

Visualisation of Personal Communication Patterns Using Mobile Phones

Bradley van Tonder and Janet Wesson

Department of Computer Science and Information Systems,
Nelson Mandela Metropolitan University, P.O. Box 77000,
Port Elizabeth, South Africa, 6031
{Bradley.vanTonder, Janet.Wesson}@nmmu.ac.za

Abstract. Ambient displays are attractive, subtle visualisations of information. They are typically situated on the periphery of human perception, requiring minimal effort to be understood. The vast volume of communication facilitated by modern means of communication has led to research into methods of visualising this information to facilitate rapid understanding. These two research areas have, however, seldom been combined to include the use of ambient displays as visualisations of personal communication patterns. The research outlined in this paper addresses this issue by combining ambient displays and visualisation of personal communication patterns in a mobile context. This paper details the development of the *AmbiMate* system, analyses its usefulness and investigates the lessons which can be learned from its implementation in order to guide the future development of such systems.

Keywords: Ambient displays, visualisation, personal communication patterns, mobile devices.

1 Introduction

Mobile phones have developed rapidly from the primitive devices of previous decades to the advanced communication platforms they are today. Today, many people view mobile phones as their primary communication device [1]. The volume of communication taking place is often overwhelming, and users can often lose track of their own personal communication habits. One answer to this problem can be found in ambient displays [2].

Mark Weiser introduced the notion of ubiquitous computing [3], whereby computers and their associated technologies would disappear into the background and require less cognitive effort on our part. Weiser also described the concept of *Calm Technology* [4], where technologies would “*empower our periphery*”. Such technologies would be less intrusive, allowing us to focus on the tasks at hand. The user has the power to decide to focus on peripheral information when he or she wants to do so.

Ambient displays provide the perfect medium for communicating information in a manner which is not intrusive. According to Mankoff *et al.* [5], ambient displays are attractive displays of information, which are situated on the periphery of human

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-3-540-92698-6_37](https://doi.org/10.1007/978-3-540-92698-6_37)

attention. Such displays typically make use of metaphorical visualisation techniques, to display information in a form which can easily be understood by their users. Ambient displays have previously been effectively used as a tool for visualising personal communication habits, as can be seen in the visualisation of personal email presented by Redstrom *et al.* [6]. These visualisations allow users to easily appreciate information relating to their personal communications habits, with little concentration or mental effort required.

A natural evolution of such systems is to consider the development of mobile ambient displays to visualise the personal communication data stored on mobile devices. Mobile devices provide a number of unique advantages in terms of privacy and personalisation. Indeed, with mobile phones typically situated on the periphery of their users' attention, they provide a medium well-suited to be utilised as an ambient display.

Recently, the work by Schmidt *et al.* [7], has investigated the use of ambient displays to visualise personal communication on mobile phones. This work was, however, limited to prototyping designs, and no actual systems were (as yet) implemented on mobile devices. In order to establish whether mobile phone-based ambient displays are useful for visualising personal communication patterns, it is necessary that such systems be implemented and evaluated.

This paper outlines the development and evaluation of the *AmbiMate* system, a mobile ambient display used to visualise personal communication patterns. The system was developed in order to investigate its potential usefulness. Practical issues encountered during the development of the system will also be discussed, and could provide valuable insight for the future development of such systems.

2 Related Work

This research integrates a number of different research areas. Research into visualisation, the use of ambient displays and studies of personal communication patterns are combined. This research seeks to establish whether the benefits of using ambient displays to visualise personal communication patterns can be extended to a mobile context.

2.1 Ambient Displays

The origins of ambient displays can be traced back to the birth of Ubiquitous Computing in the early 1990s. Mark Weiser and researchers at Xerox PARC foresaw a future where computers would be fully integrated into every area of our lives [3]. They suggested that in order for computers to integrate successfully into the environment, it would be necessary for them to disappear into the background, no longer requiring our full attention. One of the first devices to build on this idea was the *Dangling String* [8]. This device consisted of a plastic string hanging from the ceiling, which communicated the level of network traffic through its level of motion. It made use of the user's peripheral vision, as well as characteristic sounds, to communicate information to users without them having to devote attention to it. Further examples include the *Water Lamp* [2], which used ripples in water to denote a

variety of digital information, and the *Ambient Orb* [9], a translucent sphere which changed colour to denote stock prices.

