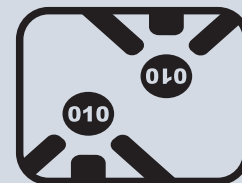


PROPUESTA ARQUITECTÓNICA PARA EL DESARROLLO DE APLICACIONES COLABORATIVAS DE CALIDAD



AUTOR

MARIO ANZURES GARCÍA

Estudiante de doctorado del Departamento de Lenguajes y Sistemas Informáticos.

Profesor Investigador de la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla

anzures@correo.ugr.es,

manzures@siu.buap.mx

MÉXICO

AUTOR

MIGUEL J. HORNOS BARRANCO

Doctor en Informática
Grupo GEDES del Departamento de Lenguajes y Sistemas Informáticos, E.T.S. de Ingenierías Informática y de Telecomunicación, Universidad de Granada

mhornos@ugr.es

ESPAÑA

AUTOR

PATRICIA PADEREWSKI RODRIGUEZ

Doctora en Informática
Grupo GEDES del Departamento de Lenguajes y Sistemas Informáticos, E.T.S. de Ingenierías Informática y de Telecomunicación, Universidad de Granada

patricia@ugr.es

ESPAÑA

Fecha de Recepción del Artículo: 20 de Octubre de 2006 **Fecha de Aceptación del Artículo 4 de Diciembre de 2006**
Artículo Tipo 2

RESUMEN.

Este artículo describe una propuesta arquitectónica basada en servicios web orientada al desarrollo de aplicaciones colaborativas de calidad. La arquitectura ha sido diseñada para abordar las carencias que presentan los modelos arquitectónicos y entornos existentes para el desarrollo de este tipo de aplicaciones. Estas limitaciones incluyen dependencias de plataformas, lenguajes y sistemas operativos, así como el hecho de que su funcionalidad, a menudo, está pensada para aplicaciones muy concretas. Nuestra propuesta, gracias a su diseño y a que está basada en servicios web, es capaz de proporcionar aplicaciones con atributos no funcionales (específicamente, atributos de calidad) como reusabilidad, portabilidad, interoperabilidad, ubicuidad y adaptabilidad, permitiendo reorganizar dinámicamente los componentes para adaptar la arquitectura a la forma en que se organiza el trabajo en grupo en cada momento. Dicha adaptación puede lograrse, si es necesario, extendiendo la propuesta con nuevos servicios (que pueden ser aplicaciones, componentes o herramientas) cuando éstos se requieran, de manera consistente y sin necesidad de realizar modificaciones en los servicios existentes. Además, el grupo puede adaptarse a nuevas necesidades o estilos de trabajo, gracias a que nuestra arquitectura permite modificar las políticas de manejo de sesión en tiempo de ejecución. Nuestra propuesta también resuelve problemas significativos encontrados en los modelos y entornos existentes para el desarrollo de aplicaciones colaborativas, como el hecho de que no proporcionan sesiones flexibles y/o no suministran un modelo consistente para integrar componentes de terceros.

PALABRAS CLAVE

Arquitectura Software

Atributos de Calidad

Aplicaciones Colaborativas

Servicios Web

Arquitectura Orientada a Servicios

ABSTRACT

This article describes a web service-based architectural proposal oriented to the development of quality collaborative applications. The architecture has been designed to overcome the lacks that existing architectural models and environments for developing this type of applications present. These

limitations include platform, language and operative system dependencies, and also the fact that their functionality is often intended for very specific applications. Our proposal, thanks to its design and since it is based on web services, is able to provide applications with non functional attributes (specifically, quality attributes) such as reusability, portability, interoperability, ubiquity and adaptability, which allow to dynamically reorganize the components in order to adapt the architecture to the form in which groupwork is organized at each moment. This adaptation can be achieved, if necessary, extending the proposal with new services (which can be applications, components or tools) when these are required, in a consistent way and without necessity to modify existing services. Moreover, the group can adapt to different needs and work styles, thanks to our architecture allows to modify the session management policies in run time. Our proposal also solves significant problems found in the existing models and environments for the development of collaborative applications, as the fact that they do not provide flexible sessions and/or they do not provide a consistent model to integrate third-party components.

KEYWORDS

Software Architecture
Quality Attributes
Collaborative Applications
Web Services
Service-Oriented Architecture (SOA)

INTRODUCCIÓN

El avance de Internet y de la Web ha fomentado que el trabajo se lleve a cabo en grupos cuyos miembros pueden estar distribuidos en diferentes lugares geográficos y pertenecer a la misma o diferente organización, tener distinta cultura, etc. Por todo ello, se requieren sistemas que permitan el desarrollo de aplicaciones CSCW (Computer Supported Cooperative Work) acordes a las necesidades actuales y, al mismo tiempo, que dichos sistemas estén soportados por modelos, metodologías, arquitecturas y/o plataformas.

Se han desarrollado una gran variedad de productos, prototipos y aplicaciones para dar soporte al trabajo en grupo, sin embargo, cada uno de ellos está orientado a la resolución de una situación colaborativa particular. Como ejemplos, podemos citar: los servicios de mensajería instantánea, como ICQ y Yahoo! Messenger; los sistemas de archivos compartidos punto-a-punto, como Napster y Gnutella; las aplicaciones groupware asincrónicas, como correo electrónico, sistemas de flujo de trabajo, calendarios de grupo y sistemas hipertexto; y las aplicaciones groupware síncronas, como la pizarra compartida, los chats, los sistemas de reuniones electrónicas, los sistemas de soporte a la decisión y los juegos multiusuario.

