Ubiquitous Computing for the Public

Patrik Osbakk
Jayway AB
Box 571, 201 25
Malmö, Sweden
patrik.osbakk@jayway.se

Erik Rydgren
Jayway AB
Box 571, 201 25
Malmö, Sweden
erik.rydgren@jayway.se

ABSTRACT
Despite the progress made in Ubiquitous Computing since it was first envisaged back in 1987 [1], its use is still isolated to the research community. In this paper, we present why we feel the use of off-the-shelf mobile information devices may aid in bringing ubiquitous computing to the public. We also outline an ongoing project that we are undertaking in hope to introduce ubiquitous computing to a wider audience together with some preliminary results.

Keywords
Ubiquitous Computing, Mobile Devices, Bluetooth

1. INTRODUCTION
Has the progress made over the last decade brought us closer to being able to achieve the vision of ubiquitous computing? Yes, most definitely. Much experience has been gained from the many ubiquitous computing projects that have been performed world-wide, covering such topics as location sensing [2], context-awareness [3], privacy [4], fieldwork [5], health care [6], etc. Even though there is still a lot more to do within all of these areas and many more, the field as a whole has matured and perhaps most importantly it has been widely recognised and established. Significant progress has also been made with the underlying technologies. Today inch-scale computing devices are readily available to the masses. These come with various types of wireless connectivity (gprs, bluetooth, wifi), have decent battery life, and most importantly can be bought at reasonable prices.

However, despite the progress made, the benefits of ubiquitous computing are still largely restricted to the research community. Few applications and systems find their way into the hands of the general public, unless they are part of an organised study. This is regretful and will increasingly become a problem if the vision of ubiquitous computing is to be achieved. It should not be forgotten that the overall aim is *not* the development of novel applications nor is it to simply distribute thousands of invisible and interconnected devices into our environment. The aim is to aid people in their daily lives. After all the vision was born when attempting to reduce the complexity of the personal computer and the attention it requires [1]. As Mark Weiser so elegantly put it “The most profound technologies are those that disappear. They weave themselves into the fabric of our everyday life” [7]. Thus it is not enough that small devices can be manufactured nor that great applications exist. For the aim to be achieved the technology needs to be gradually introduced into our everyday life so that people get the opportunity to learn and experience it. First then will it fade into the background and be ubiquitous.

2. MOBILE DEVICES
But before people are able to start using ubiquitous computing technology it must first of all be available to them. This may appear to be a trivial step but it is a step that needs to be taken nonetheless. Ubiquitous computing artefacts should be possible to obtain in the same manor as any other consumer product. It cannot be expected that the general public will search specialist forums nor to go through any other complicated procedure to obtain the technology. The technology must also be provided in a form people understand and can use, otherwise it does not matter if it is readily available. Thus the technology needs to be self-explanatory and the interaction so natural that it can occur without instructions.

Figure 1. Mobile devices

How then can we make ubiquitous computing available to the public? Well, even though most people may not think about it, they interact with inch-scale devices on a daily basis. The use of devices such as mobile phones and Personal Digital Assistants (PDA) is today so widespread in society that they are becoming pervasive. Increasingly these devices also come with the capabilities required for them to be suitable as a platform for ubiquitous computing, something several projects have shown [8][9].

Our position is therefore that off-the-shelf mobile devices should be used whenever possible. This is also the approach we are taking in our efforts to bring ubiquitous computing to a wider audience. The main benefit as we see it is that this will enable ubiquitous computing artefacts to be made available at a much larger scale. Mobile phones and PDAs are readily available to consumers and there is also a growing market from which software can be obtained. Another benefit is that the public already has come into contact with these devices. This will hopefully make it easier for users to start using the artefact produced and to let them see beyond the technology itself.

3. THE BLUE REMINDER
In an attempt to introduce ubiquitous computing to a wider audience we are currently undertaking a project which uses today’s pervasive technology to tackle the problem of remembering things.