Much of the early work into ambient displays focused on the development of physical devices and used a variety of inventive means to communicate information. The limitation of this approach was that the physical nature of these devices limited their usefulness and flexibility as a means of visualising information. Research into the use of electronic ambient displays is addressing this problem, and the full potential of ambient displays as a medium for information visualisation is becoming apparent [10].

A typical feature which distinguishes information visualisation in general, and information visualisation with regard to ambient displays, is the use of metaphors and other abstract representations of information. Ambient displays typically make use of visual metaphors rather than traditional visualisations (e.g. bar graphs or pie charts) to convey information. An excellent example of this is the *InfoCanvas* system [11]. This system visualises a number of different pieces of information through metaphors, all integrated into a single static ambient display. Information which is of personal interest to users is communicated using various elements of the display. For example, the *InfoCanvas* made use of a beach scene metaphor, with elements of the display such as shells on the beach, clouds in the sky and the height of a kite all representing different information of personal interest to the user.

Other systems, such as the *Kandinsky System* [12], put an even stronger focus on the aesthetic aspect of ambient displays. The *Kandinsky System* uses compositions of images to represent information. It takes textual information, relates this information to images, and then combines the images in a collage. A variety of techniques are used to ensure that a collage is created which is aesthetically appealing, but still communicates useful information. Other ambient displays were based on the work of artists, such as the bus schedule and weather visualisations based on the style used by the Dutch artist Piet Mondrian [6].

The *Hello.Wall* system [13] was one of the first systems to use mobile devices as ambient displays. The *Hello.Wall* system was used in conjunction with PDA-like devices called *ViewPorts*. While the *Hello.Wall* was used to publicly display information in an office environment using light patterns, it could also interface with the mobile *ViewPorts* to communicate information privately. The mobile devices could be used to decode private messages left on the *Hello.Wall*.

2.2 Personal Communication Patterns

Several studies have looked at the characteristics of personal communication from different perspectives. Four different approaches were identified as the most widely used in research into email visualisations [14]. Three of these are general enough to characterise personal communication research in general:

- *Temporal visualisations*: studying how communication and relationships with different contacts have changed over time.
- *Contact-based visualisations*: studying communication with various regular contacts and extracting patterns and trends.
- *Social-network visualisations*: understanding social groupings through their personal communication habits.

Email, instant messaging and Internet discussion groups have all been the subject of visualisation research falling into these three categories. Systems such as ContactMap [15] visualise email to allow users to keep track of the different contacts with whom they communicate. Other systems, such as the PeopleGarden system [16], which models online discussion group interactions, provide a more overall view of communication. In the PeopleGarden system, different users in the discussion group are visualised as flowers. The height of the flowers represents time spent logged in, with the each flower visualising the number of messages posted by the corresponding user.

Ambient displays have also been utilised as a medium for visualising personal communications data. Redstrom *et al.* [6] describe an electronic email visualisation that uses an ambient display to visualise email communication. In this ambient display, different rectangles represent different people, with the size of the rectangle reflecting how many emails that person has sent and received. The colour and position of the blocks remain the same, to allow identification of different users. The display was designed to be displayed publicly in an office environment.

Instant messaging has also been the subject of visualisation systems, such as the CrystalChat [17] system, which used strings of circles to represent messages sent to different contacts. Circles are used to represent messages sent on the instant messaging program, MSN Messenger, with colour representing the person sending the message and size visualising message length.

2.3 Mobile Visualisation of Personal Communication Patterns

The work of Schmidt *et al.* [7] provides some of the first research into visualising mobile communication patterns. They proposed a number of prototypes to visualise personal communication data that is typically stored on mobile devices. Information typically available in phone logs includes data relating to calls made and received, who the other party was, call duration, direction of the call and time of day. Their prototypes utilised ambient displays on mobile devices to visualise this information. Various metaphors were suggested, including an aquarium metaphor, a solar system metaphor and a flower metaphor. In the aquarium metaphor, different colour fish represent different contacts, with the size of the fish representing the total volume of communication with that contact (the larger the fish, the more contact). The direction in which the fish are swimming represents the dominant direction of communication with that contact, and the speed of the fish signifies the time elapsed since the last communication with the contact in question (the faster the fish, the shorter the time elapsed).

