Uso de Comet (Reverse AJAX) en los SIG. Prototipo de SIG colaborativo.

Diego Gómez Deck[1], Manuel de la Calle Alonso[2] ,Vidal Toboso[2] y Raquel Martínez[1]

[1] Consultar. diego@consultar.com
[2] Fomento y Medio Ambiente de Extremadura (FOMEX). Santiago Caldera Nº 4 Cáceres. mdelacalle@fomex.es

RESUMEN

En los últimos meses las empresa FOMEX y Consultar han estado desarrollando unos prototipos usando técnicas de Comet para demostrar su utilidad en el campo de los Sistemas de Información Geográfica.
Comet es una técnica de programación Web que consiste en dejar una conexión abierta entre el servidor y el navegador de internet. Con esta técnica el servidor web, envía datos al cliente ligero sin necesidad de una petición explícita. Esta tecnología permite el trabajo colaborativo en tiempo real, puesto que el cliente en un navegador web cualquiera recibe información sin necesidad de pedirla o actualizar la página web. Con este procedimiento se consigue llevar a los clientes ligeros que funcionan en cualquier dispositivo y sistema operativo funcionalidad que hasta el uso de estas técnicas no estaban disponible.
Esta técnica difiere de la programación web tradicional en la cual se enviaba una página completa para cada petición, y también difiere de Ajax en la cual los datos están ya en el cliente y son obtenidos por conexiones traseras

Técnica	Forma de enviar los datos con respecto a las acciones del usuario	Forma de enviar los datos con respecto a las acciones del navegado
Web Tradicional (Refresco de página)	Síncrono	Síncrono
AJAX	Asíncrono	Síncrono
Comet (Reverse AJAX)	Asíncrono	Asíncrono

Existen numerosos implementaciones de Comet, pero hasta ahora no ha sido muy usada en los SIG, donde puede tener muchas posibilidades sobre todo para trabajo de campo con dispositivos en los que no tenemos disponible más que un navegador de internet
Los productos que se han usado para los prototipos son gvSIG, Geoserver, openlayers, postgis, asteroid,ST2JS y SWT.

Key words: Comet, Reversed Ajax, Web, Ajax, gvSig, Postgis, Geosever, Asteroid, SWT.

INTRODUCCIÓN

El propósito de esta comunicación es intentar evaluar las posibilidades que existen para el mundo del SIG con el uso de la técnica Comet .

Existen bastantes implementaciones de esta técnica y empieza a ser ya usada en numerosos productos tanto libres como comerciales, pero no hemos encontrado aún muchas aplicaciones en el campo de los SIG y creemos que puede ser muy útil para conseguir dotar de mayor interactividad a clientes SIG ligeros, superando las limitaciones impuestas por ejecutarse dentro de un navegador.

Comet

Comet es una técnica de programación que permite que el servidor envié asíncronamente datos al cliente web sin que éste lo haya solicitado. Este procedimiento es conocido también como *server push*, *HTTP push*, *HTTP streaming*, *Pushlets*, *Reverse Ajax.*

Las aplicaciones web tradicionalmente han tenido una serie de limitaciones en relación a las aplicaciones de escritorio, esto, en los últimos años ha cambiado, de manera que cada vez más funcionalidad que encontramos en los clientes pesados, la tenemos en los clientes ligeros, y con técnicas como comet sin necesidad de instalar ningún plugin ni programa externo, funcionando en todas las plataformas y en todos los navegadores de internet.

Las aplicaciones tradicionales sólo podía recibir datos cuando los pedían al servidor, además con cada petición se recargaba toda la página en función del envio de los formularios, los primeros clientes ligeros sig son de esta manera, se usan los eventos del ratón para hacer una petición WMS que nos devuelve una imagen con el mapa pedido.

El siguiente paso ha sido el uso de AJAX (***A****synchronous* ***J****avaScript* ***A****nd* ***X****ML),* mediante esta técnica conseguimos el uso de comunicación asíncrona con el servidor en segundo plano, este procedimiento es una combinación de varias tecnologías que ya existían anteriormente. Con AJAX se consigue pedir únicamente los datos que necesitamos que sean actualizados, un ejemplo de aplicación SIG con AJAX es Google Maps, donde se utilizan conexiones traseras además de precacheo de imágenes.

