

Videoconferencia con Isabel en la Web 2.0

Fernando Escribano, Javier Cerviño, Pedro Rodríguez, Joaquín Salvachúa

Departamento de Ingeniería de Sistemas Telemáticos,

Universidad Politécnica de Madrid

Avda. Complutense, Ciudad Universitaria s/n, 28040 - Madrid, España.

{fec, jcervino, prodriguez, jsalvachua}@dit.upm.es

Resumen- Este artículo presenta una arquitectura que transforma una aplicación tradicional de videoconferencia en un servicio Web accesible desde cualquier navegador, de tal forma que cambia el modelo de uso y propone un nuevo sistema de colaboración en tiempo real que cuenta con las ventajas intrínsecas de cualquier servicio de la web 2.0. Además esta solución facilita la transición hacia un servicio típico de Cloud Computing en el que se utilizan y se liberan recursos de videoconferencia dependiendo de la demanda en cada momento.

Palabras Clave- Videoconferencia, Colaboración, Web, Cloud Computing.

I. INTRODUCCIÓN

En los últimos años se ha cambiado la filosofía de las aplicaciones que los usuarios utilizan. De tener aplicaciones instaladas en su ordenador se utilizan aplicaciones basadas en el navegador. Este nuevo escenario tiene algunas ventajas, como eliminar problemas de instalación, versionado y configuración. Por otra parte abstrae a los desarrolladores de características del sistema operativo y plataforma sobre la que se ejecuta.

Este tipo de aplicaciones web altamente interactivas se han venido a llamar RIA (*Rich Internet Applications* [1]). Este paradigma está creciendo en uso y aceptación por parte de los usuarios ya que acceden a estas aplicaciones desde cualquier dispositivo con la necesaria conexión a la red. Uno de sus principales problemas es qué hacer cuando no hay conexión. Se está trabajando en esta línea, pero estamos habituados a la conexión permanente.

En nuestro caso nos plantamos como se puede realizar un sistema de comunicación multimedia altamente interactivo dentro de una plataforma RIA. Anteriores experiencias, como Marte [2], nos había permitido conocer que algunas tecnologías tales como Adobe Flash [3] permite un acceso a los diversos recursos multimedia del ordenador desde el navegador.

En este artículo presentamos la arquitectura híbrida que permite la integración entre múltiples tecnologías de una forma flexible que da una visión perfectamente integrada. Es necesario destacar que la mayor complejidad está en la interoperabilidad a nivel de control, ya que la interoperabilidad de los códecs de audio y video es un problema técnico resuelto con relativamente poca complejidad conceptual.

La mayor dificultad está en ver qué modelo de uso existe detrás de dichas aplicaciones multimedia distribuidas; o como en el caso de Isabel dicho modelo es reprogramable al vuelo. Las arquitecturas basadas en diversas pasarelas no sólo de multimedia sino incluso de paradigma de uso aislando y adaptando cada metáfora y caso de uso en las diversas plataformas integradas. No tiene ningún sentido el

forzar que dichos casos de uso se implementen donde no son adecuados, como puede ser los teléfonos SIP.

Por otra parte la arquitectura aquí presentada nos abre la puerta a sistemas de tipo *Cloud Computing* [4] donde la videoconferencia sea un servicio disponible que los usuarios son capaces de usar de manera totalmente transparente permitiendo la *Collaboration as a Service*.

Creemos que la solución arquitectónica, usando el patrón del mediador y de las *facade*, es lo suficientemente general como para ser aplicada a cualquier arquitectura multimedia actual. Creemos que es una contribución importante, no existente anteriormente, para la evolución de la actual Web 2.0 a un internet de tiempo real y de las cosas.

A lo largo del artículo vamos a ver por este orden la aplicación Isabel [5], que es sobre la que creamos una pasarela que permite la evolución hacia un servicio Web. A partir de ahí empezaremos a describir las piezas básicas de la pasarela, desde las librerías en las que se basa (GAPI) hasta la arquitectura de los diferentes componentes y sus funciones. En el siguiente apartado explicaremos las pruebas y los principales problemas a los que nos enfrentamos durante el desarrollo y terminaremos comentando diversas conclusiones del trabajo.

II. LA APLICACIÓN ISABEL

Isabel es una herramienta de colaboración multimedia y multipunto que, mediante un innovador concepto de servicio denominado escenario flexible, permite que la colaboración remota sea lo más parecida posible a la presencial. Esto se consigue basándose en dos ideas básicas:

- Protocolos sociales: La gestión del escenario de colaboración se realiza de acuerdo a las convenciones que rigen una determinada actividad presencial. Por tanto si en una clase es el profesor el que controla las actividades en el aula, también será el profesor el que determine la interacción durante una clase distribuida con Isabel.

