Geolocation in the mobile web browser

Mattias Rost, Henriette Cramer[1], Nicolas Belloni, Lars Erik Holmquist
Mobile Life Centre
Kista, Sweden
{rost,henriette,nicolas,leh}@mobilelifecentre.org

ABSTRACT
Current mobile browser capabilities make it possible to quickly develop advanced mobile location based services without having to write device specific software, or build custom hardware. We here describe three web applications exploring using location within mobile browsers (TågAlong, NearMe and LocalURL). These explorations show clear potential for using geolocation in the web browser in order to reach a larger user base, with a greater variety of devices, thus allowing for UbiComp researchers to explore the effects of specific services and applications on a larger scale. We discuss the services, as well as the potential and challenges with using the user's location directly in the browser.

Author Keywords location-based services, mobile services, geolocation

ACM Classification Keywords D.2.m [Misc]

General Terms Experimentation

INTRODUCTION
On the desktop, an increasing amount of services and applications already run through the web browser – email clients (e.g. Gmail), word processors (e.g. Google Docs), and even photo editing software (e.g. Pixlr). The advent of HTML 5 has the potential to increase this dramatically - especially for mobile applications. Mobile development is now moving quicker than desktop development, and mobile web browsers are getting increasingly powerful. The iPhone App Store, Android market and other mobile phone vendors' app stores together now feature hundreds of thousands of apps. We believe this is only the beginning of the explosion of mobile services and applications; an even bigger boom will occur when many of these apps will be written for the mobile web instead [4]. As the capabilities of the browsers are increasing, more powerful, advanced, and innovative applications and services can be implemented for virtually any mobile phone.

Location-based services (LBS) and context-aware applications have a long history in the field of ubiquitous computing and human-computer interaction [2][6]. Many localization techniques exist, all with their specific tradeoffs [5]. Recently there has also been an increase in commercial location-based services (e.g. Google Maps, Yelp), and location sharing services (e.g. Google Latitude, Gowalla, Foursquare) with rapidly growing user bases (Foursquare >1.7M users, Gowalla >250k users). Most of the applications for using these services have been implemented for specific platforms. Even if the applications are written for the most popular brands of devices, many handsets are left out. For instance the iPhone still only accounts for 3% of the whole mobile phone market [3]. As browsers are now starting to implement the support for geo-location through Javascript APIs, many of these services can now instead be realised as mobile web applications.

We here describe explorations with location in the mobile browser. Three applications and services have been implemented for usage within a mobile web browser. Two of these services use a geo-location API; one service uses self-reported location. Based on our experience with these services, we discuss the potential and challenges with using the user's location directly in the browser.

SELF-REPORTED LOCATION
An early prototype using self-reported location was the *TågAlong* application (previously known as *Subway Friendfinder* [1]). TågAlong is an application for finding your friends in the subway. When starting the project the idea was to try and create a service that would be accessible to virtually any handset by providing a web app accessible via a mobile device's web browser. At the time of implementation, the location of the handset was not accessible through the web browser, and thus had to be acquired in a different way. In our solution, the user provided the app with their own location by picking a subway stop, and were then presented with a choice of which subway train they would get on. A two-tiered approach was thus used in which the final location (a specific subway train) can be chosen after choosing a coarser-grained location (the subway station).

Current popular 'check-in' location-sharing services like Foursquare and Gowalla also use self-reported location, providing users with control over the location they share with others and helping to overcome potential localisation issues. A two-tiered approach in which places or venues that can be manually selected by the user are suggested, based on a handset's current location is a scalable approach. Previously, platform-specific apps and specific handset capabilities were needed to achieve this. However, we can now use geolocation in the web browser to the same effect.

GEOLOCATION IN THE WEB BROWSER
HTML 5 includes a specification for a geo-location API. Through this API the client can request the location of the

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

[1] Cramer's work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme

user through Javascript. So far, this API has been implemented only to a limited extent. However, while waiting for broad support of this API, most browser vendors have implemented *their own APIs for requesting the location of the user*. Therefore, in order to make sure the web applications run on as many current browsers as possible, applications will have to check which browser the client has and uses the APIs that work for that client.