Todavía queda por resolver la comunicación servidor-cliente sin enviar una petición, esto se ha hecho hasta no hace mucho usando plugins para el navegador, aunque esto presenta algunos inconvenientes (Necesidad de distintos plugins según navegador-Sistema Operativo, obligación de hacer instalaciones suplementarias para el usuario,...) o utilizando el procedimiento de AJAX con polling, esto significa tener un proceso que actualiza los datos cada cierto periodo de tiempo, los resultados son muy similares a los que se obtienen con comet pero tienen el inconveniente del derroche de recursos, ya que se abre una conexión sin saber previamente si es necesario, en

aplicaciones con poco cambio de información o con cambios muy previsibles puede funcionar correctamente.

El paso que falta es el que se consigue con comet. El servidor envía información al navegador sólo cuando lo necesita, de esta forma se disminuye la latencia y conseguimos la comunicación asíncrona en ambos sentidos. Esto dota a los clientes ligeros de una gran capacidad de interactividad, y nosotros pensamos que esta interactividad puede ser usada entre otras cosas para trabajar con SIG.

Hay dos formas básicas de hacer el comet, y dentro de esas dos numerosas implementaciones que utilizan unos procedimientos u otros, básicamente las dos técnicas principales son:

- Streaming. El servidor abre una conexión con el navegador y no se cierra nunca, el servidor envía datos que el navegador va interpretando al vuelo.
- Long Poll. El servidor mantiene una conexión durante un periodo de tiempo determinado o hasta un evento, si no ocurriese nada, el server cierra la conexión y la vuelve a abrir de nuevo.

Para realizar nuestro experimento hemos utilizado la implementación de comet asteroid [1]. Esta implementación es por streaming y consiste en mantener un Iframe abierto que recibe javascript y que el navegador interpreta según lo va recibiendo.

Nuestro aplicativo de prueba ha consistido en implementar OpenLayers dentro del framework SWT, de manera que podemos distribuir eventos usando todas las librerías de openlayers.

De hecho nuestra aplicación va a consistir en utilizar comet para distribuir eventos entre clientes ligeros sin petición previa. Es un caso típico de gestión de emergencias, la central avisa a los clientes de un evento concreto.

Arquitectura usada

El funcionamiento de la aplicación experimental sería el siguiente: desde un puesto de control que funciona sobre gvSig, donde tenemos la misma cartografía que los clientes ligeros, se envían eventos a un servidor web donde corre el framework SWT (Squeak Web Toolkit), éste se encarga de distribuir estos eventos entre los clientes. Por otra parte existe un nivel de interacción más, ya que desde los clientes se pueden enviar eventos al servidor y éste se encarga de distribuirlos a todos los clientes y al propio puesto de control.

Pormenorizando un poco el trabajo desarrollado:

- Postgre-Postgis: Simplemente almacenamos en la base de datos la información geográfica que vamos a mostrar en los mapas, alimenta al servidor de mapas y al puesto de control directamente por jdbc.
- GvSig: En gvSig hemos simulado un puesto de control donde el operador tendría toda la información, la cartografía y la capacidad de análisis sin las limitaciones de los clientes ligeros. Se ha programado una extensión de gvSIG que envía a el framework un evento en concreto indicándole su posición geográfica y el encuadre de la zona dentro de todo el territorio que es servido desde Geoserver. Por otra parte desde un navegador en el puesto de control se puede comprobar el estado de los eventos distribuidos a los clientes. La comunicación entre postgis y el framework se hace a través del envio de una URL por el método GET desde a gvSIG.
- Geoserver: Geoserver nada más que se decica a servir los mapas por protocolos estándar de forma que puedan ser visualizados por clientes tanto ligeros como pesados.
- SWT-OpenLayers. Es el framework que se encarga de distibuir los eventos lanzados por el puesto de control. SWT (Squeak Web Toolkit) es un framework de propósito general para el desarrollo de aplicaciones web que está desarrollado sobre squeak que es a su vez un subconjunto de Smalltalk.

