

# The Web Around The Corner: Augmenting the Browser with GPS \*

Davide Carboni  
CRS4  
Parco Scientifico Polaris  
I-09010 Pula (CA) - Italy  
dcarboni@crs4.it

Andrea Piras  
CRS4  
Parco Scientifico Polaris  
I-09010 Pula (CA) - Italy  
piras@crs4.it

Sylvain Giroux  
Dep. of Mathematics and Computer Science  
University of Sherbrooke  
Sherbrooke, Canada  
sylvain.giroux@dm.usherb.ca

Stefano Sanna  
CRS4  
Parco Scientifico Polaris  
I-09010 Pula (CA) - Italy  
gerda@crs4.it

## ABSTRACT

As programmable mobile devices (such as high-end cellular phones and Personal Digital Assistants) became widely adopted, users ask for Internet access on-the-road. While upcoming technologies like UMTS and Wi-Fi provide broadband wireless communication, Web services and Web browsers do not provide any sort of location-awareness yet. As GPS receivers get cheaper, positioning devices will be embedded into commercial mobile devices. Thus, the position of the user can be used to filter and tailor the information presented to the user as already done for language preferences and user-agent.

This paper describes early results of an ongoing project called GPSWeb, which aims to provide GPS support for Web browsers and an application model for Location-Based Services. It introduces the Location-Based Browsing concept that enhances the classic Webuser-Website interaction

## Categories and Subject Descriptors

H.5.4 [Hypertext/Hypermedia]: Navigation; H.3.3 [Information Search and Retrieval]: Information filtering; J.m [Miscellaneous]

## General Terms

Experimentation

## Keywords

LBS, location-awareness, GPS, browser, Javascript

## 1. INTRODUCTION

Location-based services (LBS) have been identified as next mass-market opportunity for device manufacturers, mobile network operators and content providers[2]. Automatic location information

\*(Produces the WWW2004-specific release, location and copyright information). For use with www2004-submission.cls V1.0. Supported by ACM.

Copyright is held by the author/owner(s).  
WWW2004, May 17–22, 2004, New York, New York, USA.  
ACM 1-58113-912-8/04/0005.

can drive data filtering faster than other manual approaches. Technology infrastructure is mature to support this new kind of services: devices, networks and Global Positioning System (GPS) receivers provide programmability (necessary to implement mobile browsers), connectivity and location-awareness. While hardware is rapidly evolving, software has simply been ported to new devices with optimizations and scalability in mind: today's mobile browsing is still "Browsing with a small screen".

Two main issues comes with LBS: how to obtain the user location and how to use such an information. The former involves the integration of client side Web tools with positioning systems like GPS. In our approach, the Web browser is aware of the actual position and insert it inside HTTP requests once determined that, on the server side, there is a Web service able to process such an information and provide a time and place relevant output. In on-the-road Web browsing a query such as "restaurant" means that the user wants to know where is the nearest (place) restaurant opened (time).

## 2. HTTP REQUESTS

HTTP uses standard and extended headers to transfer request-response metadata, enabling a simple and efficient mechanism for client-server communication.

In Location-Based Browsing (LBB) the position of the user is a new metadata of each HTTP request. We propose a simple header named "User-Location" whose format is described below:

```
User-Location: LATITUDE;LONGITUDE
```

For instance, `User-Location: 8.89912;38.88354` is a valid position. If actual position is not available (e.g., due to poor or absent GPS coverage) the client should send the "not available" error:

```
User-Location: N/A
```

that informs the server that the request comes from a location-enabled browser, but the actual position cannot be determined. Finally, if the user wants to disable location facility the client sends the "not enabled" value:

User-Location: N/E

This way the server is aware that the client is equipped with a GPS and may ask the user to enable the position casting in order to get extra services or improve browsing experience.

### 3. GPS-ENABLED BROWSER

To enable Location-Based Browsing and exploit enhanced services, browsers have to retrieve actual position and embed location information into HTTP queries as discussed above. Notebook, PDAs and mobile phones have been chosen as deployment platforms. They mainly differ for processing power and displaying capabilities, but share the ability of connecting to a GPS with serial or Bluetooth wireless connection.

For the notebook platform, Mozilla has been chosen as reference browser. Mozilla is the straightforward choice for the implementation of Internet client applications because it is Open Sourced, it is reliable and efficient, it is documented and supported by an active community of programmers. Moreover, Mozilla browser is completely customizable by virtue of a large programming interface which allows developers to add new functionalities using XUL (a user interface description language) and Javascript. This development model allows to implement new functionalities and even to write new stand-alone application without writing a line of native code and enabling development of reliable, cross-platform and robust Internet clients.

