Design, Prototyping and Evaluation of Ambient Media: Lessons Learned from the Ambient Notifier

Denzil Ferreira, Vassilis Kostakos, Jakob Rogstadius, Jayant Venkatanathan
Madeira Interactive Technology Institute
University of Madeira
lizned.arierref@gmail.com, {vk, jakob, vjayant}@m-iti.org

ABSTRACT
This paper presents the design, prototyping and evaluation of the Ambient Notifier, a standalone ambient display which passively notifies users of incoming calls, missed calls and received SMSs awaiting on their mobile phone. The paper provides an overview of the system, and presents the insights obtained during its evaluation. The discussion provides a reflection on the tools that were used to build the system (including the Arduino and Android platforms), their shortcomings, and suggests how these tools can be improved to assist in the design and development of mobile and pervasive systems.

Author Keywords
Ambient media, ubiquitous computing, calm technology, peripheral awareness, ambient display, tools.

INTRODUCTION
Mobile phones are becoming an increasingly ubiquitous technology [6] due to their widespread penetration in the public, their growing functionality, and the wide spectrum of activities they support and facilitate [3,4]. Despite their mobility, empirical evidence on mobile phone practices shows that only 58% of users keep their phone within arm's reach, while 20% keep phones a bit farther away but still in the same room [7]. This evidence suggests that users need to periodically interact with their mobile device to check if there was a missed call or SMS. This can be time-consuming, especially since the phone may be in a different room, and can introduce interruptions to users' activities.

The Ambient Notifier is a standalone device designed to reduce the frequency of explicit interaction between users and their mobile devices which are due to users wishing to remain updated about the status of their phone. This paper presents an overview of the Ambient Notifier, and briefly discusses the findings from its evaluation. The discussion provides a reflection on the tools that were used to build the system (including the Arduino and Android platforms), their shortcomings, and suggests how these tools can be improved to assist in the design and development of mobile and pervasive systems.

MobileHCI 2010 September 7-10, 2010, Lisboa, Portugal.
ACM 978-1-60558-835-3/10/09.

RELATED WORK
Designing ambient displays is challenging. They need to provide adequate feedforward and feedback such that users can assess what they can do with them, what information the ambient display shows, and how well the display fades into the background. Holmquist [2] argues that ambient displays have three levels of comprehension: users understanding that information is being visualized; what the information is; and how it is being displayed. Ambient displays aim to provide users with relevant information, in the right form, at the time and place it is needed [3]. One crucial factor in their success is comprehension: how well do users understand the purpose of an ambient display and how to use it.

To facilitate comprehension, the use of non-computer artifacts in everyday surroundings have been explored as potential ambient displays due to users' familiarity with them. In one such study [5] a framework was developed that allows the mapping of information to everyday objects to effectively specify how the information is consumed. The proposed mapping represents information as changes in physical state and appearance. Empirical evidence from using ambient media in the office suggests that everyday objects can be successful ambient displays, as they improve accessibility and help promote awareness of information [5]. The shortcomings in the above framework is that it requires the user to explicitly create the mapping between the object and the information they wish to display. Additionally, it is not trivial to identify the best way to map various types of information such as scalars or continuous values. This mapping is especially important depending on how ambient displays are used.

An important distinction in the use of ambient displays is that their mode of use can be implicit or explicit [3]. Explicit interactions are actions performed on the user's initiative, while implicit interactions are automatic and possibly without direct user intervention. This suggests that ambient displays receive varying, and possibly unpredictable levels of attention. To deal with this ambiguity, ambient display devices use sound, light, vibration, color or movement to get users' attention, interact with the user and actually display information. It has been suggested that people prefer aesthetically designed objects as they need to mingle among other objects in a home or office table. For example, the Breakaway ambient display used an animated sculpture to remind people to take breaks after a long period sitting [1]. The authors further

suggest that ambient displays must function as decorative, inquisitive objects when not in use. The goal is to design for non-intrusiveness and aesthetics.

The Ambient Notifier, described next, uses light to get users' attention and present information about incoming calls, missed calls, and received SMSs. Publicly exposing the state of a user's mobile phone is a challenging task, especially since users do not tend to reveal personal information contained in the phone to others. However, previous work has explored mobile phones in the context of ambient displays and publicly revealing information [4], arguing that interaction with the phone and through the phone reveals and reflects the user's behavior.