También existe gran variedad de modelos arquitectónicos y entornos que facilitan el desarrollo de aplicaciones colaborativas, también conocidas como groupware, término que hace referencia a sistemas basados en computadora que asisten a grupos de personas comprometidas en una tarea (o

meta) común y que proporcionan una interfaz para un entorno compartido [8]. El desarrollo de este tipo de aplicaciones se fundamenta en diferentes enfoques, como por ejemplo, orientado a procedimientos, objetos, aspectos, componentes y agentes, y recientemente se está usando SOA (Service-Oriented Architecture) [10] para construir aplicaciones colaborativas distribuidas. SOA proporciona una interfaz abstracta que suministra un conjunto de servicios débilmente acoplados, asíncronos y basados en documentos. Esto hace que las aplicaciones resultantes sean reutilizables, eficientes y adaptables [10]. La manera más común de implementar SOA es a través de servicios web, ya que al estar desarrollados bajo un conjunto de protocolos y tecnologías estándares y abiertas, permiten desarrollar aplicaciones interoperables, portables y fáciles de integrar [9]. Por esto, la industria informática se centra cada vez más en la utilización de los servicios web como una alternativa para construir sistemas abiertos distribuidos sobre Internet, dado que cualquier programa accesible vía web puede ser envuelto (wrapped) como servicio web. Además, la tecnología de servicios web es ideal para implementar trabajo colaborativo, puesto que se basa en la construcción de nuevas aplicaciones a través de la combinación de servicios disponibles en la red.

En el desarrollo de modelos arquitectónicos para sistemas colaborativos se echa en falta un modelo centrado en el trabajo de grupo, que soporte las tres características claves de los sistemas CSCW: comunicación, coordinación y colaboración [8], y que permita desarrollar aplicaciones colaborativas de calidad. Este artículo presenta una propuesta arquitectónica basada en servicios web para el desarrollo de este tipo de aplicaciones.

El resto del artículo se organiza como sigue: la sección 1 presenta la propuesta arquitectónica que planteamos, la sección 2 describe los atributos de calidad que obtienen las aplicaciones basadas en dicha arquitectura, la sección 3 comenta los trabajos relacionados, y finalmente se presentan las conclusiones y trabajos futuros.

1. PROPUESTA ARQUITECTÓNICA

Esta propuesta permite desarrollar aplicaciones colaborativas de calidad, flexibles y extensibles, estando pensada para superar las carencias de los modelos arquitectónicos actuales. La Figura 1 presenta el esquema general de la arquitectura, donde se muestran sus principales capas, módulos y servicios. El diseño de nuestra propuesta permite separar los aspectos colaborativos de los aspectos funcionales de la aplicación, gracias a la utilización de una estructura en capas. Existen dos capas: la Capa de Grupo, que contiene toda la información relacionada con los aspectos del trabajo de grupo y comprende los módulos de Manejo de Sesión y de Control de Acceso Compartido y los servicios de Conciencia de Grupo y de Registro, y la Capa de Aplicación, que permite agregar aplicaciones, herramientas y/o componentes, envueltos (wrapped) como servicios web, a la arquitectura, sin necesidad de modificar los servicios que dicha arquitectura ya tiene. Ambas capas son independientes, pero están interconectadas a través de los mecanismos de comunicación.

1.1 CAPA DE GRUPO

Ésta es la capa central de la arquitectura, ya que en ella se llevan a cabo todas las actividades relacionadas con el trabajo de grupo. Por ello, en primer lugar se proporcionan al usuario mecanismos que le permiten registrarse en el sistema, con el fin de iniciar el trabajo en grupo. En segundo lugar, se establece un espacio de trabajo compartido, creándose sesiones lo suficientemente flexibles para adecuarse a las diversas formas de organizar el trabajo en grupo. En tercer lugar, se definen un conjunto de políticas para facilitar la interacción entre los usuarios y con los recursos compartidos, con el fin de evitar conflictos que lleven a la inconsistencia de la información. Y en cuarto lugar, se suministran mecanismos que den información al usuario de lo que están haciendo los demás y de lo que está pasando en el espacio de trabajo compartido. A continuación se describen los elementos que realizan cada una de estas tareas.

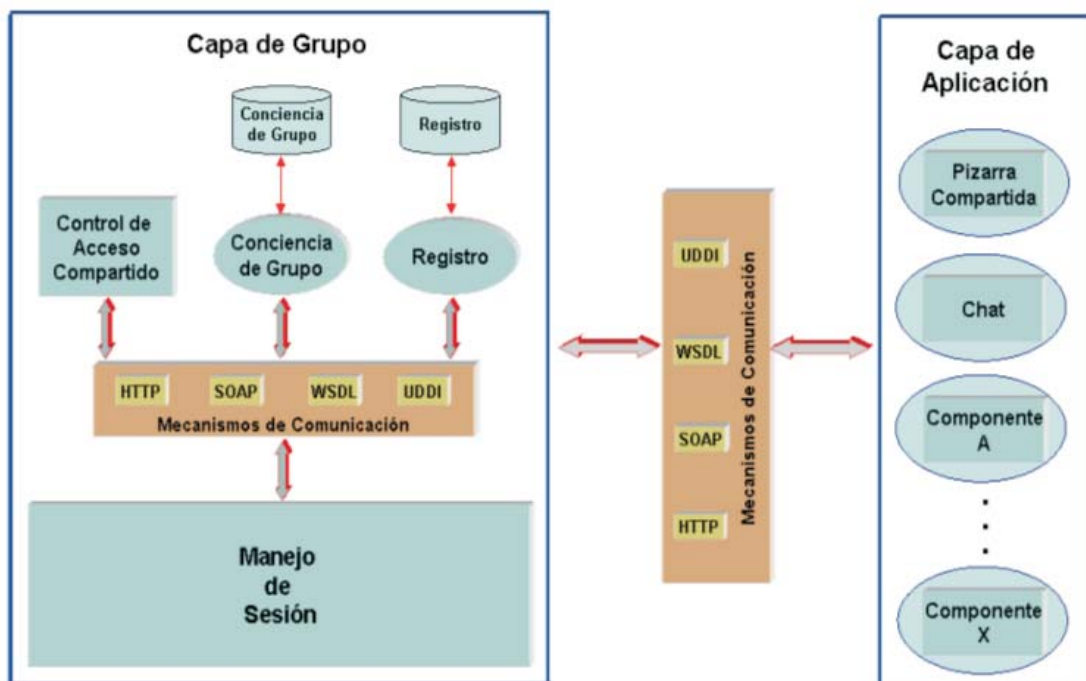
Servicio de Registro: Este servicio permite registrar una nueva sesión y al usuario que la crea y registrar usuarios en una sesión que ya está en funcionamiento. En la tarea de registrar a un usuario en una sesión, un mecanismo de autenticación permite a los usuarios identificarse dando un nombre de usuario y una contraseña, en caso de que dicha sesión lo requiera o de que sea una sesión persistente, de tal manera que los usuarios que colaboraron en ella no tengan que partir de cero cuando deseen continuar, sino desde el punto en que la dejaron. El Servicio de Registro informa de los cambios que se producen (cuando se registra una nueva sesión o un nuevo usuario) a los distintos componentes de la Capa de Grupo, por medio de un mecanismo de notificación. Este mecanismo se basa en el intercambio de documentos XML a través de mensajes, proporcionando relaciones débilmente acopladas entre servicios y conexiones flexibles y adaptables [1], ya que la interfaz del servicio web agrega una capa de abstracción al entorno. De esta manera, cada