- Producción multimedia distribuida: La vista de todos los participantes es la misma para todos e integra tanto los videos activos en ese momento como las herramientas de colaboración correspondientes a la actividad que se desarrolle. Por ejemplo durante una clase se podría mostrar el video del profesor en una posición principal, los videos de los alumnos en segundo plano, una presentación de transparencias y un panel que indica si algún alumno ha “levantado la mano” para hacer una pregunta.

En el escenario flexible de Isabel se han introducido tres servicios diferentes que permiten realizar tres clases de reuniones tal y como se detalla a continuación:

- **Congresos distribuidos:** realizados con varios auditorios interconectados con Isabel, donde los ponentes, presidentes de mesa o asistentes pueden participar desde cualquiera de ellos. En este servicio el control está centralizado y los modos de interacción son seleccionados desde una sala de control de acuerdo al programa de la conferencia.

- **Clases distribuidas:** realizadas con profesores y alumnos que participan desde diferentes sedes. Este servicio está diseñado para mantener un control estricto por parte del profesor, permitiéndole hacer presentaciones, charlas, etc... El educador también puede dar la palabra a cualquier sitio en cualquier momento.

- **Reuniones de trabajo distribuidas:** para la coordinación de proyectos u otras actividades interconectando personas o salas pequeñas mediante Isabel. En este servicio no hay control centralizado, sino que todos los participantes pueden tomar el control.

III. ISABEL GAPI

La pasarela genérica de Isabel, también conocida como GAPI (Gateway API), es un módulo de pasarela genérico desarrollado con el fin dar servicio a pasarelas hacia protocolos específicos (SIP, H323,...), de manera que la parte común a todas las pasarelas se encuentre en un único paquete y pueda ser reutilizado de manera simple y rápida. GAPI ha sido desarrollada como paquete Java [6] y ofrece las funcionalidades más comunes que puede necesitar una pasarela mediante un API simple y bien definido. La Fig. 1 muestra esta arquitectura.

La función principal del GAPI es el manejo del protocolo de señalización de Isabel que se ofrece al programador de pasarelas a través de una serie de interfaces sencillas en Java. De esta forma se permite la conexión y desconexión de usuarios a la sesión, el control de los modos de interacción y la solicitud de turno de preguntas en los modos en los que está disponible. Así mismo el GAPI informa mediante callbacks a los GAPIListeners de conexiones y desconexiones de usuarios, cambios de modo de interacción y de los flujos multimedia activos.

Por otro lado, para la gestión de los flujos multimedia, el GAPI controla una MCU que distribuye los diferentes flujos RTP de los clientes de la pasarela hacia el componente de multicast a nivel de aplicación de Isabel y de éste hacia los clientes de la pasarela.

IV. PASARELA WEB

A. Arquitectura General

El objetivo de este apartado es dar una visión general de cada una de las partes que servirá como introducción a la siguiente parte del artículo donde se entrará en detalle sobre cada una de los componentes.

En primer lugar tenemos el mundo Isabel. Como se ha explicado anteriormente, Isabel es una aplicación de videoconferencia que utiliza su propio protocolo de control sobre UDP mientras que los diferentes flujos se transmiten utilizando el protocolo RTP [7]. Por otro lado, tenemos el mundo Flash en el que la transmisión tanto de mensajes de control como de flujos multimedia se realiza mediante el protocolo RTMP [8].

Es, por tanto, labor de la pasarela traducir los tanto el control de la videoconferencia como los datos de audio y vídeo en ambos sentidos para que la comunicación entre los dos sistemas sea posible.

Además, Isabel tiene su propio mecanismo de redistribución de flujos entre pares, en el mundo Flash, para una comunicación multipunto es necesario disponer de un servidor central que se encargará de distribuir los diferentes mensajes y flujos entre los clientes. Para este desarrollo se ha decidido utilizar Red5 [9].

Finalmente, se considera parte del desarrollo de la pasarela el diseño e implementación de la aplicación Flash que correrá en los navegadores de los clientes. Como veremos, esta aplicación tiene que cumplir ciertos requisitos impuestos por Isabel, así como tratar correctamente los flujos según su proveniencia (pasarela o resto de los clientes).