3 Design

In order to investigate the potential usefulness of mobile ambient displays in visualising communication patterns, it was necessary to implement such a system. The development of this system served two useful purposes. Firstly, it provided a tool to be used in evaluating the usefulness of the system. Secondly, the development process helped to investigate the practical implications of implementing such a system.

In order to select appropriate design metaphors to implement, a pilot study was conducted amongst potential users. Four designs were chosen from related work,

three of which were proposed as prototypes in the work of Schmidt *et al.* [7], and the fourth a variation on a design used successfully by Redstrom *et al.* [6] as an ambient display. This process resulted in the decision to implement two alternative designs, one of which visualised contacts as fish (Aquarium Design) and the other visualising contacts as flowers (Flower Design).

This section outlines the design of the *AmbiMate* system, a mobile phone-based ambient display, which utilises animated metaphors to visualise personal communication data stored on mobile phones.

3.1 Functional Requirements

The *AmbiMate* system not only visualises personal communication patterns through the use of animated metaphors, but also supports a range of customisation options. These include customising the display in terms of display mode (Aquarium or Flower design), contacts to be displayed, data type, time period and update interval. An option also exists to view display details, as well as system help.

3.2 Implementation Tools

The architecture of the *AmbiMate* system was greatly influenced by the choice of implementation tools. In order to develop a 3rd party application with access to the communication data stored on the mobile device, it was necessary to target the so-called “smartphone” category of mobile phones. The Symbian OS is by far the market leader in this area, with approximately 67% of the smartphone market share in 2006 [18], making it the obvious selection as the target OS. This choice leaves two choices of implementation languages: Symbian C++ or Java Micro Edition (JME). Unfortunately, the Connected Limited Device Configuration (CLDC) of Java running on Symbian devices does not allow access to the necessary event log information, due to security considerations. Hence, a purely JME implementation of the system was not possible. However, Symbian C++ was considered inferior to JME for a number of reasons. These included a lack of garbage collection, inferior development tools, and greater complexity, particularly for the animation portion of the system. A purely Symbian C++ implementation would also result in a very limited set of devices on which the system is able to run. JME also provides the added advantage of a powerful, easy to use Scalable Vector Graphics (SVG) API called TinyLine2D [19].

It was therefore decided to implement a system which uses Symbian C++ in order to access the required data, with the animation and user interface implemented in JME. This provides the advantage of efficient access to the required data provided by Symbian C++, combined with the easy-to-use SVG animation power of JME.

3.3 Architecture

The choice of implementation tools resulted in a system consisting of two separate, but interconnected applications, as shown in Figure 1. The first application, the Symbian application, has access to data stored on the phone (contact and event information). This application then communicates with the Java MIDlet, which renders the animation based on the data it receives from the Symbian application. A settings file is used to

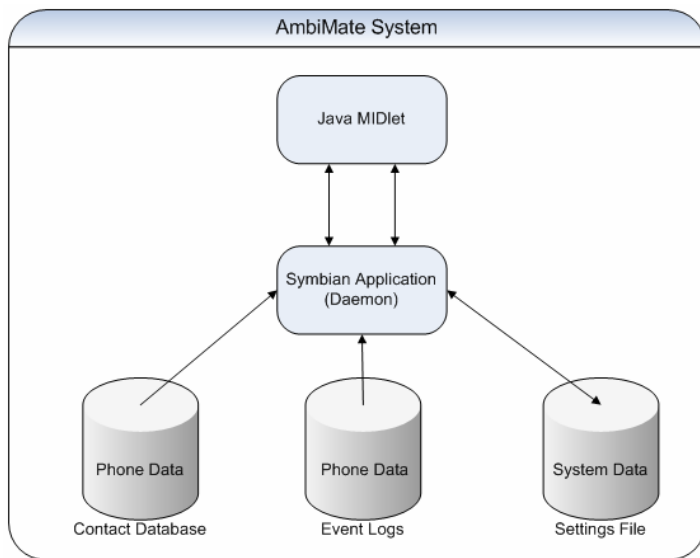


Fig. 1. System architecture of *AmbiMate* System

permanently store user preferences. When changes are made in the customisation portion of the Java application, these changes are sent to the Symbian application which updates the settings file. The Symbian application also accesses the settings file on start-up to determine which data to send to the Java application.