AMBIENT NOTIFIER

The Ambient Notifier is a standalone, wireless peripheral display that notifies users of incoming calls, missed calls, and received SMSs. While the initial prototype relied on motion to grab user's attention, the final design uses light and sound since they have a higher peripheral impact than movement [2]. An additional requirement for the Ambient Notifier was that it should be mobile so that users can easily place it in their environment.

Hardware

The main component of the system is an RGB Led (Figure 1). This is a light emitting diode that can change between red, green, blue and any combination of the three colors. When all three colors are activated the LED turns white. The LED is connected to an Arduino Duemilanove via a 220 Ohm resistor which was isolated using heat shrinkable tubbing. The Arduino board was extended with a WiShield component to provide internet access, and was attached to a battery for power. Finally, using a laser cutter an acrylic casing was built to house the electronic components. The LED shines through the acrylic case onto the lamp, thereby making the lamp light up.

Operation

To enable the Ambient Notifier to respond to events on the phone, a server was developed to run over the WiShield's TCP/IP stack. The server listens for pre-defined messages sent from the mobile phone (or indeed from any internet-capable device): *off*, *ringing*, *answered*, *dismiss* and *sms*. A small software service was developed for the Android platform that takes care of detecting different events on the phone and sending the appropriate signals to the Ambient Notifier. The state machine design pattern was used to map the LED's color to states, and to model the transitions between states (Figure 2).

By default the Ambient Notifier is in the Idle state. When a SMS is received on the user's phone, the Ambient Notifier changes from Idle to SMS state (blue). An incoming call changes the state to Ringing (green). If the user does not answer the phone, Ambient Notifier changes to Missed call state (red). These states and transitions were informed by the Peripheral Display Toolkit [8], a toolkit that empowers designers of ambient displays to manage users' attention.

Figure 1. And RGB LED connected to an Arduino Duemilanove via a 220 Ohm resistor forms the basis of the Ambient Notifier.

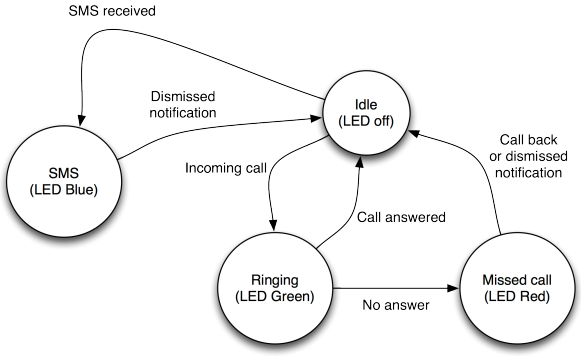

Figure 2. A state diagram showing the transitions between red, green and blue based on phone status.

Figure 3. The Arduino board is installed in an acrylic case underneath a lamp. The LED light shines through the case and lights up the lamp overhead.

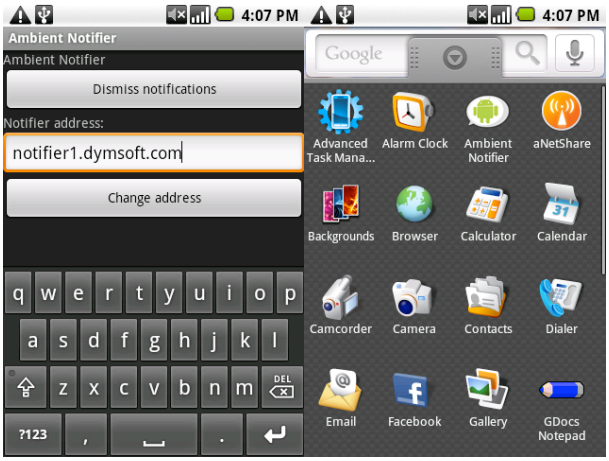

Figure 4. The user interface of software that connects mobile phones to an Ambient Notifier.

The toolkit provides support for three aspects of ambient displays: abstraction, notification levels and transitions. Abstraction involves simplifying the information to be displayed, so that it is easy to interpret at a glance. The Ambient Notifier simplifies information by changing phone events into colors. Notification levels refer to differences in information importance, which in the Ambient Notifier map between the no light and light conditions. Transitions are how users' attention is captured. In the Ambient Notifier, the RGB LED toggles between colors as events occur.