It is no secret that we all tend to forget things from time to time. To minimise our lapses in memory we can use various tools in our daily lives. The post-it note is one such tool. We may for example stick a post-it note on a book with the date when it is due back at the library. The note will then be able to prompt us, when seen, that there is something we need to remember and what it is. Although this traditional technique works well, it is not suitable for all situations. For example we may want to discuss something in person with a colleague the next time we see them. In this case we would like to be reminded only when the colleague is around. The use of an ordinary post-it note in this situation is far from ideal. The best we can do is to attach the note to something we carry with us and hope that we see it at the right time. If we forget to look at our notes, which is bound to happen occasionally, then we may miss those times when we really need to get reminded.

Figure 2. Post-it notes

Is there then an alternative? We believe there is. By using virtual post-it notes this problem can be overcome. Virtual notes are far more flexible than the traditional paper notes we otherwise use. They can be attached invisibly to anything with a unique identifier. This can include everything from people to places and things [10]. The idea is that when a person thinks they need to be reminded about something they will write a virtual post-it note and attach it to a suitable identifier. In the previous example this would be an identifier representing a colleague. Then the next time the presence of the colleague is detected the virtual note can be displayed to remind the user. The use of virtual notes also has the advantage that they allow other factors to be considered as well. For example they can be set to only appear on certain days such as birthdays or anniversaries, when a certain group of people meets, at a particular time of the day, etc.

The BlueReminder is a system being developed on this concept. Through the use of mobile phones we provide users with the ability to both write and read virtual post-it notes. In the system notes are written with a subject, a message, and a triggering condition. This condition can be the presence of one, a group, or one in a group of identifiers at any or a specific date. In addition to this the user has the ability to customise options such as the validity period and the notification action. The system will then at predefined intervals check if any of the written notes should be triggered. If a match occurs the user is notified and can view their note to be reminded.

4. Architecture

To realise the project we have chosen to use standard off-the-shelf mobile phones. They provide us with a uniform platform for which the application can be written as well as the necessary capabilities to detect nearby identifiers. The former comes in the shape of the Java 2 Platform, Micro Edition (J2ME) [11]. Whilst not the most efficient platform, it does provide the widest possible support. The J2ME technologies used include the Compact Limited Device Configuration (CLDC) 1.0 [12] and the Mobile Information Device Profile (MIDP) 2.0 [13]. The latter, i.e. detecting nearby identifiers, is achieved through the use of Bluetooth technology. First of all we assume that people etc. are identified by Bluetooth tags, e.g. mobile phones. Then this assumption is used to allow their presence to be detected using the Bluetooth Service Discovery Protocol. Hence it should be noted that no data is transmitted between devices with the BlueReminder application. The only data used are visible Bluetooth identifiers. This ability to use the mobile phone's Bluetooth functionality is provided by the JSR-82 API [14], which is implemented in several currently available devices.

Device discovery

The JSR-82 API provides a DiscoveryAgent class that can be used to create a search agent object. By using this object, an inquiry for remote Bluetooth devices can be initiated on the local device. The agent is tied to a listener object and reports events such as the discovery of a device and the completion of a search session back to it. The listener object in turn implements methods for handling each of these events.

In the BlueReminder, the search functionality is located in a separate thread that periodically calls the start method of the search agent, waits for it to notify the completion of the search, and finally sleeps for a set period of time. The time taken by an inquiry varies with the number of surrounding devices and the distance to them. Usually it takes about 15 second though. After each search, the list of discovered devices is then matched against the contacts of each available note. If a match occurs the an alert box is displayed together with an optional notification. The notification can be a combination of a vocal signal, a vibration, or flashing lights. On some devices where the screen is not available to an application at all times, for example when the lid on a foldable phone is closed, the BlueReminder periodically request access to the screen once a note has been triggered. This ensures that the alert box is displayed once the screen is made available.

Searches can also be initiated manually to add new contacts, the results are then viewed as a list on the screen. The devices presented are either marked as known or unknown depending on whether they are used in the contacts list or not. By clicking on a device identifier in the list it can be added as a new contact, or to an already existing contact. Thus each contact can have more than one device id. This is important as it is expected that people may carry more than one Bluetooth enabled device or switch between several.

Data storage

The persistent data storage available on the J2ME platform differs from that on traditional computing platforms. While the most obvious difference might be its limited size, the important difference is that applications do not enjoy direct access to the file system. Instead the data storage functionality is provided in the form of record stores. An application can have several record stores, each identified by a string value. These record stores then contain data in the form of records, each identified by an integer value.