Todo esto queda de manifiesto en la Fig. 2, donde vemos cada uno de los grandes componentes de la pasarela y su comunicación con Isabel a nivel de protocolos.

B. Arquitectura de la pasarela Isabel

En Fig. 3 se muestra la arquitectura de la pasarela en mayor detalle. En ella se muestran elementos de la capa de control y de la capa de distribución de los flujos multimedia.

Gracias al GAPI, la pasarela Flash minimiza la complejidad del código destinado al control proveniente y hacia la sesión de Isabel, limitándose a traducir determinadas órdenes que son necesarias de cara a los clientes.

A continuación se describen superficialmente cada uno de los componentes que aparecen en la figura:

- **Flash Gateway:** Es la capa superior, como se puede apreciar en la figura. Es responsable de la comunicación entre el GAPI y el Flash Manager. Escucha los eventos generados en ambas partes propagándolos de manera que el estado quede sincronizado.
- **GAPI:** Es el componente explicado anteriormente que facilita en gran medida la conversión del protocolo de control de Isabel.

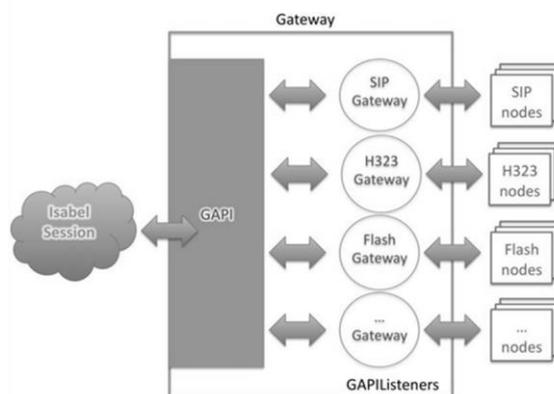


Fig. 1 Arquitecturas de utilización del GAPI

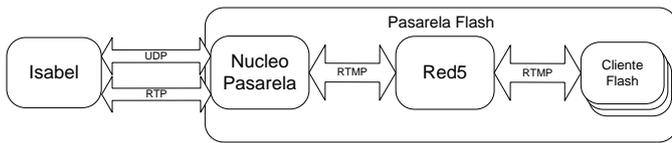


Fig. 2 Componentes que forma la pasarela.

- Flash Manager: Esta pieza junto con su principal módulo llamado Red5UserManager, se encarga de toda la comunicación con el servidor Flash.
 - MCU: La MCU de Isabel es un desarrollo de software capaz de transcodificar vídeo y audio así como de sumar flujos de audio y formar mosaicos de vídeo. La pasarela utiliza principalmente la MCU para manejar audios y vídeos según sea conveniente de cara al envío de estos a la sesión de Isabel o al servidor Flash.
 - Servidor VNC: En la pasarela, el servidor VNC es necesario para obtener de una manera sencilla la composición del escritorio de la sesión de Isabel para poder mandárselo a los clientes Flash sin que éstos tengan que componer su propia imagen a partir de vídeos separados. Además se asegura que lo que ve cada cliente es lo mismo, siguiendo el modelo de Isabel.
 - Transcoder: Este módulo se encarga de realizar las transcodificaciones tanto de audio como de vídeo necesarias para que los dos sistemas se entiendan.
 - Conexión RTMP: Este componente emula a un cliente RTMP (que normalmente es un Flash Player). Este cliente simulado es el que, de cara al servidor Flash, publica y recibe los vídeos y audios correspondientes a la sesión.
- Hasta aquí el repaso de los diferentes componentes. En las siguientes secciones del artículo se detalla el funcionamiento de los mismos.

C. Arquitectura del Red5 y Clientes Web

Tal y como se puede ver en la Fig. 4 la pasarela Flash envía el video de Isabel al servidor Red5 utilizando el protocolo RTMP de Adobe. Para ello utiliza un cliente RTMP que se conecta directamente a este servidor. Por otro lado aquellos usuarios que quieran acceder al video de Isabel tendrán que conectarse también a este servidor, de forma que el servidor además les permite interactuar en la sesión Isabel de forma activa. En el momento en el que un usuario con un cliente Flash se conecta al servidor Red5, podrá enviar el video captura de su webcam, el audio de su micrófono, podrá realizar preguntas durante un evento de Isabel e incluso controlar un escritorio remoto. Todo esto se consigue gracias a que el servidor Red5 reencaminará estos flujos multimedia y los eventos generados en los clientes Flash hacia la pasarela Flash, y viceversa.