componente realiza las siguientes tareas para adaptarse a las nuevas condiciones de trabajo en grupo. La interacción con dichos componentes se da en dos casos:

1. Cuando una nueva sesión y el usuario que la crea se registran. En este caso, sólo se notifica a cada componente que hay una nueva sesión y un participante en ella, pero aún no se puede hablar de sesión colaborativa.
2. Cuando se registra un nuevo usuario en una sesión ya iniciada (que tenga, al menos, un usuario). El Servicio de Registro interactúa con los componentes de la Capa de Grupo (ver Figura 1) de la siguiente manera:

- **Módulo de Manejo de Sesión:** El Servicio de Registro notifica al Módulo de Manejo de Sesión para que le asigne un rol al nuevo usuario, de acuerdo a la política de manejo de sesión que esté establecida. A su vez, este módulo notifica al Servicio de Registro cuando un usuario deja la sesión, para que se actualice el Registro e informe de esta nueva situación al Módulo de Control de Acceso Compartido y al Servicio de Conciencia de Grupo.
- **Módulo de Control de Acceso Compartido:** Este módulo es informado cuando un usuario se registra en una sesión en proceso, para que la política de control de acceso se adecue a la nueva situación.
- **Servicio de Conciencia de Grupo:** El Servicio de Registro le suministra toda la información necesaria (datos del usuario, nombre y tipo de las sesiones abiertas, etc.) para que sea capaz de proporcionar un contexto de trabajo compartido donde el usuario conozca qué usuarios se encuentran en una sesión determinada, qué sesiones actualmente están abiertas, si un usuario ha entrado en la sesión y el tipo de sesión (asíncrona o síncrona, y si es de este último tipo, si es una sesión persistente).

Figura 1. Arquitectura basada en servicios web para desarrollar aplicaciones colaborativas