The GPSWeb implementation in Mozilla is based upon a listener object written in Javascript which observes any new HTTP request adding the position information in the headers of the request. The extension requires the user to have some GPS hardware connected to their computer and the GPSd Java daemon, which parses the data supplied by the GPS unit, to be running.

The minimal configuration of the observer is performed by means of a toolbar where the user can activate/deactivate the observer and set the name of the GPSd device in the file system. Fig. 1 shows the user interface in the Mozilla/Firebird browser (Firebird is a new stand-alone browser based on the Mozilla code. The GPSWeb extension runs on the "old" Mozilla suite too).

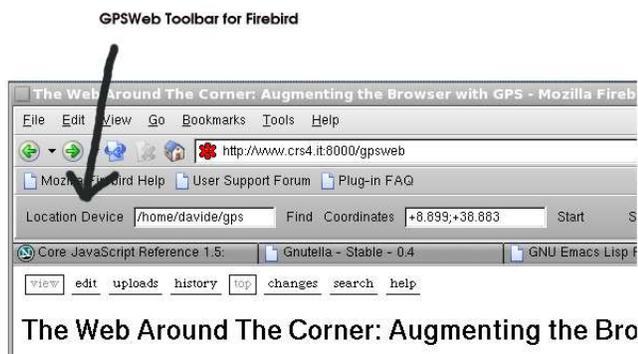


Figure 1: GPSWeb extension running on Mozilla/Firebird

Although Mozilla has been ported to most known Linux-ARM distribution (familiar), it requires lot of memory to run and it is still not suitable for entry-level PDAs. Therefore, to enable LBB on PDA device, a light version of the Konqueror web browser (Konqueror-embedded) is going to be enhanced with GPSd interface. Finally, we aim to exploit LBB on mobile phones but at the moment an open source browser is still not available for such a platform.

### 4. APPLICATION SCENARIO

To date, position information is obtained by means of manual user input. Using a browser enabled with GPS-extensions, such preliminary phase is no longer required. Existing applications, like tourist information providers, can be simplified and new ones can be developed, in various domains such as e-learning, entertainment and tourism. For instance, forums collecting remarks, opinions and experiences of tourists about a particular city or spot.

A tourist could like to receive advices and opinions written by others and eventually adds his own remarks. To date, he must unravel among hundreds of threads and messages. Adding position information in the HTTP headers, the forum application can be rewritten in such a way that new discussion threads are created and associated to a spot.

Users that reach the same spot lately can read the threads and add new messages if they want.

The functionalities described above can be grouped in three categories: create or add new geo-referenced information, perform searches according to the actual position of the user and provide the appropriate translation from legacy to Web formats.

Such functionalities, common to a wide set of applications such as online databases, blogs and news portals, have been implemented for the tourist forum application in a module called Location Based Browsing Engine (LBBE). Such a module has been designed to be flexible, easily extendable and adaptable to different applications. It consists of a context selector for defining the extent of areas around spots and the type of goal, one set of datagetter and one of datasetter.

Datasetters update or insert new geo-referenced information while the same information could be retrieved by means of dataggetters which also compose Web pages displayed to the user.

The goal identifies the GIS layer in which the user is interested. In fact, different dataggetters can be coupled to the same spots but for different kind of information. For example, in the same city we can have one dataggetter for typical restaurants, another one for common restaurants and pizzerias and a last one for monuments.

### 5. CONCLUSIONS

Location-based browsing enables new application scenarios but the full integration between Web technologies and positioning systems is still missing. This paper presents a simple solution to integrate Web browsers such as Mozilla and Konqueror for PDA with GPS devices and explain how GPS coordinates can be inserted in HTTP requests. Moreover, a simple application model for Web services is also discussed. The automatic casting of user position raises some privacy issues[1]. Thus, the user must have complete control on the GPS extension. The GPS casting can be activated and de-activated with a button click, and a black list can be managed to disable and a white-list to enable the extension automatically.

### 6. REFERENCES

- [1] L. Barkuus and A. Dey. Location-based services for mobile telephony: a study of users privacy concerns. In *9th IFIP TC13 International Conference on Human-Computer Interaction, INTERACT 2003*, July 2003.
- [2] B. Rao and L. Minakakis. Evolution of mobile location-based services. *Commun. ACM*, 46(12):61–65, 2003.