3.4 Data Design

The data needed by the *AmbiMate* system can be divided into two sections – phone data and system data. The phone data is stored on the mobile device itself. Information regarding phone calls and text messages (and any other form of communication that mobile phone users engage in) is automatically recorded in a log file by the Symbian OS when the event takes place. Information regarding contacts, such as is typically entered by users in the “Phonebook” of the mobile device, is also stored on the device.

The system data consists of user settings maintained in the settings file on the device to permanently store user preferences. Table 1 summarises this data.

Table 1. *AmbiMate* system data

Attribute	Description
mode	Currently selected display mode (e.g. Aquarium mode).
datePeriod	Date range currently being visualised.
eventType	The event type currently visualised (phone calls or text messages).
updateInterval	How often the display is updated to include events since last update.
numContacts	The number of contacts currently being visualised.
contactNum (for each contact)	Uniquely identifies a contact.

3.5 User Interface Design

Initial paper-based designs were implemented using a visual designer and run on an emulator to give a more accurate picture of the interface presented to the users. Figure 2 shows the main menu structure of the application. The significance of each menu option is summarised below:

- “Customise”, leads to a sub-menu which allows the user a variety of different customisations (Section 3.1).
- “Details” provides the user with a textual breakdown of the visualised information.
- “Start-up Settings” allows the user to customise the delay before the ambient display is activated, as it is designed to function as a screensaver-type application.
- “Help” leads to a screen giving a brief explanation of the mapping between the properties of the display and the data being visualised.

Figure 3 and Figure 4 show the two ambient display designs. Figure 3 shows the Aquarium Design, with different colour fish representing different contacts, the size of each fish size representing the volume of communication with the corresponding contact and the direction the fish are swimming indicating whether incoming or

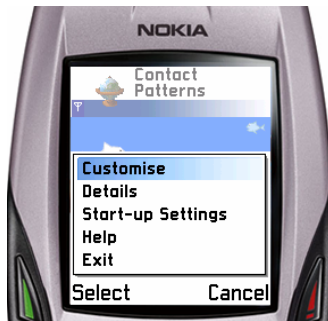


Fig. 2. Main menu structure



Fig. 3. Aquarium Design



Fig. 4. Flower Design (day-time)

outgoing communication is dominant. Figure 4 shows the Flower Design when day-time communication (6am to 6pm) is being visualised. Colour and size (height in this case) have similar meanings as in the Aquarium Design. The colour of the sky denotes whether day-time or night-time communication is being visualised. In the night-time version the sun is not visible and the sky is black.

4 Implementation

A Nokia 6600 was used for testing throughout the implementation phase. This phone model runs Symbian (S60) OS version 7.0 and also runs Java CLDC version 1.0 and MIDP 2.0. This phone represents the typical type of mobile device the system is aimed at, and hence was ideal for testing. Some aspects of the implementation, such as communication between the two applications, proved impossible to test on the emulators provided with the development SDK's, so device-based testing was necessary in order to test this functionality.

4.1 Functional Implementation

As can be seen from Figure 1, the architecture of the system relies on communication between a Symbian application and a Java MIDlet. In order to do this, an implementation built around the "MIDlet Native Services Framework", proposed by Gupta and de Jode [20] was used. A common problem with Java MIDlet development is that for security reasons, applications are unable to access native services running on the mobile device. The Native Services Framework gets around this by establishing a two-way local socket connection on a pre-defined port between a Java MIDlet and a Symbian daemon application. In this way, the Java application is able to access services it normally wouldn't be able to, such as the contact and event information needed by the *AmbiMate* system. This mechanism is used throughout the system. When the user selects customisation options in the Java application, these are sent to the Symbian application which writes them to the settings file. When the animation starts up, it first sends a message to the Symbian application, requesting that it send the information needed to render the display. When the user enters a customisation screen, the Symbian application retrieves the currently selected options and sends them to the Java application, so that the correct options are displayed. Protocols set up on either end of the communication channel ensure that traffic is kept to a minimum. The Symbian application is also responsible for all reading and writing to the settings file, which permanently stores user preferences.

