

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

The Efficient and Cost Effective Asynchronous Mail Sender for Road Warriors

CSC5000W

Kuo-Hung Lee

klee@cs.uct.ac.za

Supervised by

Prof. Gary Marsden

gaz@cs.uct.ac.za

July 1, 2007

Abstract

Email has become a very important communication tool. The rise of SPAM, however, has forced system administrators to restrict the usage of their SMTP servers. Such a restriction causes problems for users who send emails from different locations because it becomes necessary for the users to change email settings when they move between different networks.

This research studied the problems and difficulties users have in sending emails from different networks and then employed HCI techniques to design and evaluate a program aimed at solving these problems. The final program allows the user to set up and manage the connections the user connects to, and apply the SMTP settings specific to the current active connection to send emails. The connection settings contain a set of emailing rules for each connection. Users can use the rules to filter the outgoing emails and thus utilise the connections in a more efficient and cost effective way.

Table of Contents

| | |
|---|----|
| List of Figures | 5 |
| List of Tables | 6 |
| 1 Introduction | 7 |
| 1.1 Statement of the problem..... | 8 |
| 1.2 Purpose | 8 |
| 1.3 Significance of the study..... | 8 |
| 1.4 Research questions | 9 |
| 1.5 Conclusion..... | 9 |
| 2 Background | 10 |
| 2.1 User's Location..... | 10 |
| 2.1.1 Context-Aware Computing..... | 10 |
| 2.1.2 Location-Aware Computing..... | 11 |
| 2.2 Emailing..... | 11 |
| 2.2.1 Simple Mail Transfer Protocol..... | 12 |
| 2.2.2 Sending mail from foreign networks..... | 12 |
| 2.2.3 Use the local SMTP server | 13 |
| 2.2.4 Mailing Rules | 15 |
| 2.2.5 Mailing Costs | 16 |
| 2.3 Similar Applications | 17 |
| 2.3.1 Applications for manual settings change..... | 17 |
| 2.3.2 Applications for automatic settings change..... | 19 |
| 2.4 Summary | 24 |
| 3 Methodology..... | 25 |
| 3.1 Understanding Users' needs..... | 25 |
| 3.2 Prototyping | 27 |
| 3.3 Evaluation..... | 28 |
| 3.4 Conclusion..... | 29 |
| 4 Iteration One | 30 |
| 4.1 Questions | 30 |
| 4.2 Target Users | 30 |
| 4.3 Procedures | 31 |
| 4.4 Contextual Inquiry Results..... | 32 |
| 4.5 The system..... | 34 |
| 4.5.1 Connection Manager..... | 35 |
| 4.5.2 Mail Handler..... | 37 |

| | | |
|-------|-----------------------------------|----|
| 4.5.3 | SMTP Sender | 38 |
| 4.5.4 | User Interface | 39 |
| 4.6 | Demonstration Results | 44 |
| 4.7 | Iteration Conclusion | 44 |
| 5 | Iteration Two | 46 |
| 5.1 | Questions | 46 |
| 5.2 | Procedures | 46 |
| 5.3 | Alteration to the Program | 47 |
| 5.3.1 | Interface Enhancements | 47 |
| 5.3.2 | New Features | 52 |
| 5.4 | The Experiment | 54 |
| 5.4.1 | Objectives | 54 |
| 5.4.2 | The Procedures | 54 |
| 5.5 | Experiment Results | 55 |
| 5.5.1 | Log Analysis | 55 |
| 5.5.2 | Interviews | 57 |
| 5.6 | Iteration Conclusion | 58 |
| 6 | Iteration Three | 60 |
| 6.1 | Questions | 60 |
| 6.2 | Procedures | 60 |
| 6.3 | Alteration to the Program | 60 |
| 6.3.1 | Mail Sending Status | 61 |
| 6.3.2 | Connection Wizard | 63 |
| 6.3.3 | Main Form | 68 |
| 6.3.4 | Options | 70 |
| 6.4 | The Experiment | 72 |
| 6.4.1 | Objectives | 72 |
| 6.4.2 | Procedures | 72 |
| 6.5 | Experiment Result | 73 |
| 6.6 | Iteration Conclusion | 74 |
| 7 | Discussion and Conclusion | 76 |
| 7.1 | Research Questions | 77 |
| 7.2 | Contributions of this study | 79 |
| 7.3 | Future work | 79 |
| 7.4 | Conclusion | 81 |
| | Bibliography | 82 |