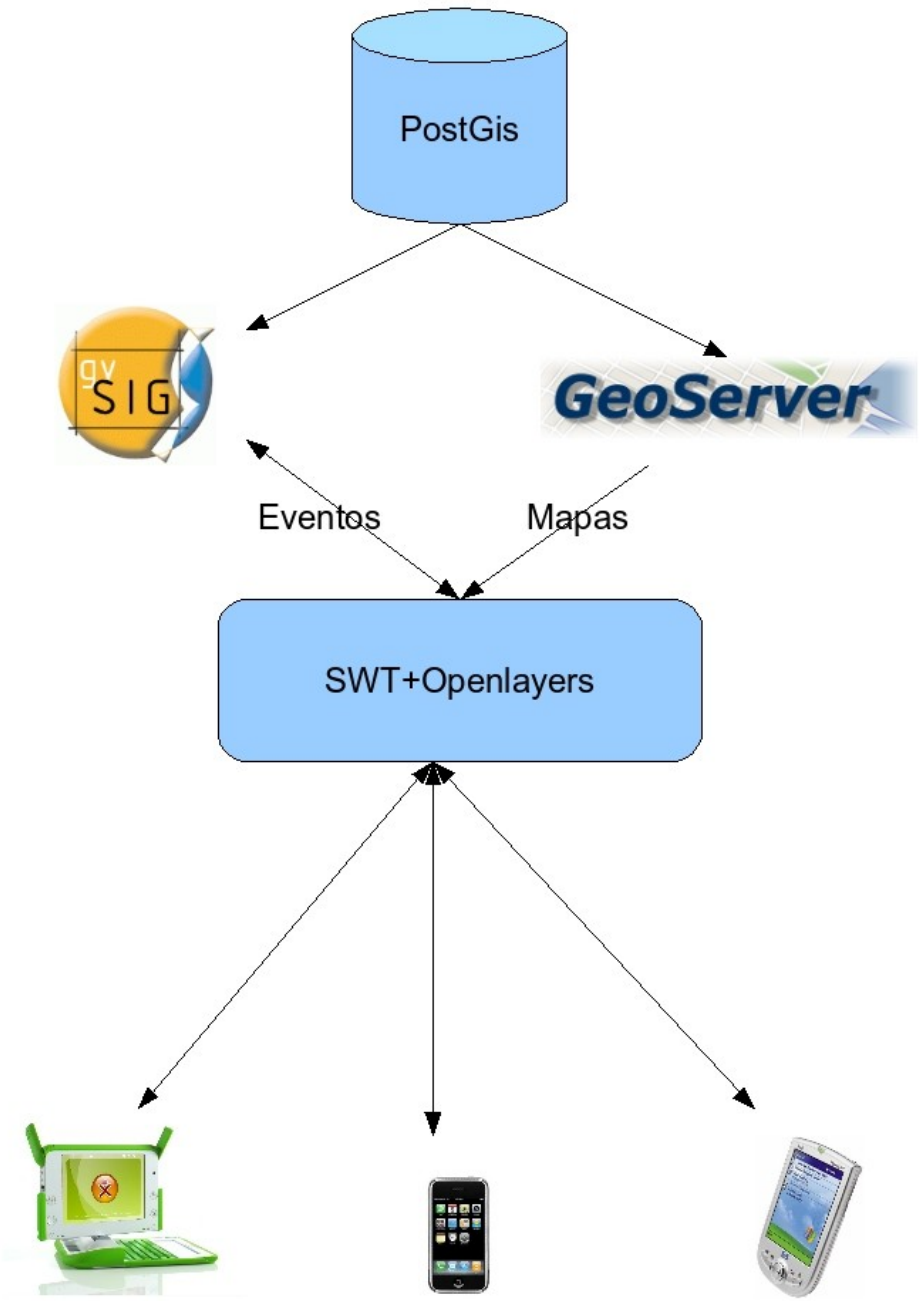

Figura 1: *Arquitectura de la aplicación.*

Éste framework se basa en que una aplicación tiene 2 partes bien definidas, una parte cliente, que corre en los navegadores en Javascript, y una parte servidora, que corre sobre Smalltalk con un web server basado en Comet. La parte cliente, no es programada en Javascript, sino que se desarrollo todo sobre Smalltalk. El framework provee un traductor a Javascript, denominado ST2JS. Este traductor respeta prácticamente toda la semántica Smalltalk y provee un juego de clases base para crear un mini-ambiente. Por lo tanto, al desarrollar se asume que en el cliente cuenta con un Smalltalk corriendo sobre cualquier navegador que soporte a Javascript. La parte servidora, es un servidor web con Comet, esta característica es la que vamos a aprovechar para nuestro desarrollo.

La comunicación entre los dos mundos

La comunicación entre el cliente y el servidor es transparente para el desarrollador, ya que el framework realiza la conexión entre ambos a través del uso de estándares web como RPC, con XML y usando JSON para serializar los objetos que pasan del cliente al servidor y viceversa. No todos los objetos son pasados por copia al cliente y no todo regresa al servidor, como sabemos, es muy costoso pasar tanta información sobre la red, es por ello, que el framework realiza ciertas optimizaciones sobre esta capa de comunicación. Los objetos del modelo de nuestro aplicativo son referencias remotas, por lo tanto, en el cliente tenemos objetos que representan a los objetos que están del lado del servidor, y solo los objetos que pasan por copia son aquellos objetos que son parte del soporte de la comunicación.

ST2JS

Este traductor respeta prácticamente toda la semántica de Smalltalk y la traduce a Javascript. Es decir, en Smalltalk podemos tener clases, instancias, mensajes de instancia, mensajes de clase, bloques, esto el traductor lo traduce a otros recursos, pues Smalltalk y Javascript no son semánticamente simétricos, pero que al final, terminarán haciendo lo que se describe en Smalltalk.