The Symbian OS maintains a contact database and a log engine database, both of which contain phone data needed by the system. The Symbian daemon application provides access to this data, something which proved more challenging than originally anticipated. The original algorithm design included a step which filtered the event logs by contact, something which the Log Engine API reference showed to be possible. When this was implemented, however, it became apparent that contact details were not being recorded at the time the log events were entered, making it impossible to filter the events by contact. Instead, the filtering process had to be done manually. Different phone number formats also proved a problem, as the numbers in

the contact database were often recorded without the international dialing code prefix (e.g. +27 for South Africa). In the phone logs, however, the numbers were recorded with this prefix attached. This problem was overcome by comparing the last nine digits of the numbers in order to determine if an event corresponded to a known contact. Further complication was added because the Symbian OS contact model does not define a standard set of required attributes to be stored for each contact in the contact database. Indeed, a contact may not even have a phone number recorded at all, or may have several different numbers recorded. These include a mobile number, a home mobile number, a work mobile number, a fixed-line number, a home fixed-line number and a work fixed-line number, all of which have to be catered for, if they are present. The API reference was also not clear about which contact model fields correspond to these values.

It was also discovered that the log engine only stores log-event information for a maximum of one month. Since this is likely to represent a large enough volume of data to appreciate contact patterns and writing an application to create a custom log-file system was beyond the scope of the system, it was decided that this data was sufficient.

The log event information was grouped by contact, and split into incoming day-time communication, outgoing day-time communication, incoming night-time communication and outgoing night-time communication. This information was then sent to the Java application, along with the animation settings retrieved from the settings file, in order to facilitate the actual rendering of the animation.

4.2 Ambient Display

The two designs (Figures 3 and 4) that were implemented, namely the Aquarium Design and the Flower Design, provide different interpretations of the visualised data, and hence much of their functionality was implemented separately. Only the Aquarium Design visualises the directionality of the communication data, while only the Flower Design incorporates time of day (when the events took place) into the visualisation. Functionality that could be generalised was implemented in an abstract parent class, which the two animations extended.

The information received from the Symbian application was used to determine the size of the display elements relative to each other, and in the case of the Aquarium Design, their directionality. These calculations obviously had to take into account each element's size relative to the overall height (in the case of the fish) or width (in the case of the flowers) of the screen. Unfortunately, CLDC 1.0 devices, such as the Nokia 6600 on which testing took place, do not provide the standard Java floating point primitive types "float" and "double", making calculation of such ratios a problem. While this problem would not exist in later devices running CLDC 1.1 (in which floating point types were added), it was decided to try and overcome this problem. An API was found (the "MicroDouble" API), which to a degree overcame this problem by storing floating point numbers using a hexadecimal representation in a "long" primitive type. Exact calculations were still not possible, leading to round-off errors, but the format proved sufficiently accurate.

Once the display elements' size (and in the case of the Aquarium Design, directionality) had been determined, their position was calculated. This was done by

sorting the data to position the most frequent contacts (and hence the largest display elements) in the centre of the screen, and less frequent contacts on the edge of the screen.

A 2D drawing Scalable Vector Graphics (SVG) library that is not part of the standard Java Micro Edition class libraries, called “TinyLine 2D” [19], was used to create the animations. Using this 2D library, it was possible to only have a single standard fish and flower template, which could then be scaled to the necessary size for each contact. Translation and rotation transformation matrices could then be applied in order to create the animation effects. Effects such as the movement of a fish’s tail independently of the rest of its body could then be achieved more easily than would have been the case had the same effect been attempted using standard JME libraries.

In order to accommodate the operation of a mobile phone as an ambient display, it was decided to implement the system to operate similarly to a typical mobile phone screen-saver. Such an implementation would mean that the user would not have to consciously activate the display once installed, and that it would run unsupervised in the background – in essence fulfilling the requirements of an ambient display. The architecture of the system implied that a number of steps were required to achieve this goal:

- Starting the Symbian daemon application when the phone boots.
- Disabling the standard system screen-saver.
- Starting the animation automatically after the desired interval.

A number of approaches were tested to start the Symbian daemon application on boot-up. Ultimately, EZ-boot, a boot manager application provided by NewLC, was used to achieve the desired result. EZ-boot waits until the phone has completed the boot sequence, and then launches applications registered with it.