Mobile phone

The Ambient Notifier requires a background process to run the users' mobile phones. A background service was developed for the Android platform, which enables the user to specify the networking address of the Ambient Notifier (Figure 4). In addition, the application indicates to the user whether the phone is actively connected to an Ambient Notifier. This software monitors the phone activity in the background, and detects events of interest. Specifically, the application monitors for incoming calls, missing calls and received SMS.

EVALUATION

Three evaluation studies were conducted. Study 1 was a heuristic evaluation adapted for ambient displays from Nielsen's heuristics [9]. Study 2 was an in-situ informal evaluation carried out with the Ambient Notifier, involving ten users who had the opportunity to interact with the system, see it in action, and provide feedback on its operation. Study 3 was a long-term evaluation involving two participants using the system for 3 weeks. In this study the participants were interviewed both before and after using the system, and the objective was to assess the extent to which the system affected the way users handle and interact with their phone.

Results

The heuristic analysis used to evaluate the Ambient Notifier is an adaptation of Nielsen's heuristics, and is reported in [9]. From this technique, the following heuristics proved to be useful during evaluation of the system:

- *Useful and relevant information* was a focus of the system by displaying phone status to users;
- *Peripherality of display* highlighted the importance of the Ambient Notifier waiting to be engaged and be non intrusive;
- *Match between design of ambient display and environments* was achieved by the Ambient Notifier's portability and small size, but in its present form is far from being an object fading into the environment;
- *Sufficient information design* indicates that the features level should be kept at an "enough" level, as was the case with the Ambient Notifier;
- *Consistent and intuitive mapping* was achieved by mapping the colors to the events consistently;
- *Easy transition to more in-depth information* was taken into consideration, as the Ambient Notifier triggers the user to use the phone in case they need to know who called, who sent an SMS and what was written;
- *Visibility of state* was achieved since users can perceive that something is going on with their phone.
- *Aesthetics and Pleasing Design* are a useful direction for further improving the Ambient Notifier.

In Study 2 ten users had the opportunity to use the Ambient Notifier and play with it during an afternoon. Their feedback suggests that it was immediately apparent to them what the notifier did and how it worked. This was reinforced by the fact that participants were given a live demonstration of the Ambient Notifier being right next to a mobile phone. This setup made it straight-forward to map events on the mobile phone with the states of the Ambient Notifier, since the two were right next to each other and participants were able to observe both the phone and the Ambient Notifier simultaneously.

Study 3, the longitudinal evaluation of the system with 2 users, revealed a number of insights. Both participants commented that although it was pleasant to have an ambient notion of the events happening on their phone, they found it annoying that they needed to interact with the phone itself in order to "reset" the Ambient Notifier. The participants commented that it would be useful to be able to "shake" or "touch" the Ambient Notifier in order to acknowledge and dismiss the notification.

One of the participants mentioned that the Ambient Notifier was quite helpful to know when the phone rang, especially when the phone was not nearby or the environment was noisy.

"...I like it, it's like having a lamp that controls my phone activities..."

"...I would leave the device near me on the desk, while I had the phone charging... it would light up the desk every time someone was trying to reach me..."

Finally, the participants commented that the battery life of the Ambient Notifier (1 day) was not sufficient, as they had to recharge the battery every night. A possible improvement

on the design of the Ambient Notifier is to enable it to recharge itself by plugging it to a wall socket, much like how mobile phones are recharged. This would avoid the inconvenience of removing the battery before every recharge cycle.

DESIGN, PROTOTYPING AND EVALUATION OF AMBIENT MEDIA: LESSONS LEARNED

The Ambient Notifier project is an example of the full cycle of design-implementation-evaluation commonplace in mobile and ubiquitous research, and touches upon a number of important issues for designers and developers. Most notably, this paper argues that modularity is an system characteristic that effects hardware, energy, and software, and is likely to play a key role in the proliferation of mobile and ubiquitous systems of the future.

Hardware

The hardware research community has been very successful in quickly bringing advances from the lab to the market. The experience of the Ambient Notifier suggests that there exists a considerable variety and richness of hardware equipment, including sensors, actuators, and increasingly fabric-like components. The hardware is cost-effective as it is usually mass produced, robust and well-documented, but typically lacks the software and tool support that would entice software engineers to "get their hands dirty".