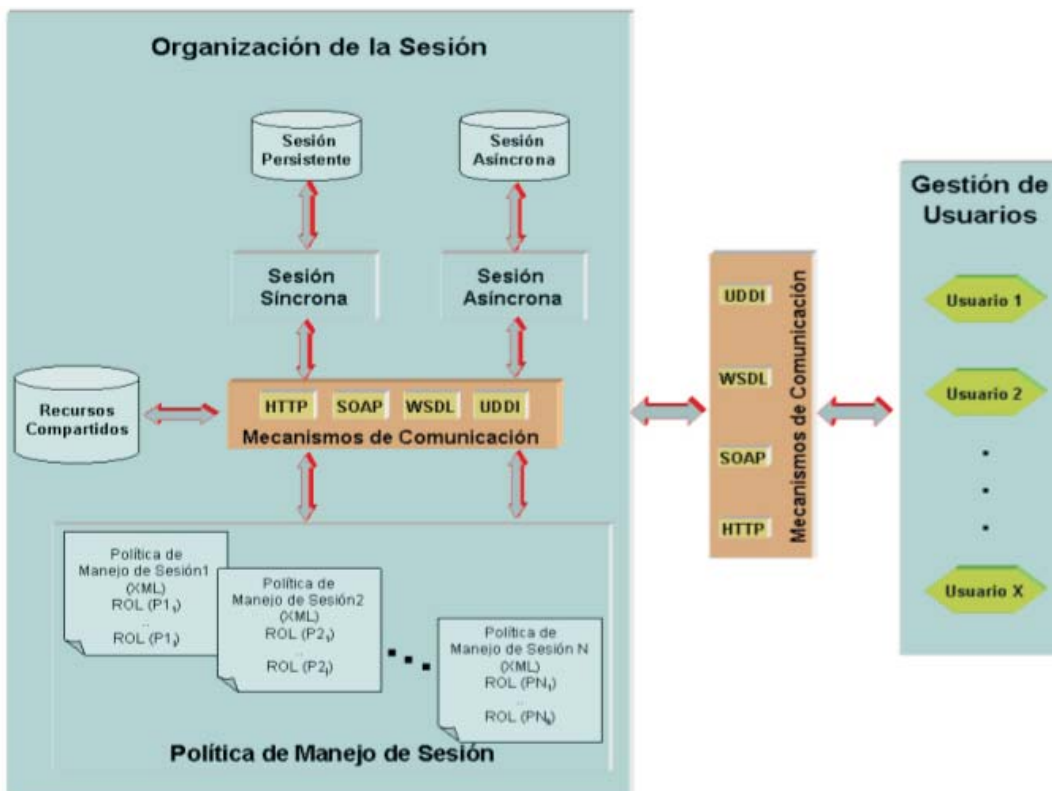


Módulo de Manejo de Sesión: Este módulo suministra los medios necesarios para adaptarse a diversas necesidades y estilos de trabajo de diferentes grupos. Por lo tanto, realiza la gestión de usuarios (registro y pertenencia al grupo) y orquesta la sesión (ver Figura 2). El Módulo de Manejo de Sesión proporciona una interfaz de usuario (mediante una página web) para: registrar una nueva sesión o registrar un usuario en una sesión ya iniciada; gestionar la entrada y salida de usuarios; actualizar la interfaz de los usuarios; definir el tipo de sesión (síncrona o asíncrona); decidir si la sesión síncrona es persistente; determinar la forma en que va a estar organizado el trabajo de grupo a través de la política de manejo de sesión; proveer información sobre el estado de la sesión; invocar a la aplicación a utilizar (por ejemplo, una pizarra compartida) y/o alguna herramienta que se requiera; y facilitar elementos que permitan detener, reanudar y finalizar el trabajo colaborativo. Las tres primeras tareas corresponden a la gestión de usuarios y las tareas restantes pertenecen a la orquestación de la sesión. Como se ha comentado, la sesión colaborativa puede ser asíncrona o síncrona. El primer tipo de sesión, al estar basada en SOA, presenta un consumidor de servicio, que es el usuario que requiere el servicio. El usuario invoca a la aplicación asíncrona (proveedor del servicio) a través de la interfaz que proporciona este módulo. La descripción del servicio se encuentra en un repositorio UDDI [20] y también en un repositorio local, denominado Sesión Asíncrona, que se usa para poder dar una respuesta inmediata en el caso de que existan muchas peticiones. Además, este repositorio local almacena la información generada por los usuarios que será compartida y/o accedida por otros participantes.

La sesión síncrona proporciona un espacio común que permite a los usuarios conectados trabajar de manera conjunta sobre los recursos compartidos, con el fin de realizar una tarea específica en un período de tiempo determinado. Para soportar las tareas llevadas a cabo por la sesión asíncrona y síncrona la arquitectura proporciona:

1. Un repositorio de Recursos Compartidos, donde para cada recurso se registra: su estado (libre u ocupado), el identificador de usuario que lo está utilizando y el rol que está desempeñando, la lista de espera de potenciales usuarios, los datos asociados al recurso, etc.
2. Políticas de manejo de sesión, que se definen en archivos XML y que pueden cambiar en tiempo de ejecución. De momento, se presentan dos tipos de políticas de manejo de sesión: (1) sesión moderada, donde el moderador o presidente coordina y controla la sesión, invitando a los participantes a la misma y proporcionando los turnos en los que cada usuario puede participar; (2) sesión de "lluvia de ideas" (brainstorming), cuyo funcionamiento es similar al de las aplicaciones de mensajería instantánea. Cada una de estas políticas determina los roles que los usuarios pueden desempeñar. Cada rol tiene asociado el conjunto de derechos de acceso sobre los recursos compartidos y las acciones o tareas que el usuario puede realizar. Un usuario puede desempeñar varios roles durante una sesión, siempre y cuando el nuevo rol satisfaga la política de manejo de sesión actual.
3. Interfaz de usuario para invocar a la Capa de Aplicación, que contiene la aplicación o componente a utilizar en la sesión colaborativa, e informa a los usuarios del estado de la sesión, así como cuando cambia algo (un usuario entra

Figura 2. Elementos del módulo de manejo de sesión



o sale, un recurso se utiliza o se deja libre, se han modificado los datos del recurso compartido, etc.), a través del mecanismo de notificación.

El Módulo de Manejo de Sesión es la parte central de la arquitectura, por lo que interactúa con el resto de los componentes de la Capa de Grupo (ver Figura 1):

- Servicio de Registro: El manejador de sesión invoca al Servicio de Registro para registrar una nueva sesión o un nuevo usuario, y también le informa cuando un usuario deja la sesión y cuando una sesión termina, para que este servicio se actualice.
- Módulo de Control de Acceso Compartido: El manejador de sesión informa a dicho módulo cada vez que se produce una de estas dos situaciones: a) Cuando un usuario entra en la sesión, para que le asigne los derechos de acceso a los recursos compartidos (de acuerdo al rol que está desempeñando) y determine si estos derechos le permiten usar el recurso compartido que solicite. b) Cuando un usuario sale de la sesión, para que este módulo se adapte a la nueva situación y para que libere el recurso que este usuario estuviese usando y se lo asigne al siguiente usuario de la lista.
- Servicio de Conciencia de Grupo: El manejador de sesión informa a este servicio de todos los cambios que ocurren, con el fin de que los almacene en el repositorio denominado Conciencia de Grupo y los transmita a la interfaz de cada uno de los usuarios que participan en la sesión o en la tarea colaborativa que se está llevando a cabo (esta última puede involucrar sólo a algunos miembros del grupo).

Finalmente, decir que este módulo soporta sesiones colaborativas flexibles, al permitir: comunicación síncrona y asíncrona, varias prácticas de trabajo en grupo (se puede cambiar la política de manejo de sesión en tiempo de ejecución), cambios de rol (un usuario pueden desempeñar varios roles en una sesión), y diferentes usuarios (que pueden unirse a y dejar la sesión en cualquier momento).