En la misma figura podemos ver como se reencaminan los flujos entre las máquinas Isabel y los clientes Flash. La línea continua representa los flujos de video de los clientes Flash. La línea con puntos representa el audio de éstos y la línea discontinua representa el video y el audio mezclados en un único flujo que Isabel a través de la pasarela Flash envía al Red5. Este flujo es la suma de todos los audios de los terminales Isabel, no de los clientes Flash, ya que así se evitan ecos y realimentación del sonido. La suma de videos de todos los participantes (incluyendo Isabel y clientes Flash) se hace con la composición que realiza Isabel mismo con el

modo de interacción elegida incluyendo, si se escoge, cualquier video de los clientes Flash que están conectados a la sesión a través de dicha pasarela.

El flujo de audio de cada cliente Flash se envía al servidor Red5, que lo transmite a Isabel a través de la pasarela. Es Isabel mismo el que toma la decisión de presentar o no cualquier flujo de video por pantalla dependiendo del modo de interacción elegido por el operador de Isabel. Dependiendo del modo de interacción el operador puede controlar la sesión incluso utilizando un cliente Flash.

El servidor Red5 puede servir diferentes sesiones de Isabel simultáneamente ya que se pueden arrancar múltiples instancias de la misma aplicación que se ejecuta sobre éste, cada una para una sesión de Isabel. Por lo que los diferentes clientes Flash se podrían conectar a diferentes sesiones de Isabel al mismo tiempo y utilizando el mismo Red5. Se pueden distinguir las sesiones de Isabel a través de la URL por la que se conectan estos clientes:
`rtmp://servername/IsabelWebGWApp/isabelSessionID.`

D. Librerías Utilizadas

Para transcodificar el video y el audio que se obtiene de los terminales Isabel para después incluirlo en una conexión RTMP se utiliza la librería de código abierto llamada Xuggler [10]. Esta librería está pensada para poder codificar, decodificar y manipular video y audio en tiempo real desde aplicaciones Java. Para ello utiliza la librería FFMpeg [11] por debajo para conseguir codificar y decodificar el video y el audio. En nuestro caso lo que hacemos por una parte es obtener el audio de Isabel a partir de una suma de todos los audios que obtenemos de la sesión de Isabel, es decir, del audio de todos los terminales que están emitiendo. En cuanto al video lo que hacemos es representar el video en una pantalla que después capturamos a través de un cliente VNC especial, de esta forma obtenemos las imágenes que se mostrarían por éste VNC y las codificamos utilizando el Xuggler.

Para realizar este cliente VNC especial nos basamos en la librería Java creada por TightVNC [12] con la que se implementa un visor VNC completamente en Java. Esta librería es también de código abierto y se distribuye con el servidor VNC del mismo nombre. En nuestro caso modificamos el código del cliente Java para, simplemente, quedarnos con las imágenes que este cliente recompone para formar la vista del escritorio.

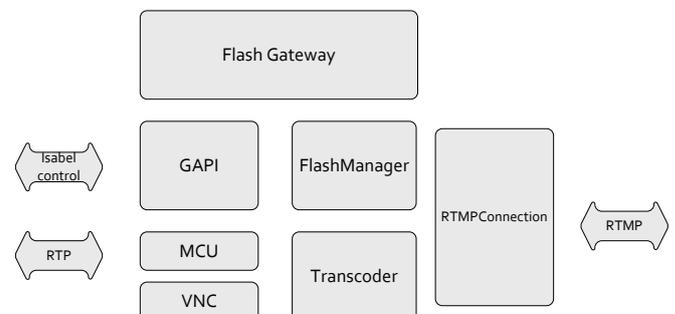


Fig. 3 Arquitectura detallada de la pasarela

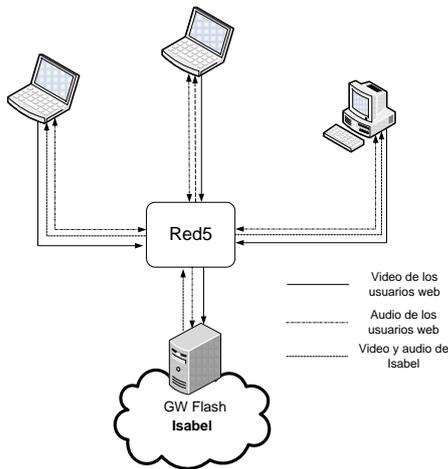


Fig. 4 Arquitectura de la aplicación Red5

Los flujos de video y audio generados por el Xuggler se empaquetan dentro de conexiones RTMP utilizando la librería que se incluye en Red5 para implementar clientes RTMP en Java. De nuevo utilizamos esta librería por ser de código abierto, y porque en este caso era la única librería que conocíamos, ya que la parte del servidor también está implementada sobre los mismos APIs, ya que se ejecuta con el servidor que lleva su nombre: Red5.

