Realtime collaborative WEB GIS.

Abstract

During the past months the companies Igo Software and *Consultar* have been developing some prototypes using Comet in order to show its utility on the Geographic Information Systems area.

Comet is a web programming technique based on leaving an open connection between server and internet browser. According to this technique the web server sends data to the thin-client, with out an explicit requirement. This technology enables collaborative work in real time because the client receives information in any web browser without asking for it or updating the website.

This process makes possible to bring the thin client that work on any device and operating system functionality to the use of these techniques were not available.

This technique is not like the traditional web programming which sends a full web for each request and it is also different from Ajax in which the data are already in the client and are obtained by rear connections.

Techniques	**Sending data process according to the user actions**	**Sending data process according to the web browser actions**
Traditional Website (web update)	Synchronous	Synchronous
AJAX	Asynchronous	Synchronous
Comet (Reverse AJAX)	Asynchronous	Asynchronous

There are several implementations of Comet, but at this time it hasn't been used in GIS, where it can bring many possibilities especially for field work with devices with no more than one web browser.

The products that were used for these prototypes are; *gvSIG, Geoserver, openlayers, postgis, asterois, ST2JS and SWT.*

Keywords*: Comet, Reversed Ajax, Web, Ajax, gvSig, Postgis, Geosever, Asteroid, SWT.*

Introduction

The purpose of this communication is try to evaluate the chances for GIS world using Comet.

At this time there are quite of few implementations of this technology and it's being used in many free or even commercial products, but we haven't found yet many applications in the GIS area and we believe that it could be very useful in order to give more interactivity to the GIS thin clients. It will be possible overcoming the limitations imposed by executing within a browser.

Comet
Comet is a programming technique that enables the web server to send asynchronous data to the web client without any previous requirement. This process in known as *server push*, *HTTP push*, *HTTP streaming*, *Pushlets*, *Reverse Ajax.*

The web applications has presented many different limitations related to the desktop applications. It has changed during the last years, so it is possible to find more functionality in thin clients that were founded in the heavyweight clients. It is possible using techniques such as Comet, without installing any plug or external program, running in all platforms and all internet browsers.

The traditional applications could only receive data when they asked for it to the server, and because of the forms sent the whole page was reloaded. The first GIS thin client work this way, the mouse events are used to make a WMS petition that brings back an image with the map asked before.

Next step was using AJAX (Asynchronous Javascript And XML). With this technique we got the use of the asynchronous communication with the second field, this procedure is a combination of several technologies that already were used. AJAX enables asking only the data that need to be updated. An example of using GIS and AJAX together is Google Maps, where rear connections are used instead of image precaching.

The communication between server and clients is not solved yet. The way it was made a few time ago was using plugins for the browser, even the troubles that appeared (There were needed different plugins according to the browser operating system. The user had to make some additional installations...). Other way was using the process with AJAX and polling that means having a process that updates the data each certain period of time . The results are very similar to the ones gotten by Comet, but the problem is the waist of resources because of the opening of a new connection without knowing if it was really needed before. It could work properly in applications with low information exchange.

The last step is related to Comet. The server sends information to the browser only when it is needed, thus decreases latency and get asynchronous communication in both directions. This gives thin clients a great capacity for interactivity, and we think that this interactivity can be used for applications such as GIS.

There are two basic ways to do the Comet, and different implementations that uses several procedures or others. The main techniques are:

- *Streaming*: The server open a connection to the browser which never get closed. The server sends data that the bowser interprets on real time.
- Long Poll: The server keeps a connection during a period of time until a new event, if nothing happens the server closes the connection and re-start it again.

In our experiment, we used the implementation of Comet Asteroid [1]. This is an Streaming implementation that consists on keeping an open IFrame that receives Javascript and interprets it as soon as it receive it.

Our test application was based in the implementation of Openlayers into the SWT framework so we can distribute events using the whole Openlayer libraries. In fact, our application will consist in using Comet to distribute events between thin clients without a previous request.

In a typical case of emergency management, the main office notifies the clients if there is a specific event.

Used Architecture

This is the way how the experimental application works:

From the checkpoint, that works in gvSig and has the same cartography than the thin clients, events are sent to a web server which runs the SWT (Squeal Web Toolkit) framework which is responsible of the distribution of the events between the clients and even to he checkpoint.

Summary of the developing of the work:

- Postgre-Postgis: It is just to store the geographic information that will be shown on the maps in the data base. Feeds the map server and the checkpoint directly by jdbc.
- GvSig: In gvSig we simulated a checkpoint where the operator would have the whole information, the cartography and the analysis capability of the thin client limitations.
- Geoserver: Geoserver just serve the maps by standard protocols, so it can be seen by thin or heavyweight clients.
- SWT-Openlayers: This is the framework that have to distribute the sent events by the checkpoint- SWT (Squeak Web Toolkit) is a framework for general purpose used to the development of web applications. It is developed on Squeak, that is a modern implementation of Smalltalk.