Módulo de Control de Acceso Compartido: Coordina las interacciones para evitar que se presenten conflictos en el espacio de trabajo compartido debido a las actividades colectivas y competitivas que se establecen entre los usuarios, proporcionando permisos que se conceden dinámicamente de manera temporal a los usuarios que colaboran. De esta forma, se atenúan las condiciones de competencia, se garantiza el uso mutuamente exclusivo de los recursos y se proporciona seguridad, imparcialidad, adaptabilidad y estabilidad a los miembros de la sesión. El permiso que se concede al usuario depende del rol que desempeñe en la sesión y especifica que a un usuario concreto se le permite enviar, recibir o manipular datos compartidos en un instante dado. La política utilizada por defecto será "libre para todos", donde los conflictos se resuelven ordenando la secuencia de solicitudes para acceder al recurso compartido, siguiendo el criterio "el primero que lo solicite será el primero que pueda utilizarlo", que garantiza la imparcialidad antes mencionada. La solicitud se hace al repositorio de Recursos Compartidos,

que verifica si el usuario tiene los derechos de acceso necesarios para usar el recurso:

1. En caso de tenerlos:

- Si el recurso está libre, se le permite utilizarlo para realizar la tarea que tiene encomendada, reflejando las modificaciones en la interfaz de los usuarios que estén involucrados en la tarea que se lleva a cabo.
- Si el recurso está siendo utilizado por otro usuario, se informa al usuario que hizo la petición a través del servicio de notificación y dicha solicitud se anexa a la lista de espera asociada al recurso. El usuario puede esperar a que sea su turno para usar el recurso o quitarse de la lista de espera en cualquier momento. Cuando el recurso se libere, se elegirá el primer usuario de dicha lista con derechos de acceso y se le asignará el recurso. Este proceso se repite hasta que todos los usuarios de la lista de espera usen el recurso o la sesión termine.

2. Si el usuario no tiene los derechos necesarios para acceder al recurso, se le notifica que no puede utilizarlo.

Servicio de Conciencia de Grupo: Para ser capaz de cooperar, los usuarios deben ser conscientes de la presencia de los demás miembros en la sesión y de las acciones que cada uno ha realizado y está realizando. Una de las tareas principales de cualquier sistema CSCW es proveer a los usuarios de la información necesaria para proporcionar la conciencia de grupo. Ésta ayuda a los participantes de la sesión a establecer un contexto común, coordinar sus actividades, evitar sorpresas y reducir la probabilidad de conflicto en el grupo. Por ello, este servicio almacena cada acción realizada por los usuarios de la sesión en el repositorio Conciencia de Grupo, de modo que cada acción se notifica al resto de participantes por medio de un mecanismo de notificación. Este servicio informa a los usuarios de:

- Lo que otros usuarios están haciendo en el espacio de trabajo compartido, que se ve reflejado en la interfaz de cada usuario.
- Los cambios en el tamaño del grupo, es decir, indica a los demás usuarios cuándo alguien entra o sale de la sesión.
- Los cambios de estado de un recurso (si está libre u ocupado) y las modificaciones que sufre un recurso.
- La lista de usuarios de la sesión.
- La lista de sesiones que actualmente se están ejecutando.
- El estado del usuario, a través de iconos.

1.2 CAPA DE APLICACIÓN

En esta capa se tiene la aplicación colaborativa específica en la que los usuarios están interesados, es decir, la que realmente desean utilizar para llevar a cabo el trabajo en grupo, como por ejemplo, una pizarra compartida. Al estar diseñada con un estilo arquitectónico orientado a servicios, esta capa permite agregar otras aplicaciones, componentes o herramientas que se encapsulan como servicios web, para proporcionar

funcionalidades acordes a las nuevas necesidades del grupo o del trabajo colaborativo que se está llevando a cabo. Estos servicios deben ser descritos y posteriormente publicados en un repositorio UDDI, así como también en un repositorio local, para evitar cualquier limitación en tiempo de ejecución y dar una respuesta inmediata a una gran cantidad de peticiones simultáneas. La aplicación colaborativa que se ejecuta por defecto es la pizarra compartida. Para agregar otra aplicación, componente o herramienta se siguen dos pasos:

1. Invocar al servicio web:

- Se elige el servicio deseado de la lista de servicios disponibles.
- El servicio invocado debe proporcionar: nombre de la aplicación y recursos que se van a compartir, para que la nueva aplicación se pueda integrar en el entorno colaborativo.

2. Una vez obtenido el servicio:

- Los recursos compartidos se registran en el repositorio denominado Recursos Compartidos, para que puedan ser utilizados por cada usuario, de acuerdo a los permisos que tengan asignados por el rol que desempeñan.
- Se informa al Servicio de Registro para que envíe la lista de usuarios que participan en la sesión creada para dicha aplicación.
- El Manejador de Sesión proporciona los roles de cada usuario con sus respectivos derechos a los recursos compartidos y notifica al Servicio de Conciencia de Grupo de la existencia de la nueva aplicación.
- La aplicación añadida se muestra a cada usuario en una nueva ventana, con el nombre de la aplicación en su barra de título y el nombre de los usuarios que la están utilizando y los recursos compartidos en el área de

trabajo de dicha ventana.

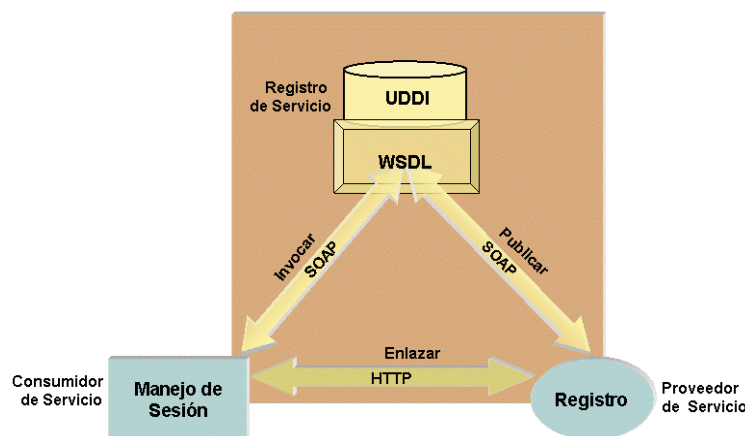
De esta manera, se puede extender simple y rápidamente la funcionalidad de la arquitectura con nuevos servicios web, cuando éstos se requieran, y sin necesidad de modificar los servicios ya existentes en la misma.

