibir una solicitud de reserva, el servidor comprueba en primer lugar si el usuario solicitante posee los derechos adecuados para acceder a la información requerida. Si es así y la información solicitada no se encuentra reservada, se modifica el estado de los nodos del árbol afectados - que pasarán a la condición de reservados - y se notifica al cliente sobre el resultado de su solicitud.

Desde el momento en que el cliente recibe esta confirmación, el usuario puede modificar las partes de la información compartida que ha reservado. Para ello se transfieren del servidor al cliente los datos reservados, modificándose los mismos de forma local en el cliente. Cuando el usuario ha realizado las modificaciones, envía los nuevos datos al servidor, el cual actualiza con ellos la información compartida. El proceso termina cuando el usuario libera la información compartida que ha reservado. Las etapas básicas del proceso de reserva y modificación de la información compartida se resumen en la Figura 1.

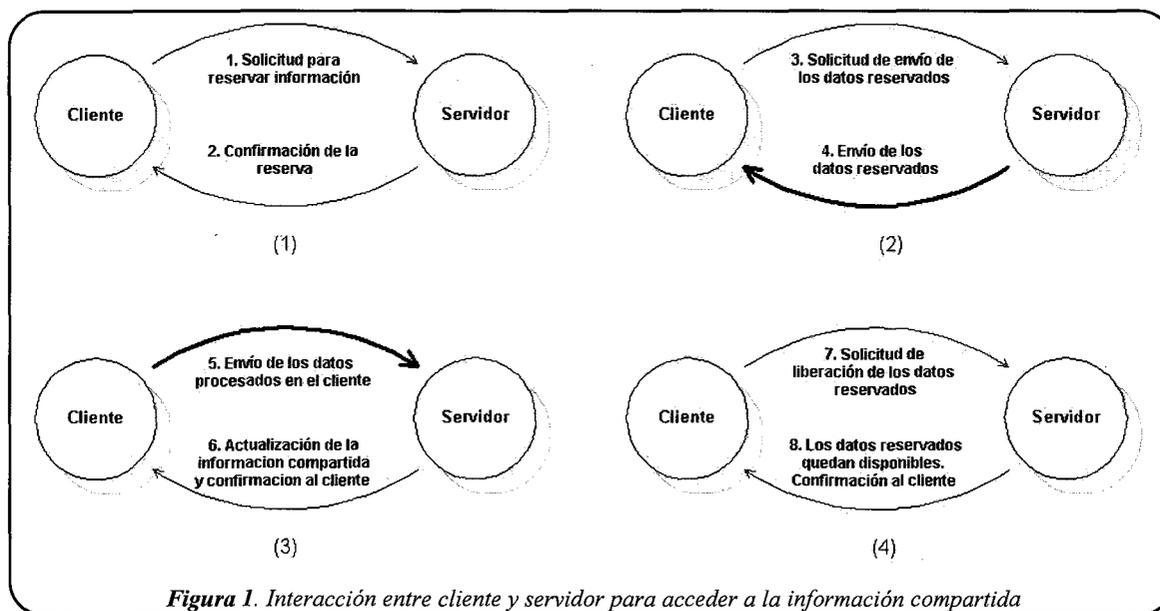


Figura 1. Interacción entre cliente y servidor para acceder a la información compartida

Una reserva hecha en el servidor hace que éste bloquee el acceso a la información reservada, de forma que nadie - excepto el usuario propietario de la reserva - pueda realizar modificaciones sobre la misma.

Siguiendo con nuestro anterior ejemplo, se podría obtener acceso exclusivo para modificar toda la información de un feto humano reservando todos los nodos de su estructura, desde el nodo raíz hasta el último nivel en su jerarquía. También sería posible bloquear solamente la información correspondiente a la cabeza, tronco, u otras partes, solicitando la reserva de los correspondientes nodos en la estructura del árbol.

Debido a la limitación ya indicada en la comunicación cliente-servidor - que imposibilita que el servidor pueda enviar mensajes a los clientes sin un requerimiento previo de éstos - hemos previsto un mecanismo de propagación asíncrono para notificar los cambios en la información compartida. Así, cuando un usuario desea conocer los cambios realizados sobre la información compartida debe solicitarlo al servidor, el cual procede entonces a enviar al cliente la información actualizada.