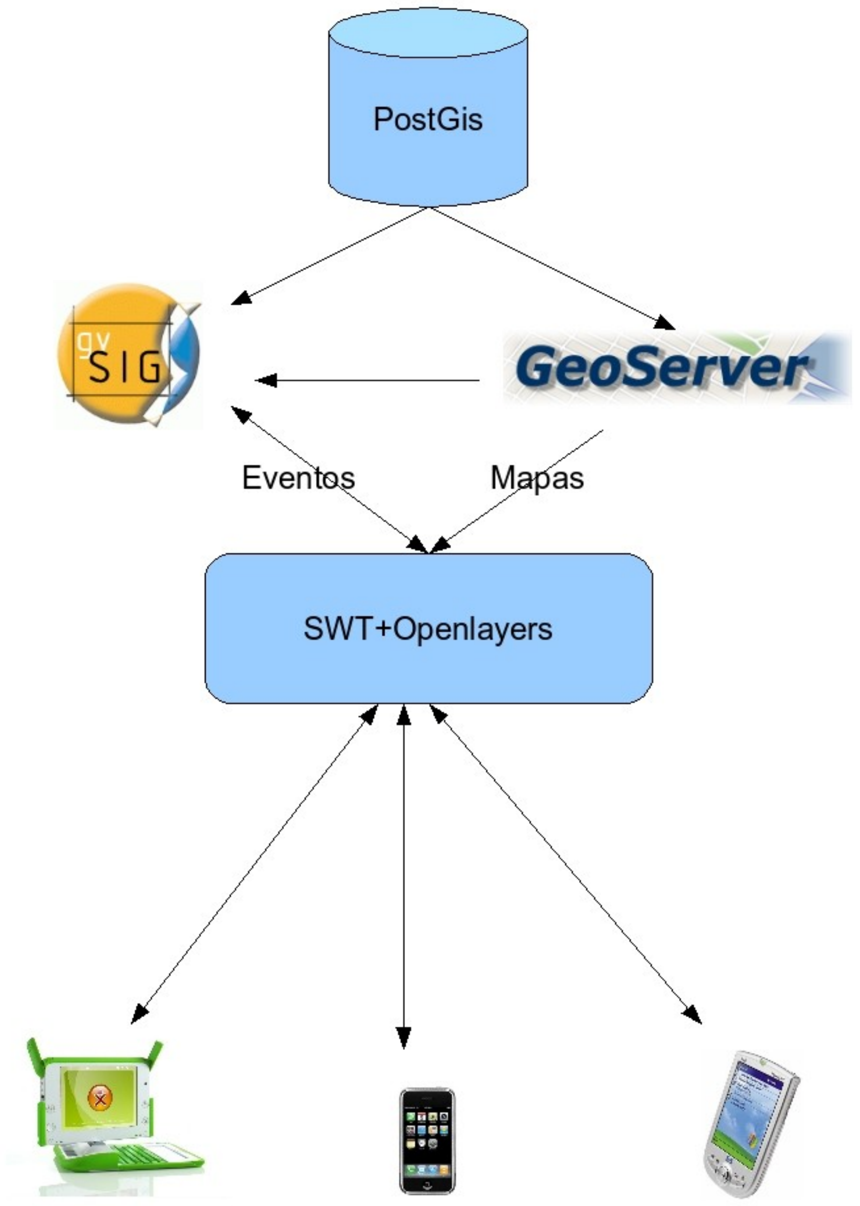

Figure 1: Application *architecture*

This framework lays on an application that has 2 well defined parts. One of this parts is the client, which runs in the browsers in Javascript and also a server part, which runs on Smalltalk with a web server based on Comet. The client part is developed on Smalltalk not programmed on Javascript. The framework gives a translator to Javascript named ST2JS. This translator respects the whole Smalltalk semantics and provides a set of base classes in order to create a mini-smalltalk environment on the browsers. During the development time it is assumed that there is a Smalltalk running on any browser that supports Javascript.

The server part is a web server with Comet, this is the characteristic that we will use for our developing.

Communication between two worlds.
The communication between client and server is transparent to the developer because the framework enables the connection between both of them through web standards as RPC with XML and using JSON in order to serialize the objects that pass from the client to the server and turns back.
Not all the objects pass through copy to the client and also not everything turns back to the server, as we all know is very expensive to pass so many information on the net, so the framework makes certain optimizations on this communication layer.

The objects of our application model are remote references, that's why there are in the client objects representing those who are in the server side, only those objects that pass through copy are those who are part of the medium of communication.

ST2JS

This translator respects the whole Smalltalk semantic and translates it to Javascript. It means that there can be classes, messages, class messages, blocks in Smalltalk and the translator will translate it to equivalent Javascript structures. Smalltalk and Javascript are not symmetrical semantically, but they will do what is described on Smalltalk.

SWT and OpenLayers

The specific work done on the framework is modeling the classes and events that are needed so the Openlayers could be able to operate running from the SWT server.

The Javascript objects are wrapped in Smalltalk objects, so the Javascript that was generated by the framework can have a perfect communication with the whole Openlayers libraries.

According to this, the events generated by our checking point are distributed by the framework.

Using this tool brings us several advantages such as:

- Be able to write in just one language independent from the browsers and operative systems.
- A MVC distributed models.
- Be able to use a Comet implementation, with out changing anything from the web server.

Conclusions

We think that possibilities inside GIS might be huge and brings closer the heavyweight clients functionalities to the think clients. In the other hand although the available bandwidth keeps growing, using this kind of techniques makes possible a better optimization of the resources and a bigger possibility for collaboration and interaction between distributed users.

Adding the developing of services such as WPS to this techniques, maybe in the working places there won't be needed anything than an internet browser (with all it benefits) to have all the functionality needed by GIS.

References

- 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