Los clientes Flash se implementaron utilizando el framework de código abierto Flex [13], creado por Adobe, y que permite crear aplicaciones que se ejecutan sobre el reproductor Flash Player, que hoy en día se puede instalar sobre muchos navegadores web. Estos clientes son los que se conectan al Red5 para interactuar en la sesión de Isabel. Como ya hemos dicho anteriormente estos clientes enviarán audio y video hacia Isabel, por lo que de nuevo se hará uso de la librería Xuggler para que desde la pasarela se obtengan estos flujos multimedia y poderlos transcódicar a códecs y formatos que conoce Isabel.

E. Configuraciones de la pasarela

Debido a las posibilidades que ofrece tanto las librerías de Xuggler como de Red5 la pasarela se ha implementado de forma que a través de ciertas variables de configuración permita ofrecer unos servicios bien diferenciados.

De esta forma es posible configurar la URL del servidor a donde se va a realizar la conexión RTMP y el nombre principal del flujo de video y audio. También es posible separar los flujos de video y audio para tener una pasarela que únicamente envíe audio, al igual que cambiar el ancho de banda utilizado en ambos sentidos: uno en cuanto al video y al audio que se recoge de Isabel, y otro en cuanto al audio y al video que se recoge de los clientes Flash.

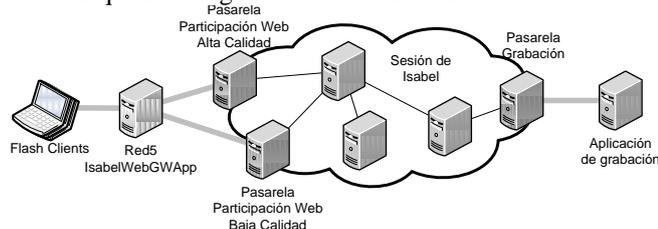


Fig. 5 Escenario real de una sesión Isabel con pasarelas

Además permite la utilización del códec H.264 en vez del códec por defecto, que es Sorenson Spark. Y otros parámetros adicionales como la configuración del servidor VNC al que controlaran los clientes Flash cuando activen el modo de compartición de escritorio de Isabel.

Todo esto permite que la pasarela tenga dos funcionalidades clave:

- La primera de ellas es la función de participación Web, que es la que hemos comentado a lo largo del artículo, que a su vez se puede dividir en diferentes pasarelas con diferentes anchos de banda y calidades de video para dar servicio a una mayor variedad de usuarios (teniendo en cuenta simplemente el ancho de banda del que pueda disponer cada uno). Como resultado tendríamos que en una sesión de Isabel podrían estar conectadas dos o más pasarelas apuntando al mismo Red5 (pero a diferentes instancias del servidor), por lo que los clientes Flash a través del mismo cliente podrían conectarse a la sesión de Isabel a mayor o menor calidad dependiendo del ancho de banda disponible en su conexión hacia el Red5.

- La segunda es la aplicación de la pasarela como centro de streaming y grabación de la sesión de Isabel en formato FLV [14], para poder ser reproducido tanto en tiempo real como en diferido por parte de reproductores Flash, los mismos que permiten hoy en día ver videos en YouTube, por ejemplo. A su vez podríamos tener streaming y grabación a distintas calidades.

El resultado es que en una misma sesión de Isabel podremos ver conectadas (como en Fig. 5) varias pasarelas equivalentes que tienen objetivos diferentes: unas podrían dedicarse a dar servicio a usuarios que quieren participar en una sesión (permitiendo incluso varias calidades de video) y otras a ofrecer el video para que otros usuarios puedan visualizarlo (también con varias calidades).

F. Audio y Video

Una de las principales labores de la pasarela es traducir los flujos de audio y vídeo entre el mundo Isabel y el mundo Flash. En este proceso intervienen cuatro componentes: el servidor VNC, MCU, RTMPConnection y Transcoder.