SWT y OpenLayers

El trabajo específico que se realiza sobre el Framework es modelar las clases y eventos necesarios para que Openlayers pueda funcionar ejecutado desde el servidor de SWT.

Lo que se hace es envolver los objetos javascript de Openlayers en objetos smalltalk, de manera que el javascript que genera el framework se entiende perfectamente con todas las demás librerías de Openlayers.

A partir de esto, los eventos generados por nuestro centro de control son distribuidos por el framework.

El uso de esta herramienta nos ofrece numerosas ventajas como:

- Escritura del código sólo en un lenguaje independizándos de los navegadores y sistemas operativos
- Modelo MVC Distribuido
- Uso de una implementación de comet directamente sin tener que tocar nada del código del servidor web.

CONCLUSIONES

Las posibilidades dentro del mundo SIG pensamos que pueden ser muy grandes, y que acercan las funcionalidades de los clientes pesados a la de los lilgeros, por otra parte, aunque el ancho de banda disponible es cada vez más grande, técnicas como esta permiten una mayor optimización de los recursos y una mayor posibilidad de colaboración en interacción entre los usuarios.

Si a técnicas como esta, añadimos el desarrollo de servicios como WPS, podemos estar próximos a no necesitar en la mayoría de los puestos de trabajos más que un navegador de internet (con las ventajas que esto conlleva) para tener toda la funcionalidad necesaria en un SIG

REFERENCIAS

- GOMEZ-DECK D.(2006) Asteroid,(A small Comet) http://wiki.squeak.org/squeak/5851
- WIKIPEDIA. Comet.
- KHARE R. (2005), Beyond AJAX: Accelerating Web Applications with Real-Time Event Notification
- GOMEZ-DECK D. ST2JS . http://www.squeaksource.com/ST2JS.html
- GOMEZ-DECK D. SWT. http://www.squeaksource.com/SWT.html
- http://ceibo.wordpress.com
- http://igosoftware.wordpress.com