Once the daemon application is booted, it then continues running in the background, running a low-priority thread to check whether the time has elapsed after which the ambient display should be activated. Once the ambient display has been activated, it continues running a low-priority thread in the background to disable the standard system screen-saver which would otherwise obscure the ambient display.

JME provides a useful mechanism for automatically starting MIDlets, known as the Push Registry. Applications can register themselves on a particular port with the Push Registry when they are installed. When the application is not running, the system’s Application Management Software (AMS) listens for incoming connections on that port, and if it detects such a connection, starts the MIDlet registered at that port. This mechanism is used to auto-start the ambient display once the specified time interval has elapsed. At the appropriate time, the Symbian application opens a socket connection on a pre-defined port, which results in the Java MIDlet being activated.

4.3 Certification, Performance and Integration Issues

MIDP 2.0 applications come in two main forms, namely trusted and untrusted applications. Untrusted applications are not signed with the digital certificate of a Certification Authority (CA), and can only access restricted API’s if the user specifically grants permission to do so. This would obviously be undesirable in an ambient display, as given the architecture of the system, the user would be constantly

prompted to authorise network access to allow communication between the Java MIDlet and the Symbian application. As a result, the MIDlet had to be signed.

In order to allow the *AmbiMate* system to integrate seamlessly into the operation of a mobile phone and continue running in the background, it is also necessary for the animation to pause when it loses focus, and resume when it regains focus. Failing to do so would result in phone performance being drastically reduced when the user attempted to perform another task, such as sending a text message. Fortunately, the standard MIDlet lifecycle provides for this, with methods to determine when the MIDlet gains and loses focus [21]. Implementing these methods allows the ambient display to remain dormant in the background while the user is busy, and then resume once the user has completed his/her task.

The issue of whether to leave the backlight on proved a particularly tricky one for this system. Typical system screensavers on devices such as the Nokia 6600 operate with the backlight off, in order to conserve battery power. However, leaving the backlight off with the ambient display active would make it difficult to appreciate the visualisation being performed. A balance needed to be found between conserving battery power, while still making the display visible. The best compromise was to leave the light on for short intervals each time the display was updated. By doing so, the user would be made aware that the display was updating, and the phone's battery would not be put under unnecessary strain.

The limited processing power of mobile devices also had to be taken into account when designing the animations. Attempting to create animations that were too detailed pushed the boundaries of the relatively limited processing power of mobile devices.

5 Evaluation

This section discusses the evaluation of the *AmbiMate* system, the primary goal of which was to evaluate the usefulness of the system.

5.1 Evaluating Ambient Displays

Ambient displays remain a relatively new research frontier in computing and standard techniques for evaluating them are still being developed [22]. Mobile phone-based displays are newer still. Many of the ambient display systems developed previously, have merely been exploratory in nature, with little or no evaluation being conducted.

Those evaluations of ambient display systems that have previously been conducted have used a variety of approaches. Some systems, such as the artistic bus schedule ambient display system described by Skog *et al.* [10], have been evaluated using a combination of field studies and user interviews. Others have been evaluated by testing users' ability to understand and/or recall the information being visualised [22].

Kaikkonen *et al.* [23], compared field studies and laboratory testing for evaluating the usability of mobile applications. They found field studies to be more time-consuming than laboratory testing, and also that field studies provided no significant benefits for evaluating the usability of mobile applications. Skog *et al.* [24] argue that only by conducting longitudinal evaluations of ambient displays can issues specific to their use in a particular environment be uncovered. Hence, a longitudinal field study allowing users to interact with the system on their own personal mobile phones in everyday situations would likely be the best means of evaluating the *AmbiMate* system.

Unfortunately, due to limitations in terms of the number of devices that *AmbiMate* is able to run on, it proved infeasible to conduct field studies. Instead, a more traditional evaluation involving user testing was conducted. This evaluation and its results are discussed in Section 5.2.

5.2 User Testing

Methodology. The *AmbiMate* system was evaluated by eight users. Participants were selected to represent a cross-section of the target user population of the system. As a result, only experienced mobile phone users were selected as participants. Participants included male and female, undergraduate and postgraduate students and staff at NMMU in order to involve a balanced cross-section of the user population.