1.3 MECANISMOS DE COMUNICACIÓN

Estos mecanismos son esenciales, debido a que a través de ellos los usuarios pueden conectarse a sesiones, establecer diferentes políticas de coordinación y definir diversas políticas de control de acceso a los recursos compartidos, entre otras cosas. Estos mecanismos gestionan tanto la comunicación externa (entre capas) como la comunicación interna (entre los módulos y/o servicios de una capa). Para los mecanismos de comunicación usamos las siguientes tecnologías estándares de los servicios web (ver Figura 3):

- HTTP (HyperText Transfer Protocol): Protocolo usado para establecer la comunicación entre los usuarios de los servicios web.
- SOAP (Simple Object Access Protocol) [3]: Protocolo que proporciona mecanismos para intercambiar mensajes en formato XML y hacer llamadas a procedimientos sobre Internet.
- WSDL (Web Services Description Language) [24]: Formato basado en XML para describir servicios web de manera formal, de modo que se especifica lo que hace un servicio web, dónde está localizado y cómo se invoca.
- UDDI (Universal Description, Discovery and Integration) [20]: Conjunto de esquemas XML estándar, mensajes SOAP y especificaciones API para construir catálogos, es decir, directorios que contienen un registro/repositorio de descripciones de servicios web.

Figura 3. Mecanismos de comunicación



2. ATRIBUTOS DE CALIDAD

El desarrollo de una aplicación colaborativa es mucho más complejo que el de una aplicación monousuario, ya que se tienen que considerar actividades de grupo y protocolos sociales para realizar un buen diseño. Por ello, existe una amplia variedad de metodologías, arquitecturas y entornos que soportan su desarrollo.

Nosotros estamos interesados en la arquitectura software, que se define como “la organización básica de un sistema personificado en sus componentes, las relaciones entre éstos y su entorno, y los principios que guían su diseño y evolución” [2]. Para comenzar el desarrollo de la arquitectura software se puede partir de un documento de especificación de requisitos. Dicho documento debe contener requisitos funcionales (del negocio, del usuario, del sistema, etc.) y requisitos no funcionales (reglas de negocio atributos de calidad del sistema, interfaces externas y políticas, etc.). De todos ellos, estamos más interesados en los atributos de calidad, por lo que seguidamente enumeramos y describimos brevemente los principales atributos de este tipo presentes en nuestra propuesta arquitectónica.

2.1 REUSABILIDAD

La reusabilidad está relacionada con la capacidad de ensamblar o construir sistemas software utilizando componentes predefinidos. Cada elemento de nuestra arquitectura está basado en SOA y es diseñado independientemente, permitiéndonos separar perfectamente las responsabilidades de cada uno, lo que facilita su reutilización. Como SOA es implementado con servicios web, proporciona una interfaz que nos facilita descubrirlo, enlazarlo y ejecutarlo sin importar el lenguaje, plataforma o sistema en el que haya sido implementado. De esta manera, puede ser reutilizado en cualquier entorno colaborativo que lo requiera. Además, los servicios web pueden ser generalmente reutilizados, debido a que son unidades de software modulares. Finalmente, cada aplicación colaborativa puede ser implementada ensamblando los componentes reutilizables de la arquitectura, como por ejemplo: el Servicio de Registro y el Servicio de Conciencia de Grupo.

2.2 PORTABILIDAD

Las aplicaciones portables son aquellas que se transfieren fácilmente de un entorno de software y/o hardware a otro. El hecho de que nuestra propuesta arquitectónica esté basada en la tecnología de servicios web permite que las aplicaciones colaborativas desarrolladas sean independientes del lenguaje, de la plataforma y del sistema operativo, debido principalmente a que:

1. Los servicios web se implementan (comúnmente) bajo una serie de protocolos estándares y tecnologías abiertas.
2. SOA proporciona una interfaz abstracta basada en procesos interconectados que facilitan un conjunto de servicios asíncronos, débilmente acoplados y basados en documentos. Por ejemplo, el Servicio Registro,

implementado mediante clases Java en un sistema operativo Windows y en un ordenador portátil, puede ser invocado y utilizado desde un ordenador convencional que usa un sistema operativo Linux.

2.3 INTEROPERABILIDAD

Este atributo permite la comunicación e interacción entre sistemas. La interoperabilidad se logra debido al hecho de que los servicios web proporcionan una interfaz que permite publicar, descubrir, negociar (para adquirir y combinar), enlazar, ejecutar y desenlazar servicios independientemente del lenguaje, plataforma o sistema que haya sido utilizado para implementarlos. Gracias a que los servicios web usan protocolos web estándares para comunicación, proporcionan independencia de la plataforma y el uso de XML para la representación de datos suministra independencia a nivel del lenguaje de programación. XML también tiene la capacidad de transformar aplicaciones ya existentes en servicios web accesibles, facilitando la interacción entre sistemas.

2.4 UBICUIDAD

La computación ubicua se refiere a una computación que se mueve fuera del “escritorio” y se encuentra inmersa en el ambiente que nos rodea [25]. La ubicuidad es proporcionada por la arquitectura gracias a que los servicios web pueden ser publicados, requeridos, descubiertos, enlazados y ejecutados desde cualquier dispositivo (incluso dispositivos móviles), sin importar el lenguaje, plataforma o sistema que se utilice.

2.5 ADAPTABILIDAD

Atributo que se basa en la necesidad de ajustar o transformar la funcionalidad del sistema de acuerdo con los nuevos requisitos que van apareciendo en el mismo (cambios en el entorno, nuevas necesidades de los usuarios, etc.), de tal manera que éste siga funcionando correctamente. La forma en que se realiza el trabajo en grupo puede variar a lo largo del tiempo, debido a necesidades individuales o colectivas. Esto implica que una aplicación colaborativa debe ser capaz de adaptar su funcionalidad a lo largo de su fase de explotación. Para ello, además de proporcionar elementos que faciliten la adaptabilidad, es necesario establecer una serie de restricciones a comprobar antes de permitir ciertas modificaciones en la funcionalidad de la aplicación colaborativa, con la finalidad de preservar su consistencia y correcto funcionamiento, ya que en ocasiones es posible que no se pueda realizar la adaptación. La arquitectura presentada soporta la adaptación de la funcionalidad de la aplicación colaborativa en dos sentidos principalmente:

