

Experiencing Location-Aware System: A Case Study

Omar Mubin¹, Abdullah Al Mahmud¹, and Suleman Shahid²

¹ Department of Industrial Design
Eindhoven University of Technology
Den Dolech 2, 5600 MB, Eindhoven, The Netherlands
{o.mubin, a.al-mahmud}@tue.nl

² Tilburg centre for Creative Computing
Department of Information and Communication Sciences
University of Tilburg, Tilburg, The Netherlands
s.shahid@uvt.nl

Abstract. In this paper, we identify practical and technical issues that we experienced during designing and experimenting a location aware system. Our system was designed using the Spot location aware system and the Event Heap server under the TeamSpace framework. We conducted a series of test runs of the application with developers, using wireless mobile devices such as the HP IPAQ 5500. From our experience, it shows that currently the Spot system is not viable enough for designing an imminent location aware application. Finally, this paper concludes with general remarks about how our findings and recommendations can be taken into account for the improvement of the Spot system and consequently for similar location-aware systems.

Keywords: Location-aware systems, Mobile devices, Ubiquitous computing.

1 Introduction

One of the recent developments in Ubiquitous computing and Mobile Computing has been in wireless technology. This has led to numerous advancements and breakthroughs in location aware computing. Localized information has become an integral part of different mobile devices. Mobile context of use varies dynamically with changes in location. A system is context-aware if it can extract, interpret, and use context information and adapt its functionality to the current context of use [4]. The methodology of context-aware applications has been studied in different application areas, e.g. tourist guidance, exhibition guidance, e-mail, shopping, mobile network administration, medical care etc. In these studies, the location of the user is the main attribute used in the context-adaptation. Though these studies identified general limitations of location aware applications, none of them discussed the technological challenges in detail. The state of art location aware systems are for example Cybreminder [1] and PlaceMail [5]. Even for these studies contemplation of technological issues is generally avoided, or is not the focus of attention. Location Aware systems operate in social settings hence there are a number of complex interactions in play, not only between the user and the system but also between the components of the system and between the system and the environment. This paper presents the

technical predicaments via a critical evaluation of an application built upon a location aware system known as the Spot system [6]. The server is the Event Heap provided by TeamSpace [2]. The paper discusses the design aspects of the location aware application. The study was carried out in principal to build and evaluate a pervasive computing application using the Spot System for the real world. The goal of our evaluation was not to improve the usability of such systems, as we feel there were first hand technical issues with regards to the Spot System that needed to be addressed before conducting user evaluations.

2 Application Design

Our application was built upon two basic components: i) The Spot (location) system [6] and ii) The Event Heap server that had been deployed within the TeamSpace infrastructure [2]. Moreover the application was deployed and tested on the HP IPAQ 5500. It should be noted that the Spot system that we employed was a specific version for the IPAQ namely: Adaptive Context-aware Services (ACAS-Spot) [3].

2.1 The Spot System

While interacting with the Spot System users have the freedom of defining their current position ('a Spot') and are prompted by the Spot system whenever they enter the same location again. Moreover the Spot system also provides an option of adding/attaching labels or additional strings to any Spot.

Each Spot profile is built on the client's side at time of defining the Spot. A Spot profile consists of information and attributes regarding the Spot. When a user arrives at a physical location, his/her current Spot location is computed by repetitive scanning to find a match. It goes without saying that if it is a newly discovered location the user is prompted to define a Spot; otherwise the current Spot location is identified. Given in Figure 1 is a snap shot of the Spot interface. According to the Figure 1, the Spot System has located Spot: "Grup_11" with 98% accuracy.