One decision that had to be made in relation to the Ambient Notifier's hardware was whether to opt for a modular construction or an all-in-one solution. The reason the Arduino board was chosen was its modularity. This allowed the design to use only the bare minimum amount of electronics needed, and as such improved battery life of the Ambient Notifier considerably. Hardware modularity allows for a pick-and-pix flexibility that can result in highly customizable hardware solutions. On the other had, all-in-one solutions (such as small form-factor PCs) provide more flexibility in terms of software but usually result in energy inefficiency.

It is also important to note that laser cutters, and other types of DIY systems (such as 3D printers) are fast becoming affordable. As in the case of the Ambient Notifier, these systems allow for even further customization of the look of the final system. This approach is an important new way to think about how the mobile and ubiquitous systems of the future will be built: via modular hardware and DIY equipment that ensure high levels of customization, as opposed to all-in-one ready to use kits.

Energy

Despite significant advances in hardware, energy remains an important issue for mobile and ubiquitous systems. Energy efficiency is crucial as it can drastically effect the system's autonomy, and its ability to keep up with users' daily routine. Experience with the Ambient Notifier suggests that system autonomy less than 12 hours typically results in breakdowns with users' daily routine. Autonomy above this threshold can be quite useful for users, since they can cope by re-charging the system during the night when they are most likely to not need to use the system. Energy efficiency is thus crucial, and can be improved both by hardware and software improvements.

For example, the Ambient Notifier's wireless connectivity had the most impact on its energy consumption. The WiShield consumes about 230mA when transmitting data, 85mA while receiving data and about 10mA while on standby. Using a 9V battery the system lasted for a full day. Such energy requirements provide further support for a modular approach to building system with high autonomy, if the battery is actually going to be smaller than the hardware itself.

Software

While like hardware there exists a plethora of software tools for building systems, the interaction and compatibility between various software tools is not as ironed out as in the case of hardware. In the case of the Ambient Notifier a variety of software tools and platforms had to be used together, both for the lamp as well as the mobile phone.

Programming the Arduino was done with a language based on C/C++ and the Arduino IDE which is available for Linux, Mac and Windows operating systems. Setting up the environment to work with the WiShield requires the use of firmware and the drivers for the wireless board. The provided samples allow for rapid interaction with the WiShield and the Arduino board. The Arduino IDE is based on JAVA, being cross-platform compatible and features syntax highlighting, bracket and brace matching and automatic indentation. Still, it lacks some features found in more recent programming IDEs, such as code suggestions, completion and syntax checking. Error checking was only performed when compiling, accessible from the menu, which leads to frequent compiling requests to check for errors. Compiling the code requires the use of Arduino IDE, but Eclipse could be used to develop code in C/C++ [11]. More recently, Netbeans IDE can be used to develop applications for the Arduino [10] but no integration with the serial output of the Arduino.

Eclipse is the choice for developing Android applications, mostly due to Google's support with a plugin called ADT (Android Development Tools), together with Android's SDK. Testing can be done using the Android emulator or with a real device, allowing for development even without an Android device. Eclipse advanced programming features such as refactoring, syntax highlighting and error checking, code hints and autocomplete minimizes development effort for Android devices.

Networking

While networking has matured considerably due to the proliferation of desktop computers and mobile phones, it still remains a pending issue for mobile and ubiquitous applications. Even though connectivity may be taken for granted, tools for achieving useful networking on top of the available connectivity are far from perfect. For instance,

while the Ambient Notifier was fully connected to the internet, it did not have the ability to use DNS, and rather relied on direct IP addressing. While this was a limitation due to the hardware drivers, it is a good example of connectivity crippled by the lack of useful tools.

For a successful modular, end-user development approach to mobile and ubiquitous systems, the existence of appropriate networking tools is a necessity. The increasing variety of connectivity technology ranging from short range NFC and RFID to Bluetooth, WiFi and ZigBee makes the need for networking tools and developer support ever more crucial.