The types of data that need to be stored by the BlueReminder are notes, contacts, and settings. For this three separate record stores are used. To simplify the handling of data, a common mechanism has been created that exploits the similarities of the different record stores. The mechanism consists of a collection of general storage classes and interfaces. For each type of record store the methods needed to convert the relevant objects into byte streams has then been implemented. This is required since a record holds data in the form of byte arrays.

The BlueReminder also implements non-persistent storage structures to be used while the application is running. One such structure is the device buffer. The buffer is used to store an adjustable number of devices found in past searches using a first in, first out buffering scheme. This is an important functionality as it supplies the users with a way of adding discovered devices to the contacts list at a time they find convenient rather than upon discovery.

Graphical user interface

Two levels of GUI design are offered by MIDP2.0. A lower level API makes it possible for a developer to assume complete control over the appearance of an application. This is however a very timeconsuming way of developing, thus J2ME also provides a higher level API. At the higher level, classes implementing common objects such as lists, buttons and textboxes are provided. For the BlueReminder the higher level API was considered more appropriate since the application only utilises standard GUI components. This also has the added advantage that the layout of the application's interface is optimised for the devices on which it is run, lowering the learning curve for new users.

Each screen of the application has been implemented as its own class. Whilst not the most efficient design it does make the code highly readable and easier to extend. The GUI receives input from the user through the device keyboard or through the touch screen if this feature is supported. The input is directed to the GUI component currently in focus, such as a text field. Each screen class implements a command listener interface. The method of this interface is called when the user submits a command through clicking a button, selecting a menu alternative or clicking an item in a list.

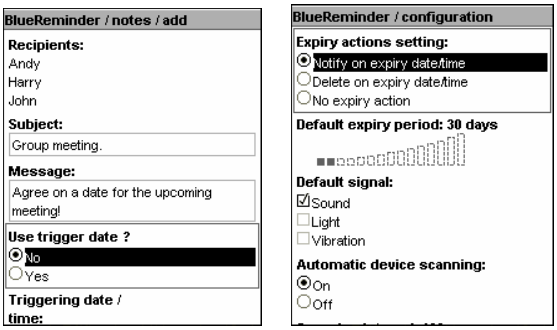

Figure 3. BlueReminder screenshots

5. Preliminary findings

The application has been tested on two popular types of mobile phones from two well known manufacturers. The first phone is a high-end device with a large touch-screen while the second phone is a medium-range device with a small screen supporting output only. The former runs the Symbian operating system and the latter run a vendor specific. The aim of these initial tests has been to verify the concept as well as the conditions for wide scale deployment.

Concept

Overall the concept of the BlueReminder has proved to work well. Notes can be written and reminders issued in everyday situations. This certainly extends the usefulness of these phones. Even though the appearance of the application differs due to the different nature of the devices, the application has been found to be easy to use on both phones. The use of the higher level API and its device specific implementation certainly plays a part in this. This is an important result to enable wide scale deployment as otherwise the effort needed to port the application to each device would be significant.

Energy consumption

We have also investigated how the BlueReminder affect the energy consumption of the mobile device. This is important as the energy consumption need to be kept to a minimum as not to influence the users' desire and ability to run the application. The approach used has been to leave a number of mobile phones running for eight consecutive hours in three different states:

- State 1: Phone in standby mode, Bluetooth switched off
- State 2: Phone in standby mode, Bluetooth switched on.
- State 3: Phone in standby mode, Bluetooth switched on, BlueReminder running

At the start of the test all the mobile phones were fully charged and the power level recorded. Then at the end of the test the power level was checked and recorded again for each phone. By comparing the results between the different states for a given model an indication of the energy consumption has been found. The preliminary results show that running the BlueReminder consumes a significant amount of battery power. This was expected as the regular device discovery operations actively use the Bluetooth radio. Initial figures indicate that state 3 consume twice the energy as state 2. However the results also show that whilst the energy consumption is significant it does not drain the batteries to the degree as to hinder everyday operation. After the BlueReminder has run for 8 hours (state 3) there is still approximately 90-95% of a full battery left. These figures are very much preliminary, further tests need to be performed to verify them.