List of Figures

| | |
|---|----|
| Figure 1: Web Interface for Email account | 15 |
| Figure 2: Mobile Net Switch | 18 |
| Figure 3: Outlook Express SMTP Bulk Change | 19 |
| Figure 4: Autoroute SMTP..... | 20 |
| Figure 5: Easy SMTP Server..... | 21 |
| Figure 6: SMTP Spotter..... | 22 |
| Figure 7: X-Ray Mail Assistant | 23 |
| Figure 8: The different modules in the program | 35 |
| Figure 9: Tooltip Message | 40 |
| Figure 10: The General tab in the Main form..... | 41 |
| Figure 11: The Options tab in the Main form | 41 |
| Figure 12: The Connections List tab in the Main form | 42 |
| Figure 13: The Mails List tab in the Main form | 42 |
| Figure 14: The About tab in the Main form | 42 |
| Figure 15: The Selection form..... | 43 |
| Figure 16: The New/Update Connection form..... | 43 |
| Figure 17: The Mails List tab for iteration one and two..... | 48 |
| Figure 18: The Connections List tab for iteration one and two | 49 |
| Figure 19: The Options tab for iteration one and two | 50 |
| Figure 20: The About tab for iteration one and two | 51 |
| Figure 21: The Connection Types in the New/Update Connection.. | 52 |
| Figure 22: The Proxy settings | 53 |
| Figure 23: The Mail Sending Progress form..... | 61 |
| Figure 24: The Mails List for iteration two and three | 62 |
| Figure 25: The Connection Wizard..... | 64 |
| Figure 26: The ISP list in the Connection Wizard..... | 65 |
| Figure 27: The SMTP validation in the Connection Wizard | 66 |
| Figure 28: The Mailing rules in the Connection Wizard | 67 |
| Figure 29: The Main form for iteration two and three | 69 |
| Figure 30: The notification for minimising the program | 70 |
| Figure 31: The Options window for iteration two and three | 71 |

List of Tables

| | |
|---|----|
| Table 1: Target Users | 31 |
| Table 2: The icons used in system tray | 40 |
| Table 3: The icons in the system tray in the third iteration..... | 69 |

University of Cape Town

1 Introduction

Emails have become one of the most important communication tools available [22]. People use emails for their business communication and to send documents like proposals, invoices, reports, etc. People also use it privately to keep in touch with friends, show the photographs of their holidays, and to share funny clips with others. Emailing is also cheaper when compared to telephones and faxes. It is also more efficient compared to ordinary mails. And it has better acceptance by the public as a proper communication channel as compared to chat and Instant Messenger programs [22]. Most organisations cannot afford not to have an email system.

The financial cost of sending/receiving emails can differ according to network connection. For people who need to send a lot of emails everyday, they might send emails from work, from home, from a clients' office, or even in coffee shops. When one is at work, sending/receiving emails might be free and reliable. When one is at home, they might be using ADSL, which might be even faster than their office connection, but users pay for every upload and download. One might be using a 3G/Wi-Fi/Dial-up connection when they are away, which is even more expensive, but more convenient. It is our belief that users might want to send only certain emails using a specific connection due to the connection cost, or the connection speed.

When users need to send emails using different connections, they might need to change some settings on their computer. One service provider might not let the users use their outgoing email server when the users are not connected using the specific set of username and password. Some service providers might ask for a login name when users connect to their outgoing email server; some service providers might even make use of the secure socket layers (SSL) to send/receive emails. We hypothesise that these settings are too difficult for most computer users to change, and even sophisticated users might find it annoying to need to change these settings every time they change to a different connection.

1.1 Statement of the problem

The problems for users who send emails from different locations with their laptops are that they need to change email settings which can be either annoying or difficult, and there is no systematic way for users to control what email to send using what Internet connection.

1.2 Purpose

This study investigates making the emailing task easier for mobile computer users who send emails from different locations. This objective is fulfilled by designing and implementing additional software to handle all the users' outgoing emails. This software detects the connection users connect to, and automatically configures a specific email setting suitable for such a connection. The program also decides what email to send using what connection based on the email size, priority, and sending costs.