También hemos previsto mecanismos de supervisión de las reservas realizadas en el servidor, con el fin de afrontar los fallos que puedan surgir durante el funcionamiento de la aplicación, evitando que la información

reservada quede bloqueada permanentemente si se pierde la comunicación entre el servidor y los clientes que hayan efectuado reservas. Al mismo tiempo, hemos dotado al cliente con capacidad de almacenamiento local como medida de prevención ante posibles fallos en la comunicación de éste con el servidor.

El modelo de aplicación CSCW que proponemos se aplica con éxito en un gran número de aplicaciones. Como ejemplo, supongamos una herramienta cooperativa para desarrollo de software, en la que cada estructura en árbol del servidor puede representar un proyecto software. Dentro de cada árbol, sus nodos pueden representar diferentes partes dentro de la jerarquía de un proyecto: subproyectos, módulos, y así hasta bajar a los archivos de código fuente. En tal entorno de desarrollo los usuarios podrían editar, compilar y verificar concurrentemente diferentes partes de un mismo proyecto software.

Otro ejemplo podría ser una aplicación de gestión y edición cooperativa de documentos de texto. En este caso cada estructura en árbol representará un conjunto de documentos de texto organizados jerárquicamente, que pueden ser editados concurrentemente. Este segundo ejemplo ha sido precisamente el tipo de aplicación que hemos desarrollado como primer intento de transformación de nuestro modelo en una aplicación real, y que describiremos en los siguientes apartados.

4. UNA APLICACIÓN DE EJEMPLO: UN EDITOR COOPERATIVO PARA DOCUMENTOS DE TEXTO

La aplicación de edición cooperativa de documentos de texto que aquí presentamos ha sido desarrollada para poner en práctica nuestras ideas para el diseño de una aplicación CSCW, basada en la WWW y desarrollada en Java. Conforme ha las ideas que hemos expuesto en nuestro modelo, el servidor almacena y controla el acceso a los documentos, que son editados de forma compartida entre los clientes. Cada uno de los archivos de texto almacenados y controlados por el servidor tiene asociado un objeto, que denominamos objeto documento, y que contiene los datos necesarios para el control de acceso al mismo por parte de los clientes.

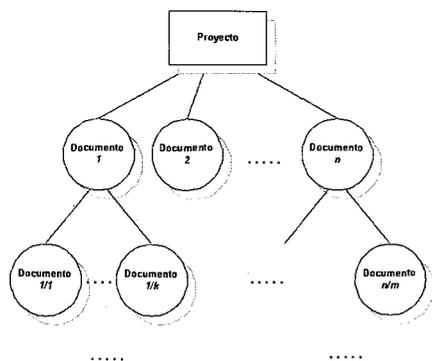


Figura 2. Estructura jerárquica de documentos

Estos objetos documento están organizados de forma jerárquica, y constituyen los nodos de las estructuras en árbol que el servidor utiliza para realizar el control de acceso a la información compartida. Las estructuras en árbol que contienen los objetos documento reciben en esta aplicación la denominación de proyectos. De esta forma, los objetos documento pertenecientes a un proyecto representan sus archivos de texto. La organización descrita de proyectos y documentos se ilustra en la Figura 2.

Por medio de un sistema de reservas, basado en los principios que ya hemos indicado, los usuarios modifican en exclusión mutua la información contenida en los archivos de texto del servidor. Pueden realizarse reservas a distintos niveles dentro de la jerarquía en que está organizada la información:

- A nivel de proyecto: se reservan todos los archivos de texto cuyos objetos documento están incluidos en el árbol de un determinado proyecto.
- A nivel de documento: reservando a discreción los objetos documento correspondientes a los archivos deseados.
- A nivel de partes de un documento: se reservan fragmentos de texto dentro de un archivo de un documento.

Siguiendo el modelo de aplicación CSCW propuesto, hemos utilizado el lenguaje Java para desarrollar nuestro programa cliente, que tiene la capacidad de funcionar como un Applet - ejecutado desde un navegador WWW estándar - o como una aplicación Java independiente. Por su parte, el programa servidor ha sido desarrollado como un Servlet [Sun00] (un Servlet es un programa Java que se ejecuta integrado con un servidor WWW).

En la Figura 3 se resumen la arquitectura básica de nuestra aplicación, con sus principales elementos y las relaciones básicas entre los mismos.