Stability

Finally tests have been performed to examine the stability of the BlueReminder. Stability is here defined a measure of the predictable behaviour the application exhibits when run over long periods of time. For the benefits of the BlueReminder to be gained it needs to run continuously in the background of the device, routinely performing a device discover. If the application is terminated, the user runs the risk of not being reminded. Good stability is therefore a primary design goal. Throughout the design process the minimum period of time the application must be stable was set to 24 hours, meaning that the application must at the very least remain stable for this period under normal conditions in order to be practical.

During our testing stability problems have been discovered, on the more powerful of the two test platforms. After running the application for several hours, sometimes several days, the Bluetooth service of the device unexpectedly stalls. This prevents the BlueReminder from performing anymore searches. Whilst this abnormal state can be detected by measuring the time an inquiry takes, the J2ME platform does not provide any mechanism to address this issue. The only way to recover from the abnormal state is by manually restarting the Bluetooth service on the device through the control panel of the phone in question. Thus the application has been adapted to alert the users if the Bluetooth service stalls. This is not optimal though.

Two interesting observations have been made regarding the probability of a lockup occurring. First, the probability for the Bluetooth to stall is higher on some devices than others, even if they are of the same model, use the same firmware, and run identical versions of the application. This leads us to believe that there are other factors involved. We can only speculate what these factors are but possible causes include different patterns of device usage or variations in the quality of the hardware. Secondly, the time before a device enters into an abnormal state becomes significantly shorter as searches are performed more frequently. A sleep period between searches of at least 180 seconds was found to reduce the probability of service lockups significantly on most of the devices tested.

To establish where the problem lays, a simplified test application has been developed in Personal Java that performs an equivalent device search. This implementation has them been subjected to the same type of stability tests as the J2ME implementation. The results found were the same. After different lengths of time the Bluetooth service stalls on the mobile devices. This indicates that the problem is neither in JSR-82 API nor in the J2ME implementation of the BlueReminder. The fact that we have so far only observed this problem on one of the platforms further strengthens our belief that it stems from either the low-level operating system interface towards the hardware or the actual hardware itself.

On the high-end devices infrequent virtual machine errors have also been encountered. By careful optimisation of the application’s memory usage, these problems have been minimised. On the second, medium-end, devices no stability problems occurred per say but the application times-out after being inactive for a longer period of time. The reason for this is that the device is not capable of running the application in the background. This has obviously limited the ability to perform long running test on this device. However the tests performed have not given any indications of stability issues being present.

Overall the stability tests suggest that there is a need to look at the underlying operating system and the Java Virtual Machine. However the fact that one of our two test platforms did not present any stability problems indicates that BlueReminder itself is stable. Also the relatively low frequency of problems suggests that it is possible to use the application on a day to day basis. Finally upcoming mobiles are likely to improve upon stability as both the JVM and the Bluetooth services matures.

6. FURTHER WORK

Our findings have identified several areas for further work.

The current test results need to be further verified. This includes running more test on a wider range of devices. Whilst the results are not expected to differ, it will allow them to be quantified. In particular it would be useful to gather statistical information on how long the application is stable as well as more exact measure of the energy consumption.

There are a number of additional functions, minor changes and optimisations to be made before a large scale deployment can be made. Currently the BlueReminder lacks the ability to import/export contacts and notes. This functionality is needed for corporate use. Then there is the issue of the stability. This issue need to be further investigated, ideally in close cooperation with device manufacturers, to see if the situation can be improved upon. Finally there is the energy consumption for which the current implementation has not been optimised. Here improvements can be made by for example: only searching when there are notes, increasing the interval between searches, disabling the search at certain times, etc.

It is also desirable to run a case study with the target audience, the public, to study their use of the technology. Interesting topics include their interaction with the devices, how they incorporate the new tool in their daily lives, how they consider their privacy being affected, etc.