1.3 Significance of the study

In this study, we investigated users' opinions toward different types of Internet connections in terms of sending emails. We also examined how users manage email settings for different Internet connections. From these findings, the researcher developed a program to simplify the tasks of changing the email settings and filtering outgoing emails when users move between connections.

With the program we developed, users were able to make better bandwidth utilizations when they send emails. Emails will be sent using the most appropriate connection dependent on email size, sending cost, and criticality. It will also make emailing possible on most of the Internet connections users can reach because users will not have to try and remember all the email settings for all the different connections. Furthermore, these benefits also imply possible savings in users' emailing costs because low importance emails can be directed through the cheaper, or even free, connections.

1.4 Research questions

There are three research questions for this topic:

1. How do people think about sending emails in terms of cost, speed and urgency?
2. Will users understand what this additional program can do for them?
3. Will users trust this program to manage their outgoing emails?

1.5 Conclusion

As shown from the above analysis, it can be established that email is an important communication tool. Users use it to communicate to others for both business and personal purposes. Because there are many different types of Internet connections available, it became an issue for users to change the connection specific email settings. In this research, we developed a program to automatically modify users' email settings and to study users' response to the program.

For the next chapter of this report, we will discuss the background information of the emailing problems and the methodologies we employed to perform this study. Then we will go through the development, evaluation and findings for the different iterations of the software, and draw conclusions from this study.

2 Background

In this chapter, we will discuss location information and how it relates to this project. We will also briefly look at the emailing protocols, their problems and possible solutions. For the last section of this chapter, we will look at some existing software that attempts to solve the emailing problems.

2.1 User's Location

This study investigates how users' email sending behaviour changes when they are at different locations. The related fields of study are Context-aware computing and Location-aware computing. We will discuss the two fields in the following section.

2.1.1 Context-Aware Computing

Services that are conditional to the users' context are often referred to as context-aware computing [4]. The context information includes places, time, environments, services, etc. For example, when a student is in the classroom (place) during the lecture time (time), the mobile phone should switch to the silent mode (service). The action of switching to the silent mode is a context-aware service. The goal context-aware computing is trying to accomplish is to make the interaction between users and the interface effortless, or even invisible [21].

There are different definitions for context-aware computing, such as: "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task." [4] Or "adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time" [33]. What we are trying to do in this study is to modify users' email sending services based on their location information. The other context information such as people, social situation, and near-by resources [32] were not required in the process of modifying the emailing services. Thus Location-aware computing will be more

suitable in this study than Context-aware computing.

2.1.2 Location-Aware Computing

Location-aware computing is a branch of context-aware computing. It uses only the location information to provide useful services to the users. The focus of location-aware computing is on how to sense the location of the user, and how services can be modified to suit the location of the user, etc. Generally speaking, location information [18] [28] refers to the user's physical location and different services can be provided based on this information. Location-aware computing makes use of such information and it adapts services based upon the location change.

There are a number of ways to retrieve the location information for a given user. These technologies come with different levels of accuracy, coverage, and costs. The popular commercial mechanisms such as RFID, GPS system, Wi-Fi and mobile phones are very good at getting users' location information with different cost and accuracy, but none of these are suitable for our study. For this study physical location is not important, we are only interested in location so far as it relates to the Internet connection the user connects to. Thus the simplest and most practical solution is to identify the user's connection instead.

For this study, different locations actually mean different connections users connect to. There can be a home connection, office connection, or even a mobile connection to represent the user's location – it is possible to imagine a physical location which affords multiple network "locations". We will use the connection information to determine how the emailing settings will be adjusted.

2.2 Emailing

There is a large amount of research focusing on email systems. For instance, some research focuses on the automation and machine learning for mail clients [16]; some focuses on the integration of mail client and personal information management (PIM) [42] and some focuses on how the emails should be displayed to the users [41]. But so far, there appears to be no research about how the outgoing emails should be handled by the mail client,

and the different criteria users may have for sending emails. In the following section, we will look at the email sending protocol and the constraints of the current systems.