Each participant in the evaluation was presented with a test plan, consisting of a task list to be performed. The task list comprised the following tasks:

- Activating the system;
- Viewing help;
- Viewing display details;
- Changing the colour associated with a contact;
- Changing the data type being visualised;
- Changing the time period being visualised; and
- Changing the display mode (from Aquarium Design to Flower Design).

A number of questions were included in the test plan after each task, in order to determine the users' understanding of the visualised information, and the changes in the display as a result of the customisations they were asked to perform. On completion of the test plan, participants were also required to complete a questionnaire – a customised version of the widely used Questionnaire for User Interface Satisfaction (QUIS). Users were asked to rate the system according to various criteria using a 5-point Likert scale.

A Nokia 6600 with the *AmbiMate* system installed was used for the user testing. Users were passively observed while carrying out the test plan and their comments were recorded.

Results. Table 2 shows the summarised results for each of the main categories of the user satisfaction questionnaire completed by participants in the evaluation.

Table 2. Quantitative questionnaire results summary (n=8)

Variable	Mean	Median	Std. Dev.
Overall Reactions	4.04	4.04	0.33
Interface Design	4.23	4.20	0.43
Terminology & System Info	4.56	4.63	0.37
Navigation & Functionality	4.64	4.79	0.31
Information Visualisation	4.13	4.13	0.48
Learning	4.25	4.33	0.64
System Usefulness	4.13	4.00	0.64

General comments received were strongly positive. Users commented that the display was visually appealing and easy to understand.

In the final question of the evaluation questionnaire, users were asked to rate the usefulness of the system on a scale of 1 to 5. The mean rating for this question was 4.13, with a median of 4. All but one user (who gave a rating of 3) gave a rating of 4 or 5. This provides a fairly clear endorsement of the usefulness of the system by the participants in the evaluation. Several users throughout the different evaluations also expressed a keen interest in having the system installed on their own private mobile phones.

Users were also asked to rate the usefulness of the visualised information. This question is slightly more complicated, because ambient displays are typically designed to “*support monitoring of non-critical information*” [5]. As a result, the information sources visualised by ambient displays typically include information that is of passing interest, rather than critical importance. Given this, the mean rating of 3.88 and median of 4.0 given by the participants for the usefulness of the visualised information can be regarded as highly positive. Clearly, while the answer to this question remains largely subjective and dependent on an individual user’s needs, the vast majority of users found the information to be useful.

6 Lessons Learned

A number of problems were encountered during the development of the *AmbiMate* system, from which valuable lessons can be learned. Some of these are summarised below:

- A truly device-independent mobile ambient display to visualise personal communication patterns is not possible, mainly because a purely JME implementation is not presently possible (Section 3.2).
- Any application which, like the *AmbiMate* system, relies on data from the contact database needs to take into account the possible lack of information in some fields (Section 4.1).
- While JME proved to be a superior choice for the development of the ambient display, it is by no means an easy task to integrate a Java MIDlet into the operation of a Symbian phone as a screensaver. Special measures had to be taken to achieve screensaver functionality (Section 4.2).
- Special consideration has to be given to battery life when developing mobile ambient displays (Section 4.3).
- Limitations in terms of the processing power of mobile devices also need to be considered when designing a mobile ambient display (Section 4.3).
- It is crucial that Java MIDlets be signed in order to avoid annoying error messages that would detract from the ambient nature of such systems (Section 4.3).

7 Conclusions

The development and evaluation of the *AmbiMate* system described in this paper provide valuable insight into the usefulness of mobile ambient displays. In particular,

the results of user testing show that users found the system and the information visualised to be highly useful (Section 5). However, considering that ambient displays are by their very nature designed to fit into the everyday life of the user, future work incorporating field studies is needed to confirm the results of this evaluation.

The development of the *AmbiMate* system also identified a number of problems and limitations in the successful deployment of a mobile ambient display system, from which lessons can be learned (Section 6). These include lessons regarding implementation tool selection, device independence, battery life, processing power and data access. Seamless integration into the functioning of the mobile device is no trivial task, and the problems encountered and lessons learned in the development of the *AmbiMate* system could provide valuable insight for anyone implementing such systems in the future.