1. Adaptación a una nueva forma de organizar el trabajo en grupo. Esto se hace a través de las sesiones colaborativas, que permiten cambiar en tiempo de ejecución:
 - El tamaño del grupo, proporcionando mecanismos que permiten a los usuarios unirse a o dejar una sesión. La forma de unirse a una sesión es por invitación (donde se le informa acerca de los datos de autenticación) o solicitando a un miembro del grupo su autorización

para entrar a la misma. La forma de indicar que un usuario deja una sesión por cualquier motivo es mediante el cierre de la interfaz que tenía asignada. Cuando un usuario se une a la sesión debe ajustarse a las restricciones impuestas por la política de manejo de sesión; por ejemplo, el rol que se le asigne depende de dicha política, así como los derechos de acceso a los recursos compartidos. Lo mismo sucede cuando un usuario deja la sesión; por ejemplo, en caso de que el responsable de moderar una sesión activa salga de ésta, se debe asignar otro moderador para que la sesión siga funcionando.

- Los roles que desempeñan los usuarios, ya que un usuario puede cambiar de rol siempre y cuando cumpla con las restricciones de la política de manejo de sesión. Por ejemplo, en algunos casos dicha política no permite asignar el rol de moderador al nuevo usuario, porque no permite la existencia de dos moderadores.
- Las políticas de manejo de sesión, pero sólo el usuario que modera la sesión o aquél que ha sido elegido por votación puede cambiar dicha política. Este usuario asigna los roles y derechos de acceso de acuerdo a la nueva política de manejo de sesión elegida.

2. Adaptación a los nuevos requisitos del trabajo o a las nuevas necesidades del grupo. Consiste en adaptar la aplicación colaborativa de tal manera que permita agregar nuevas funcionalidades para realizar nuevas tareas o terminar una tarea más rápidamente. Para ello, la arquitectura permite agregar (como se indica en la sección 1.2) una nueva aplicación, componente o herramienta colaborativa (que se encuentra en la Capa de Aplicación, envuelta como servicio web) que satisfaga las nuevas necesidades y requisitos del grupo.

3. TRABAJOS RELACIONADOS

En los últimos años ha surgido una gran variedad de modelos arquitectónicos y entornos para desarrollar aplicaciones colaborativas, y también aplicaciones colaborativas basadas en servicios web.

Los modelos arquitectónicos, tales como Interactor Models [16], PAC [6], y MVC [14], modelan el sistema como un conjunto de componentes y de relaciones entre éstos. También existen extensiones propuestas para sistemas CSCW, donde podemos incluir a PAC* [5] y a la arquitectura genérica de Dewan [7], y nuevos modelos, como la taxonomía de Patterson [21], COCA (Collaborative Objects Coordination Architecture) [18], Clock [13] y Clover [17]. Un inconveniente de estos modelos es que están centrados en los aspectos interactivos de los sistemas, a excepción de COCA, que se orienta a modelar políticas de coordinación independientes de las otras dos características (comunicación y colaboración), pero que no facilita la interoperabilidad.

También existen diversidad de entornos que facilitan el desarrollo de aplicaciones colaborativas; sin embargo, todos presentan inconvenientes, como por ejemplo: Groupkit [22], que proporciona una librería de componentes para construir interfaces multiusuario, aunque no son fáciles de personalizar

ni pueden interoperar entre ellas. JSDT [4], que proporciona un conjunto de APIs para construir aplicaciones colaborativas con mecanismo de conciencia de grupo, pero en el que dichas aplicaciones son fuertemente acopladas al entorno y dependientes de éste. COAST [23] permite desarrollar aplicaciones colaborativas sincronas basadas en documentos a través de una arquitectura general; sin embargo, no ofrece mecanismos para agregar componentes de terceros. TOP [15] permite desarrollar aplicaciones a través de la definición de diez objetos, pero no facilita cambiar la forma en la que se organiza en grupo en tiempo de ejecución ni tampoco permite la incorporación de componentes de otras aplicaciones. ANTS [12] facilita el desarrollo de aplicaciones CSCW genéricas, proporcionando servicios de supervisión y conciencia de grupo y un modelo arquitectónico de tres capas; su principal inconveniente es que no permite extender la aplicación con otras aplicaciones colaborativas. Finalmente, CoopTEL [11] define aplicaciones colaborativas basadas en modelos de componentes y aspectos, sin embargo, no permite añadir otras aplicaciones groupware.

Además, hemos estudiado y analizado las aplicaciones colaborativas basadas en servicios web más representativas y hemos observado que no existe una arquitectura flexible y extensible para desarrollar sistemas CSCW basados en este tipo de servicios. Generalmente el empleo de servicios web brinda la posibilidad de establecer comunicación asíncrona y sincrónica; sin embargo, sería conveniente utilizar las tecnologías estándares de los servicios web como mecanismos de comunicación para establecer mecanismos de control de acceso a recursos compartidos, de notificación y de conciencia de grupo. Para la coordinación sería deseable desarrollar un manejador de sesión que permita crear, unirse a o dejar una sesión en tiempo de ejecución, además de proporcionar un repositorio que sirva de base a los mecanismos de conciencia de grupo y a los mecanismos de notificación, de tal forma que aseguren la consistencia de la información compartida.