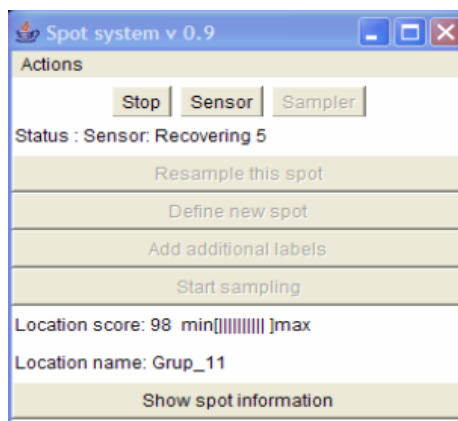


Fig. 1. The Spot System Interface

2.2 Application Scenario

In order to implement the application a practical scenario was conceptualized. The context was as follows:

In a university, there are numerous food joints and cafés. Whenever a student comes in the range of an eating joint, which is having its own TeamSpace server, the menu for the day of that eating joint will pop up on the display of the IPAQ (We assume that students have a IPAQ 5500 deployed with the Spot System and our application). In addition, whenever the student goes out of the range of the Spot the menu should disappear. Each eat joint has its own Teamspace server and a Spot has been defined in its name by the Spot system.

3 Results and Discussion

Spot System. We found considerable time lag in the Spot system as it did not refresh sensing results immediately. When a new Spot was defined it took about 20 iterations of Wi-Fi scanning. An ordinary sensing operation takes about 3 iterations [6]. Moreover the problem is compounded by the fact that there is no feedback from the Spot system.

Currently the Spot system cannot handle multiple redefinitions of Spots at one location. Even if there is a defined Spot found in the current location, the user is still allowed to define a new Spot. It would be preferable to support only the addition of new labels for an already initialized Spot location. For a Spot, users can simply share semantic information via their context by simply defining new labels for the same location. However this feature of adding labels is compounded by the fact that at times the system is incapable of differentiating between Spots that lie in the vicinity of each other ~5 meters[6]. So the next time the user arrives at the same location there is no assurance about which Spot will show up. Currently the system does not use a user login mechanism (only nick names are supported) and it is easy to imitate another user. A Spot should simply be associated with one particular login; other users should only be able to lookup the Spot and attach labels to it.

Another critical issue that was observed was that Spot system is not device independent. The same Spot is treated differently by different devices. For example a Spot sensed on the IPAQ did not have the same Spot ID when the same Spot was sensed on a laptop.

Event Heap. The Event Heap does not guarantee ordering of events from different sources [2], even though at first sight it may seem to function as a simple FIFO structure. With respect to multiple clients (two or three) the Event Heap handles synchronization suitably such that all clients are served by the Event Heap server. However when we tested the application with three different devices (two different IPAQ's and a laptop), all 3 clients would get the menu displayed but the Event Heap server was not differentiating the two IPAQ's and was treating them as users with same ID's even though their IP's were different. For our application this was not a serious dilemma as it did not support customized or personalized interactions.

4 Conclusion

We have presented a crude evaluation of an indigenously developed location aware system. The system was designed using the Spot location aware system and the Event Heap server under the TeamSpace framework. We feel that the Spot system is in its primitive stages and more refinement needs to be carried out in terms of both hardware and software. We propose the next evaluation amongst potential users. Most of the issues identified are more technical in nature and it would be rather demanding to expect from novice or naive users that they would ascertain these problems. This is more so the case in location aware systems; for to acknowledge the usability aspects the application must function correctly (in a social context).

References

1. Dey, A., and Abowd. G. CybreMinder: A context-aware system for supporting reminders. In Proceedings of Symposium on Handheld and Ubiquitous Computing, pp. 201–207 (2000)
2. Event Heap. Available from: <http://dsv.su.se/fuse/int8/lab2/eventheap.pdf>
3. Kanter, T., Kilander, F., and Jansson, C.G.: Adaptive and Context Aware Services, Status report for the ACAS project (2003)
4. Korkea-aho, M.: Context-aware application surveys. Available from: <http://www.hut.fi/mkorkeaa/doc/context-aware.html> (2000)
5. Ludford, P.J., Frankowski, D., Reily, K., Wilms, K., and Terveen, L.: Because I carry My cell phone Anyway: Functional location-Based Reminder Applications. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 889–898. ACM, New York (2006)
6. Spot System. Available from: <http://dsv.su.se/fuse/int8/lab2/SpotDoc.pdf>