References

1. Smit, T., Hurst, R.: The Power of Mobile IP, ITWeb Market Monitor (2006)
2. Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B., Yarin, P.: Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information. In: Proceedings of 1st International Workshop on Cooperative Buildings (1998)
3. Weiser, M.: The Computer for the 21st Century. Scientific American (1991)
4. Weiser, M., Brown, J.S.: Designing Calm Technology. Powergrid Journal (1995) [Accessed on 22 March 2006], <http://www.powergrid.com/1.01/calmtech.html>
5. Mankoff, J., Dey, A.K., Hsieh, G., Kientz, J., Lederer, S., Ames, M.: Heuristic Evaluation of Ambient Displays. In: Proceedings of CHI 2003 (2003)
6. Redstrom, J., Skog, T., Hallnas, L.: Informative Art: Using Amplified Artworks as Information Displays. In: Proceedings of Designed Augmented Reality Environments (2000)
7. Schmidt, A., Hakkila, J., Atterer, R., Rukzio, E., Holleis, P.: Using Mobile Phones as Ambient Information Displays. In: Proceedings of CHI 2006 (2006)
8. Weiser, M., Brown, J.S.: The Coming Age of Calm Technology (1996) [Accessed on 22 March 2006], <http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm>
9. Ambient Orb. Ambient Devices Inc. (2006) [Accessed on 22 October 2006], <http://www.ambientdevices.com>
10. Skog, T., Ljungblad, S., Holmquist, L.: Between Aesthetics and Utility: Designing Ambient Information Visualizations. In: Proceedings of IEEE Symposium on Information Visualization (2003)
11. Miller, T., Stasko, J.: Personalized Peripheral Information Awareness through Information Art. In: Davies, N., Mynatt, E.D., Siio, I. (eds.) UbiComp 2004. LNCS, vol. 3205, pp. 18–35. Springer, Heidelberg (2004)
12. Fogarty, J., Forlizzi, J., Hudson, S.E.: Aesthetic Information Collages: Generating Decorative Displays that Contain Information. In: Proceedings of UIST 2001, ACM Symposium on User Interface Software and Technology (2001)
13. Streitz, N., Prante, J., Röcker, C., Van Alphen, D., Magerkurth, C., Stenzel, R., Plewe, D.: Ambient Displays and Mobile Devices for the Creation of Social Architectural Spaces. Kluwer Academic Publisher, Dordrecht (2003)

14. Viégas, F., Golder, S., Donath, J.: Visualising Email Content: Portraying Relationships from Conversational Histories. In: Proceedings of CHI 2006 (2006)
15. Nardi, B., Whittaker, S., Isaacs, E., Creech, M., Johnson, J., Hainsworth, J.: ContactMap: Integrating Communication and Information Through Visualizing Personal Social Networks. Communications of the ACM (2002)
16. Xiong, R., Donath, J.: PeopleGarden: Creating Data Portraits for Users. In: Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology (1999)
17. Carpendale, S., Tat, A.: CrystalChat: Visualising Personal Chat History. In: Proceedings of 39th Hawaii International Conference on System Sciences (2006)
18. Canals: Worldwide Smart Mobile Device Research (2006) [Accessed 30 November 2006], <http://www.canalys.com/pr/2006/r2006071.htm>
19. TinyLine: Programmer's Guide to TinyLine 2D API (2006) [Accessed on 11 October 2006], <http://www.tinyline.com/2d/guide/>
20. Gupta, A. and de Jode, M.: Extending the Reach of MIDlets: how MIDlets can access native services (2005) [Accessed on 19 October 2006], http://developer.symbian.com/main/downloads/papers/MIDlet_Native_Services_Framework/MIDlet_Native_Services_Framework_v1.1.zip
21. de Jode, M.: Programming the MIDlet Lifecycle on Symbian OS (2004) [Accessed on 19 October 2006], <http://www.symbian.com/developer/techlib/papers/midplifecycle/midplifecycle.pdf>
22. Plaue, C., Miller, T., Stasko, J.: Is a Picture Worth a Thousand Words? An Evaluation of Information Awareness Displays. In: Proceedings of 2004 conference on Graphics Interface (2004)
23. Kaikkonen, A., Kallio, T., Kekäläinen, A., Kankainen, A., Cankar, M.: Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. Journal of Usability Studies (2005)
24. Skog, T., Ljungblad, S., Holmquist, L.: Issues with Long-term Evaluation of Ambient Displays. IT University of Göteborg (2006)