7. RELATED WORK

The idea of an virtual post-it note is not new [15][16]. The stick-e notes concept allowed electronic notes to be attached to locations. These notes would then trigger events upon being detected, which in general involved displaying note information to the user. The stick-e note concept has also been proven to work in a fieldwork environment. This is a good indicator that verifies that the concept indeed can be used in real situations. The feasibility of using bluetooth on mobile devices to mediate contextual information has also been shown [17]. MobiTip for example uses bluetooth to communicate people's comments on things in the environment from user to user. Collectively these comments then form the basis for tips that can be given to users.

8. CONCLUSION

Introducing ubiquitous computing to the general public is not an easy task. Making it a natural part of everyday life is even more difficult. However a first step towards achieving this goal will be to make the technology available in a form people understand and can use. Our position is that the use of off-the-shelf mobile devices can aid us in taking this step. Our ongoing project, the BlueReminder, uses today's mobile phones to provide a pervasive service that allows virtual post-it notes to remind users. What makes this project different from previous efforts is that the focus is on aiding people's interaction *with each other*. Furthermore the project is undertaken with the explicit intention of *reaching a wider audience*. Our results so far highlight some issues that need to be investigated further, in particular with respect to stability over long periods of time. However despite this the overall indication is that a large scale deployment of ubiquitous computing technology is indeed within reach today.

9. ACKNOWLEDGEMENTS

Many thanks to the joint Jayway/Leadway project team for their effort in realising the BlueReminder.

10. REFERENCES

[1] Weiser, M., R. Gold, et al. (1999). "The origins of ubiquitous computing research at PARC in the late 1980s." IBM Systems Journal 38(4): 693-696.

[2] Harter, A., A. Hopper, et al. (2002). "The Anatomy of Context-Aware Applications." Wireless Networks 8(2-3)

[3] Schilit, B. N., D. M. Hilbert, et al. (2002). "Context-Aware Communication." IEEE Wireless Communication 9(5)

[4] Osbakk, P. and N. Ryan (2004). The development of a privacy-enhancing infrastructure: Some interesting findings. Ubicomp Privacy: Current Status and Future Directions Workshop, UbiComp 2004, Nottingham, UK.

[5] Ryan, N., J. Pascoe, et al. (1999). FieldNote: a handheld information system for the field. TeleGeo'99, 1st Int. Workshop on TeleGeoProcessing, Lyon, France.

[6] Consolvo, S., P. Roessler, et al. (2004). The CareNet Display: Lessons Learned from an In Home Evaluation of an Ambient Display. UbiComp 2004, Nottingham, UK.

[7] Weiser, M. (2002). "The Computer for the 21st Century." Pervasive Computing 1(1): 19-25.

[8] Rukzio, E., A. Schmidt, et al. (2004). An Analysis of the Usage of Mobile Phones for Personalized Interactions with Ubiquitous Public Displays. Workshop Ubiquitous Display Environments, UbiComp 2004, Nottingham, UK.

[9] Mäntyjärvi, J. and T. Seppänen (2003). "Adapting applications in mobile terminals using fuzzy context information." Interacting with Computers 15(4): 521-538.

[10] Kindberg, T., J. Barton, et al. (2002). "People, Places, Things: Web Presence for the Real World." Mobile Networks and Applications 7(5): 365-376.

[11] Java 2 Platform, Micro Edition. Last Accessed 26/04/2005. *http://java.sun.com/j2me/*

[12] JSR-000030 J2ME Connected, Limited Device Configuration. Last Accessed 26/04.2005. *http://jcp.org/aboutJava/communityprocess/final/jsr030/index.html*

[13] JSR-000118 Mobile Information Device Profile 2.0. Last Accessed 26/04/2005. *http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html*

[14] JSR-000082 JavaTM APIs for Bluetooth. Last Accessed 26/04/2005. *http://jcp.org/aboutJava/communityprocess/final/jsr082/index.html*

[15] Brown, P. J., J. D. Bovey, et al. (1997). "Context-aware applications: from the laboratory to the marketplace." IEEE Personal Communications 4(5).

[16] Pascoe, J., D. Morse, et al. (1998). "Developing Personal Technology for the Field." Personal Technologies 2(1)

[17] Rudström, Å., M. Svensson, et al. (2004). Using Bluetooth as a Mediator of Social Context. UbiComp 2004 Adjunct Proceedings (demo), Nottingham, UK.