Como se ha adelantado anteriormente, el servidor VNC va a proporcionar el vídeo que se destina a los clientes Flash. Para ello, se ejecuta una sesión Isabel en su interior para obtener la composición de la pantalla propia de la aplicación. A partir de ahí, el servidor VNC pasa las imágenes al Transcoder que se encarga de codificarlas al formato necesario pasándolo finalmente al RTMPConnection que encapsulará el flujo en paquetes RTMP enviándolos finalmente al servidor Flash. En cuanto a los formatos de vídeo, las mayores restricciones las encontramos, como vemos en la tabla, en la plataforma Flash que, además de hacer necesario el uso de RTMP, impone restricciones en cuanto a los códecs que permite utilizar.

Formatos	Reproducción	Captura
Sorenson Spark	SI	SI
On2 VP6	SI	NO
H.264	SI	NO

Tabla 1 Códecs aceptados por el reproductor Flash.

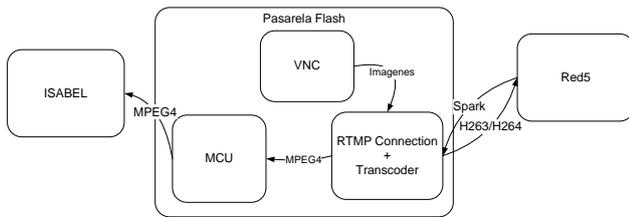


Fig. 6 Transcodificación del video

En el otro lado, Isabel permite utilizar (tanto en reproducción como capturando de una cámara) una gran variedad de códecs, siendo MPEG4 el más habitual. Así, normalmente es necesaria una transcodificación de los vídeos provenientes de los clientes Flash antes de que estos puedan ser vistos por Isabel.

En ese caso, los datos recibidos mediante RTMPCOnnection en Sorenson Spark se pasan al transcoder que se encarga de pasarle el vídeo en formato MPEG4 y en paquetes RTP. Finalmente, la MCU retransmite estos paquetes hacia Isabel. El proceso en ambos sentidos puede verse en la Fig. 6.

El caso del audio es más sencillo ya que en ningún caso hace falta transcodificación debido a que tanto Isabel como los clientes Flash son capaces de operar con Speex. En este caso, la única labor del Transcoder es introducir los datos en paquetes RTP cuando la comunicación es en el sentido hacia el mundo Isabel. Además, para ahorrar ancho de banda se utilizan las capacidades de la MCU como sumador de audios uniendo todos los flujos provenientes de Isabel en uno único que será el que se transmita a todos los clientes Flash.

G. Control de los Modos de Interacción

Para controlar los modos de interacción de Isabel desde los clientes de la pasarela web así como la solicitud del turno de preguntas, se ha implementado un mecanismo que permite presentar al usuario distintos botones dependiendo de la actividad de colaboración que se está desarrollando. Por ejemplo durante una conferencia distribuida en la que el control está fuertemente centralizado a los usuarios de la pasarela solo se les permite pedir turno de pregunta, mientras que si la actividad es una reunión de trabajo distribuida en la que cualquiera puede tomar el control de la reunión, el usuario de la pasarela tiene una serie de botones que le permiten cambiar el modo de interacción entre los disponibles.

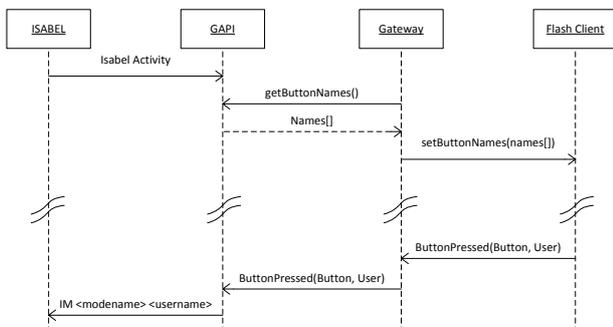


Fig. 7 Mecanismo de control de los modos de interacción.

El mecanismo implementado tiene los siguientes pasos tal y como se muestran en Fig. 7:

1. Al conectar un cliente de la pasarela se le da una lista con los nombres de los botones que tiene que presentar al usuario. Desde el punto de vista del software de la pasarela todos estos botones son idénticos en funcionamiento.

2. Cuando el usuario pulsa uno de estos botones, la misma función es llamada en el servidor Red5, que en respuesta avisa a la pasarela Web de que un botón ha sido pulsado por un usuario.

3. La pasarela, mediante una llamada al GAPI, envía el mensaje pertinente a Isabel mediante su protocolo de señalización interna.