La ventana principal del cliente se muestra en Figura 4. En el panel izquierdo de la misma podemos ver la lista de proyectos existentes en el servidor, mientras que en el panel derecho se muestran los nodos del árbol de los documentos pertenecientes al proyecto seleccionado.

El color de los iconos que representan los documentos varía según el estado de los mismos (disponible,

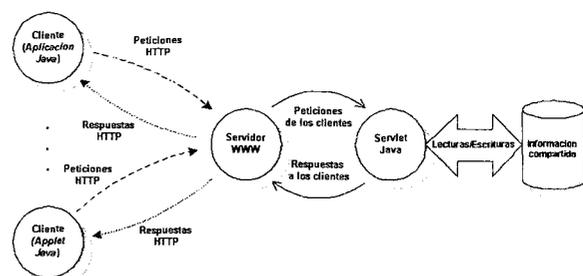


Figura 3. Componentes de la aplicación desarrollada



Figura 4. La ventana principal del cliente de nuestra aplicación

reservado total o parcialmente, etc.). A través de la ventana principal del cliente los usuarios pueden realizar diversas acciones básicas sobre los proyectos y sus documentos (crear, borrar, reservar, editar, liberar, etc.).

En la figura 5 se muestra la ventana de reservas, a través de la cual el usuario puede reservar fragmentos de texto dentro de un documento. Para editar una parte del texto de un documento hay que seleccionar el área de texto deseada y realizar la solicitud de reserva al servidor. Las partes de texto del archivo que se encuentran ya reservadas se muestran en caracteres de color rojo, además, el usuario puede obtener información sobre las reservas existentes y los usuarios que las han realizado. Desde el momento en que el servidor acepta una reserva - hecha sobre un documento o una parte del mismo - el texto reservado es editado en una ventana específica para la edición de dicho texto en el cliente.

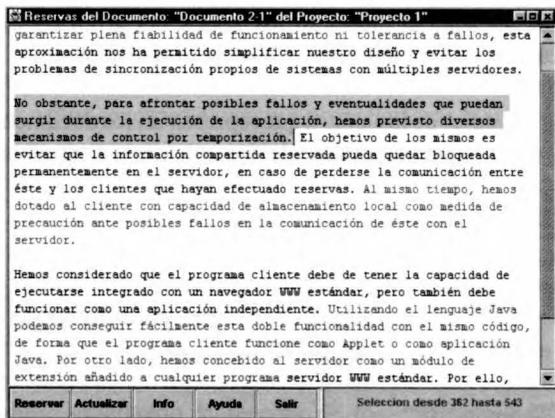


Figura 5. Ventana de reserva de partes del texto de un documento

5. CONCLUSIONES

El uso de la WWW y el lenguaje de programación Java aporta indudables ventajas para el desarrollo de aplicaciones para trabajo cooperativo. Y estas ventajas son aun más evidentes cuando se trata de aplicaciones CSCW diseñadas para la cooperación entre usuarios que utilizan sistemas de computación heterogéneos, y se encuentran distribuidos en áreas geográficas dispersas.

En este artículo hemos presentado un modelo de aplicación CSCW con arquitectura cliente-servidor, basada en la WWW y desarrollada en Java. Hemos descrito sus principales características, incidiendo sobre todo en los mecanismos que se emplean para realizar el control de concurrencia y el acceso a la información compartida. En nuestro modelo, dicha información compartida se caracteriza por tener una estructura jerárquica, estructura que puede aplicarse con éxito en una diversidad de aplicaciones.

Para finalizar, presentamos una aplicación de ejemplo desarrollada para poner en práctica los principales aspectos de nuestro modelo, aplicados a la edición cooperativa de documentos de texto.

6. BIBLIOGRAFÍA

- [Kup98] E. Kupstas Soo. «CSCW on the Web» <http://www.cs.unc.edu/~kupstas/csw.html>. 1998.
- [Ben97] R. Bentley, T. Horstman, J. Trevor. «The World Wide Web as anabling technology for CSCW: The case of BSCW». The Journal of Collaborative Computing: Special issue on CSCW and the Web, 2-3, 1997.
- [Sun00] Sun Microsystems. «Java Servlet API» <http://java.sun.com/products/servlet/index.html>. Febrero 2000