Evaluating mobile and ubiquitous systems

Evaluating systems such as the Ambient Notifier is a challenge because lab settings can be unrealistic, and short-term sessions may not reveal the full range of feedback that users may possibly have to give. In designing Ambient Notifier a number of decisions had to be evaluated, including how to display information that is relevant for users; how much do users need to know; does that information reveal private information to others; does the device blend into the background or does it stand out as different.

The use of tools such as the Arduino and Eclipse for building rough prototypes to test the functionality of a ubiquitous device has its benefits. It allows for quick adaptation and changes to the prototype according to user test feedback and allows developers to test what potential usability problems exist at an early in the design stage. Having participants use the prototype revealed much more than designers alone would have realized.

Even though the Ambient Notifier was such a prototype, once it was taken back to the lab the users indicates missing the device afterwards. This suggests that the available tools (both software and hardware) for quick prototypes can produce results that are very close to an actual product.

End-user customization

The modular approach advocated in this discussion plays an increasingly important role in end-user customization. Some of the feedback received from users was that the software could be customized to have a number of new features, such as new types of notification, or notifications for calendar and email events. It is interesting to note that when users talked about software they referred to the software on the phone as opposed to the software on the Ambient Notifier itself. This perception can provide fertile ground for customization via an app-store model which is gaining increasing popularity.

An app-store approach can motivate developers to implement modular systems that can be easily modified or upgraded by users, and further motivates developers to work such improvements by generating income for them. On the other hand, app-stores have proved popular with users since they offer an easy way of extending the functionality of their devices and systems.

CONCLUSION

This paper presented the development and evaluation of the Ambient Notifier, a stand-alone lamp-like device that uses color to indicate to users the activity on their mobile phone. This device minimizes the amount of explicit interaction between users and devices as it allows users to get updated on the status of their phone with a single glimpse.

The discussion of this paper focuses on the design and implementation issues that the Ambient Notifier project revealed. It argued that modularity is important in many respects, and should be a guiding strategy for mobile and ubiquitous systems as well as tools for building such systems.

ACKNOWLEDGEMENTS

The authors thank all participants. This work is supported by the Portuguese Foundation for Science and Technology (FCT) grants CMU-PT/SE/0028/2008 (Web Security and Privacy) and CMU-PT/HUMACH/004/2008 (SINAIS), and by an IBM Faculty Award.

REFERENCES

1. Jafarinaimi, N.; Forlizzi, J.; Hurst A.; Zimmerman, J.: Breakaway: An Ambient Display Designed to Change Human Behavior. *CHI 2005, April 2-7, Portland, Oregon, USA*
2. Holmquist, L.: Evaluating the Comprehension of Ambient Displays. *CHI 2004*.
3. Becta - British Educational Communications and Technology Agency.: Emerging Technologies for Learning. *Volume 2, 2007*
4. Schmidt, A.; Häkkilä, J.; Atterer, R.; Rukzio, E.; Holleis, P.: Utilizing Mobile Phones as Ambient Information Displays. *CHI 2006, April 22-27, Montréal, Québec, Canada*.
5. Gellersen Hans-W.; Schimidt, A.; Beigl, M.: Ambient Media for Peripheral Information Display. *Personal Technologies, 1999, pp. 199-208*.
6. Weiser, M.: The computer for the 21st century. *Scientific American, 1991, pp. 94-104*.
7. Patel, S.; Keintz, J.; Hayes, G.; Bhat S.; Abowd G.: Farther Than You May Think: An Empirical Investigation of the Proximity of Users to Their Mobile Phones. *Ubicomp 2006, 2006, LNCS 4206, pp. 123-140*
8. Matthews, T.; Dey, A. K.; Mankoff, J.; Carter, S.; Rattenbury, T.: A Toolkit for Managing User Attention in Peripheral Displays. *UIST '04 pp. 247-256*.
9. Mankoff, J.; Dey, A. K.; Hsieh, G.; Kientz, J.; Lederer, S.; Ames, M.: Heuristic Evaluation of Ambient Displays. *CHI 2003 - Proceedings of the CHI 2003 ACM Conference on Human Factors in Computing Systems, September, 2002*.
10. Arduino Development using Netbeans IDE: *http://java.dzone.com/news/arduino-development-using, 21-05-2010*
11. Arduino Development using Eclipse IDE: *http://www.arduino.cc/playground/Code/Eclipse*, 26-04-2010