H. Control de Escritorio Remoto

Para permitir el control del escritorio remoto por parte de un usuario de la pasarela web que lo que está viendo es un flujo de vídeo, hemos implementado un mecanismo que recoge eventos de ratón y teclado en los clientes flash sobre el video y los envía hacia la pasarela. Puesto que cada cliente puede estar viendo el vídeo a una resolución diferente es necesario escalar las coordenadas de la posición del ratón a una resolución estandarizada. En la pasarela estos eventos se reproducen programáticamente sobre el cliente VNC que corre en la pasarela. Esto nos permite controlar todos los elementos interactivos que aparecen en una sesión de colaboración Isabel: escritorio remoto, panel de preguntas, pizarra, etc...

Este mecanismo solo se activa cuando el modo de Isabel lo permite y para los usuarios que indica el modo de interacción. Para ello el GAPI avisa del cambio de modo de interacción a la pasarela mediante un evento que contiene la lista de escritorios activos. Esta lista se pasa al servidor red 5 que avisa a los clientes flash de si deben activar o no su escritorio.

V. PRUEBAS DEL SISTEMA

A. Ancho de banda consumido

Uno de los objetivos fundamentales de la pasarela es que pueda ser usada por el mayor abanico posible de sistemas. Esto incluye tanto sistema operativo y CPU como calidad de la conexión disponible. En este contexto, la disponibilidad de la plataforma Flash en los sistemas operativos más populares asegura el primer punto aunque, como veremos más adelante, plantea algunos problemas en cuanto a CPU.

En el apartado de los requisitos en cuanto a la conexión se han realizado numerosas pruebas durante el desarrollo de la pasarela para intentar identificar los principales problemas que puedan surgir. La principal conclusión obtenida de estas pruebas es la necesidad de que coexistan más de una pasarela con diferentes parámetros de calidad para poder ofrecer mejor servicio a conexiones de mayor y menor ancho de banda.

Por otro lado, en casos específicos como redes inalámbricas o redes con pérdidas nos encontramos con el problema de que TCP sea el protocolo sobre el que se apoya RTMP. Como sabemos [15], TCP no está pensado para estos escenarios e interpreta la pérdida de paquetes como congestión de red disminuyendo el tamaño de la ventana de congestión lo que reduce la velocidad de transmisión. Esta

situación afecta gravemente a los flujos de video y audio hacia los clientes de la pasarela haciendo que obtengan un rendimiento sensiblemente peor al esperado llegando incluso a cortarse la comunicación.

Este efecto negativo empeora aún más en escenarios en los que aparte de las pérdidas de paquetes coexisten grandes retardos en la red, como es el caso de los enlaces por satélite, por lo que en estos casos no serviría una solución basada en TCP.

B. Problemas de Eco

Durante las pruebas uno de los problemas que surgieron desde un comienzo fue el eco existente al conectar un cliente Flash. Esto surge principalmente por el nuevo tipo de usuarios que se conectan a la pasarela Web y su peor conocimiento de aspectos técnicos de los sistemas. Así como Isabel está más enfocada a su utilización por parte de un operador que se encarga de configurar el sistema la pasarela se caracteriza por su configuración más simple. Para ello abordamos diferentes soluciones que pasaran por que los usuarios no tuvieran que hacer pruebas de eco, ni tener por qué conocer detalles de cómo viaja el flujo de audio entre sus ordenadores y las sesiones de videoconferencia.

La primera solución que utilizamos fue la de activar el cancelador de eco que viene de por sí en el reproductor Flash, este cancelador intenta eliminar el eco producido cuando un usuario está escuchando la videoconferencia a través de altavoces y utilizando un micrófono de baja calidad como son muchos de los que vienen en las mismas Webcams. Este escenario no es el correcto para establecer una videoconferencia ya que los micrófonos de este tipo recogerán todo el sonido ambiente incluido aquel que es emitido por los altavoces, por lo que el nivel de eco es muy alto. En estos casos el reproductor de Flash funciona correctamente siempre que el único audio emitido y capturado sea a través de él. Es decir, no vale si a la vez están haciendo uso de la E/S de audio otras aplicaciones. Además hay casos en los que el resultado no es el esperado. De hecho Adobe en el propio API de Flash avisa de que este cancelador de eco no quita todo el eco en todos los escenarios posibles.