4. CONCLUSIONES Y TRABAJO FUTURO

Se ha presentado una propuesta arquitectónica para desarrollar aplicaciones colaborativas de calidad basada en servicios web. Dicha propuesta permite reducir el tiempo, esfuerzo y coste en el desarrollo de este tipo de aplicaciones, haciendo además que éstas sean flexibles y extensibles, y dotándoles de mecanismos de conciencia de grupo. El proceso de desarrollo se reduce debido a que la aplicación se construye rápidamente ensamblando cada elemento de la arquitectura por medio de una interfaz de usuario (página web). Las aplicaciones colaborativas que se desarrollan con nuestra arquitectura son: flexibles (proporcionan diferentes estilos para organizar el trabajo en grupo en tiempo de ejecución) y extensibles (permiten agregar nuevos servicios web, si se requieren, sin necesidad de modificar los servicios existentes). Además, la arquitectura proporciona a las aplicaciones basadas en ella otros atributos más generales (atributos de calidad), como son: reusabilidad (de componentes), portabilidad (independencia respecto a la plataforma, lenguaje de implementación o entorno), interoperabilidad (comunicación y colaboración entre

sistemas), ubicuidad (ejecutables desde cualquier dispositivo, incluso uno móvil) y adaptabilidad (a los nuevos requisitos y necesidades del grupo).

El trabajo futuro se concentrará en:

- 1) Modelar las políticas de manejo de sesión mediante ontologías, para proveer sesiones más flexibles y adecuadas a la diversidad de formas de organizar el trabajo en grupo.
- 2) Definir un mecanismo para desarrollar aplicaciones colaborativas adaptativas, o sea, que automáticamente detecten las nuevas necesidades y se adapten a ellas.
- 3) Incrementar el número de políticas de control de acceso a recursos compartidos.
- 4) Reforzar la conciencia de grupo, estableciendo nuevos mecanismos de notificación.
- 5) Redefinir mecanismos de autenticación a través de SAML (Security Assertion Markup Language) [19].

5. AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto español CICYT TIN2004-08000-C03-02.

6. REFERENCIAS

- [1] ALONSO, G., CASATI, F., KUNO, H., and MACHIRAJU, V. *Web Services: Concepts, architectures and applications*. SpringerVerlag, 2004. 354 p.
- [2] Architecture working group: Recommended practice for architectural description of software-intensive systems. IEEE Std 1471 2000.
- [3] BOX, D., EHNEBUSKE, D., KAKIVAYA, G., LAYMAN, A., MENDELSON, N., NIELSEN, H. F., THATTE, S., and WINER, D. *W3C's Simple Object Access Protocol (SOAP)*. 2003. <http://www.w3.org/TR/SOAP/>.
- [4] BURRIDGE, R. *Java Shared Data Toolkit user guide version 2.0*. Sun Microsystems, JavaSoft Division. 1999.
- [5] CALVARY, G., COUTAZ, J., and NIGAY, L. From single-user architectural design to PAC*: A generic software architecture model for CSCW. In Proc. of the SIGCHI. 1997. 242-249 p.
- [6] COUTAZ, J. PAC-ing the architecture of your user interface. In Proc. of the Fourth Eurographics Workshop on DSVIS. Springer-Verlag. 1997. 15-32 p.
- [7] DEWAN, P. Multiuser architectures. In Proc. of the IFIP TC2/WG2.7 Working Conference on Engineering for Human-Computer Interaction. 1995. 15-32 p.
- [8] ELLIS, C.A., GIBAS, S.J., and REIN, G.L. Groupware: some issues and experiences. *Communications of the ACM*, Vol. 34-1. 1991. 39-58 p.
- [9] ERL, T. *Service-Oriented Architecture: A field guide to integrating xml and web services*. Prentice Hall. 2004.
- [10] ERL, T. *Service-Oriented Architecture: concepts, technology, and design*. Prentice Hall. 2005).
- [11] FUENTES, L., PINTO, M., AMOR, M., and JIMÉNEZ, D. CoopTEL: a component-aspect middleware platform. In Proc. of ACM/IFIP/USENIX Int. Middleware Conference. 2003.
- [12] GARCÍA, P., and GÓMEZ, A. ANTS Framework for Cooperative Work Environments. *IEEE Computer Society Press*, Vol. 36-3. 2003. 56-62 p.
- [13] GRAHAM, T.C.N., and URNES, T. Integrating support for

temporal media in to an architecture for graphical user interfaces. In Proc. of the International Conference on Software Engineering (ICSE). ACM Press. 1997. 172-182 p.

- [14] GOLDBERG A. *Smalltalk-80: The interactive programming environment*. Addison Wesley. 1984.
- [15] GUERRERO, L.A., and FULLER, D. CLASS: A computer platform for the development of education's collaborative applications. Proc. of CRIWG. 1997. 51-60 p.
- [16] HARRISON, M., and THIMBLEBY, H. (eds.): *Formal Methods in HCI*, Cambridge University Press. 1990.
- [17] LAURILLAU, Y., and NIGAY, L. Clover architecture for groupware. In Proc. of the ACM Conference on CSCW. 2002. 236-245 p.
- [18] LI, D., and MUNTZ, R. COCA: Collaborative Objects Coordination Architecture. In Proc. of CSCW. ACM Press. 1998. 179-188 p.
- [19] OASIS (Organization for the Advancement of Structured Information Standards): *Security Services (SAML) TC*. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss (2004).
- [20] OASIS (Organization for the Advancement of Structured Information Standards): *Universal Description, Discovery, and Integration (UDDI)*. <http://www.uddi.org/> (2003).
- [21] PATTERSON, J.F. Taxonomy of architectures for synchronous groupware applications. In Proc. of the CSCW'94 Workshop on Software Architectures for Cooperative Systems, Vol. 15-3. ACM SIGOIS. 1994. 27-29 p.
- [22] ROSEMAN, M., and GREENBERG, S. Building realtime groupware with GroupKit: A groupware toolkit. *ACM Trans. Computer-Human-Interaction*, Vol. 3. 1996. 66-106 p.
- [23] SCHUCKMANN, C., KIRCHNER, L., SCHÜMMER, J., and HAAKE, J.M. Designing object-oriented synchronous groupware with COAST. In Proc. of CSCW. 1996. 30-38 p.
- [24] W3C Consortium: *Web Services Description Language (WSDL)*. <http://www.w3.org/TR/wsdl> (2001).
- [25] WEISER M. The computer for the 21st century. *Scientific American*, Vol. 265-3 1991. 94-104 p.