Both in the W3C geo-location API, and the custom-made APIs, the location can be requested at two levels of accuracy – accurate or coarse grained. The accurate location is a location given by a GPS chip whereas the coarse grained location is given by different types of base station triangulations (cell-tower based, wifi based, etc, depending on the implementation). Acquiring more accurate location will take more time, and will drain batteries. In our examples, the exact location is not crucial and we have therefore chosen to use the coarse grained location in order to allow for a quicker responsiveness.

NEAR ME

The first application we implemented using the geo-location API is *Near Me*. This is a web page that simply lists who is close to your current location. The motivation for this was to test location in the browser and using existing social networks. The service uses *Facebook Connect* (developers.facebook.com/docs/) for user authentication in order to remove any need to register for the service, and to leverage users' existing social networks.

When the web app's page is loaded, it asks the browser for location of the device. The location is then sent to a server. On the server, user locations are stored in a database and the current user's location in the system is thus updated. Once updated, the database is queried for people nearby, sorted by distance, and sent back to the web client. The client renders the list of nearby users sorted by distance, stating the name, the distance, and the time when the location was updated (Fig. 1). Future work includes exploration of various presentations of distance information, as well as the pros and cons of integration with more meaningful semantic locations (e.g. from existing LBS).

LOCAL URL

As a second experiment, we explored the concept of finding web pages relevant to locations near you (e.g. time table websites near a station, company websites when passing the company's headquarters). Our LocalURL application allows adding web pages to users' (current) location, and lists web pages added by others as relevant to your current location. When loading the web app's page, it asks the browser for its location. The location is sent to the server and the server returns web pages that have been added nearby sorted by distance (Fig. 2).

CONCLUSIONS

In this paper we have presented two experimental mobile

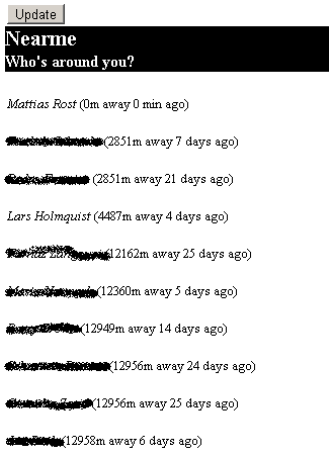

Figure 1 NearMe

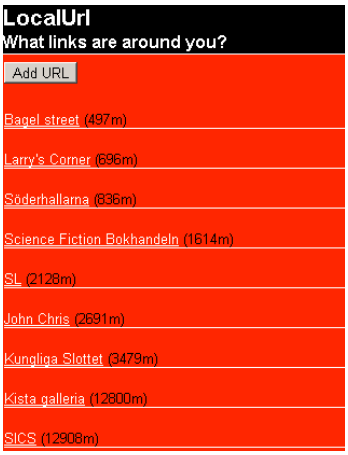

Figure 2 LocalURL

web applications leveraging the location of the device directly in the browser. This has clear potential for UbiComp research, as we can now quickly write advanced mobile location based services for a large variety of handsets; we no longer need to write device specific software, nor build custom hardware. This enables UbiComp researchers now relatively easily reach wide audiences, which would allow to explore the effects of specific services and applications on a larger scale. While a number of issues (such as browser and API compatibility) will have to be addressed to reach the full potential of using geolocation in mobile browsers, we can now aim at developing strategies for promotion and distribution to take advantage of this opportunity. We intend to continue our experimentations and spread applications to gain some use in order to gain experience of how they are used.

REFERENCES

1. Belloni, N. Holmquist, L.E. and Tholander, J. See You on the Subway: Exploring Mobile Social Software. *Ext. Abstracts CHI 2009*, ACM Press (2009)
2. Chen, G. and Kotz, D. *A survey of context-aware mobile computing research*. Technical Report UMI TR2000-381, Dartmouth College, 2000.
3. Gartner, Statistics of World Wide Mobile Phone Sales: http://www.gartner.com/it/page.jsp?id=985912, April 2010.
4. Holmquist, L.E. The Age of the Mobile Mash-Up, *TechCrunch*, http://bit.ly/aQmUNB, May 29th, 2010.
5. Munoz, D., Lara, F. B., Vargas, C., and Enriquez-Caldera, R. *Position Location Techniques and Applications*. Academic Press (2009).
6. Raper, J., Gartner, G., Karimi, H., and Rizos, C. 2007. A critical evaluation of location based services and their potential. *J. Location-Based Services* 1, 1 (2007), 5-45.