La segunda posibilidad, aparte de utilizar este cancelador de eco, fue la de implementar en la aplicación otro adicional que controlara la ganancia del micrófono a partir del nivel de audio que llegaba desde la videoconferencia. Esta supresión de eco, que está basada en sistemas antiguos de supresión de eco, supone que todo el audio que hay en una videoconferencia está producido por una sola persona en un momento dado, que es la persona que en ese momento está hablando, por lo que en realidad lo que hace es que si recibes audio desde la videoconferencia corta el audio de tu micrófono (si es que hay) porque supone que sólo puede ser eco. Lógicamente este no es el sistema de reducción de eco más eficaz, de hecho hoy en día se utilizan muchos otros que responden mejor a situaciones reales, pero sería el único que nosotros podríamos utilizar debido a que el API de Flash no nos permite procesar la señal de audio que llega de la videoconferencia ni la que capturamos de los micrófonos. Aún así decidimos que no era la mejor solución ya que hay situaciones en las que si alguien a través de su micrófono está emitiendo ruido hacia la videoconferencia los demás usuarios no podrán hablar correctamente.



Fig. 8 Capturas de pantalla de la pasarela Web

La tercera posibilidad fue la de implementar un sistema de Push-to-talk de forma que aquel usuario que quisiera hablar a través de la videoconferencia debería mantener presionado un botón mientras durara su intervención. Así se eliminaría el eco producido por este tipo de usuarios y sólo se introduciría eco cuando aquel que tuviera un mal ajuste de su micrófono y altavoces interviniera a la vez que otros usuarios. Este sistema está muy extendido en los terminales móviles (PoC [16]). Por lo que por su sencillez de desarrollo y uso pensamos que sería la mejor solución y la implementamos.

VI. CONCLUSIONES

En la figura anterior podemos ver una captura de pantalla de una sesión de videoconferencia con Isabel y la pasarela Web.

La solución planteada cumple todas las expectativas que teníamos en un principio. Se han incluido y permitido la interoperación de una compleja aplicación multimedia distribuida, como es Isabel, con aplicaciones tipo RIA. Las principales ventajas la disponibilidad inmediata desde cualquier terminal que disponga de un navegador con el plugin Flash instalado para participar en cualquier conferencia que se esté realizando, superados los debidos controles de seguridad.

Esto permitiría, por una parte, una integración con otras aplicaciones sociales tales como Facebook o Twitter permitiendo una mayor interactividad y una mayor facilidad de colaboración para usuarios nómadas. Por otra parte elimina algunas barreras de entrada para muchos usuarios tales como la instalación y configuración de dichas aplicaciones. Por ello el trabajo presentado aquí es un gran avance de usabilidad para la realización ubicua de complejas sesiones distribuidas.

Actualmente los trabajos futuros se están centrando en la evolución de la arquitectura aquí presentada a un esquema de SaaS (*Software as a Service*), lo que permitiría eliminar problemas de escalabilidad y configuración a usuarios finales.

Por otra parte se considera el portarlo a tecnologías HTML5-JavaScript si bien estas aún adolecen de las capacidades de captura de audio y video (cámara y micrófono) de los diversos terminales existentes, siendo perfectamente adaptable a cualquier dispositivo (tablet o móvil) en cuanto dichas capacidades estén disponibles.

REFERENCIAS

- [1] T. Noda and S. Helwig. "Rich internet applications". Technical report, <http://www.uwebc.org>, 2005.
- [2] J. Cerviño, P. Rodríguez, G. Huecas, J. Salvachúa, F. Escribano, "Marte 3.0: Una videoconferencia 2.0", VII jornadas de JITEL, septiembre 2008.
- [3] Adobe Flash: www.adobe.com/go/gntray_prod_flash_home_es
- [4] B. Hayes, "Cloud computing", Communications of the ACM, 2008.
- [5] J. Quemada, G. Huecas, S. Pavón, J. Salvachúa, "La Aplicación Isabel: Actividades Educativas interactivas a través de Internet", COIT, Jun-Jul 2008.
- [6] Java: www.java.com
- [7] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, Jan. 1996.
- [8] Real Time Messaging Protocol: http://www.adobe.com/go/tn_16631
- [9] Red5: www.red5.org.
- [10] Xuggler: www.xuggler.com/xuggler/
- [11] FFmpeg: www.ffmpeg.org/
- [12] TightVNC: www.tightvnc.com/
- [13] Flex SDK: www.adobe.com/es/products/flex/
- [14] Adobe, "Video File Format Specification Version 9".
- [15] "Impact of mobility on TCP/IP: an integrated performance study", IEEE Journal on Selected Areas in Communications, 1995.
- [16] Manzoni, P.; Ghosal, D.; Serazzi, G. "Push to talk over Cellular (PoC) – Architecture", Open Mobile Alliance, Feb. 2008.