2.2.1 Simple Mail Transfer Protocol

The Simple Mail Transfer Protocol (SMTP) [27] is the standard protocol for sending electronic mails across the Internet. The mail client programs connect to the SMTP server to send emails. The email is then relayed to the SMTP server of the mail recipient which then puts the email in the recipient's mailbox.

This concept worked fine until the rise of 'Spam'. Spam is also known as Unsolicited Commercial Email [7]. It is sent by the advertiser to a list of thousands of email addresses. According to the Internet security industry, more than 90 percent of all Emails in cyberspace are spam [40]. In order to prevent Spam senders from using random SMTP servers, Internet Service Provider (ISP) companies started to restrict the usage of their SMTP servers [8]. Some ISPs make the users authenticate themselves before sending emails. In most cases, the ISP companies just check if the mail sender is connected to their network. This method requires no additional settings to be changed on the users' mail client program, other than the SMTP server address, which reduces the amount of possible support calls.

2.2.2 Sending mail from foreign networks

As we mentioned earlier, user authentication is one way of sending mail when connected to a foreign network. There are two main alternate authentication methods: User Logins and the POP-Before-SMTP method. For the login method, users are requested to log in to the SMTP server in order to commence the mail sending process. Once the user is identified as a user registered on the system, the mail will get sent. The problem with this method is that users need to set up their mail client so the mail client will perform the login process.

The second authentication method is POP-Before-SMTP (also known as Receive-Before-Send). What this means is that users are requested to perform

“Receive Mails” before performing “Send Mails”. What this method does is when the user logs into the POP/IMAP server to receive emails, their current IP address will be recorded on the server for a period of time (normally several minutes). When the user tries to send emails from the foreign networks, the SMTP server will check if the sender’s IP address is in the record, if not, the mail will be blocked.

By design, this method does not require any additional setup. Users will receive mail when they start their mail client, then reply to emails during the same session, and send them off. However, this method does not work if the user fails to send mails before his IP address is removed from the server. It also fails if the user does not attempt to receive mail before he/she sends mails.

User authentication is one solution for sending mails from foreign networks. The other solution is to connect to the same subnet the SMTP server belongs to, thus mail will not be blocked anymore. This can be done by VPN, SSH Tunnel, etc. With this method, users can send emails as they belong to the same subnet as the SMTP server. The downside of this method is that it requires a complex setup process: there needs to be a gateway already set up to accept the SSH/VPN connection from the user, and the user’s machine needs to have the right software/driver installed in order to initiate the connection to the gateway.

The biggest problem with all the solutions mentioned above is that they require certain degrees of settings/services modifications which only ITS/MIS technicians have the privilege to change. It is impossible for the users to use the SMTP server from a foreign network if none of the modifications are performed.

2.2.3 Use the local SMTP server

Since it might be impossible for the users to use the SMTP server from foreign networks, what can the users do? Users could have their own SMTP server running locally so he/she does not have to worry about mails being blocked by the remote SMTP server. Alternatively, users can make use of the web-interface (if any) provided by the mail server. In that way, users can

send/receive mails from the browser of any computers connect to the Internet. Another solution is to use the SMTP server provided by the Internet Services Provider (ISP) users currently connects to.

To run the SMTP server from users' own machine looks like a possible solution, although it sounds like overkill. The fact is it might not work because some SMTP servers now use the Distributed Sender Blackhole List (DSBL) [5] to block mail from the IP addresses identified as possible SPAM senders. For users with dynamic IP addresses, the sequence of IP addresses assigned to the user might be in the series of IP addresses recorded for suspicious mailing activities, thus the mail is blocked. Even worse, users might be identified as SPAM senders by the SMTP servers with sensitive SPAM filters.

Web interface to email is very convenient and accessible. Squirrelmail [38] (see figure 1) and Roundcube webmail [31] are two examples of webmail systems. With the web interface for mail, users can check email from anywhere with an Internet connection and web browser. They do not even have to download the mail into the machine they working on which means it is safe to send/receive mail on other people's PCs, or even public PCs. The problem with the web-interface is that it only works when there is an Internet connection. No Internet, no mail! It does not work for users who want to download all their mails and respond to the mails offline. It also does not work for users who want to have all the email correspondence stored in their own machine.

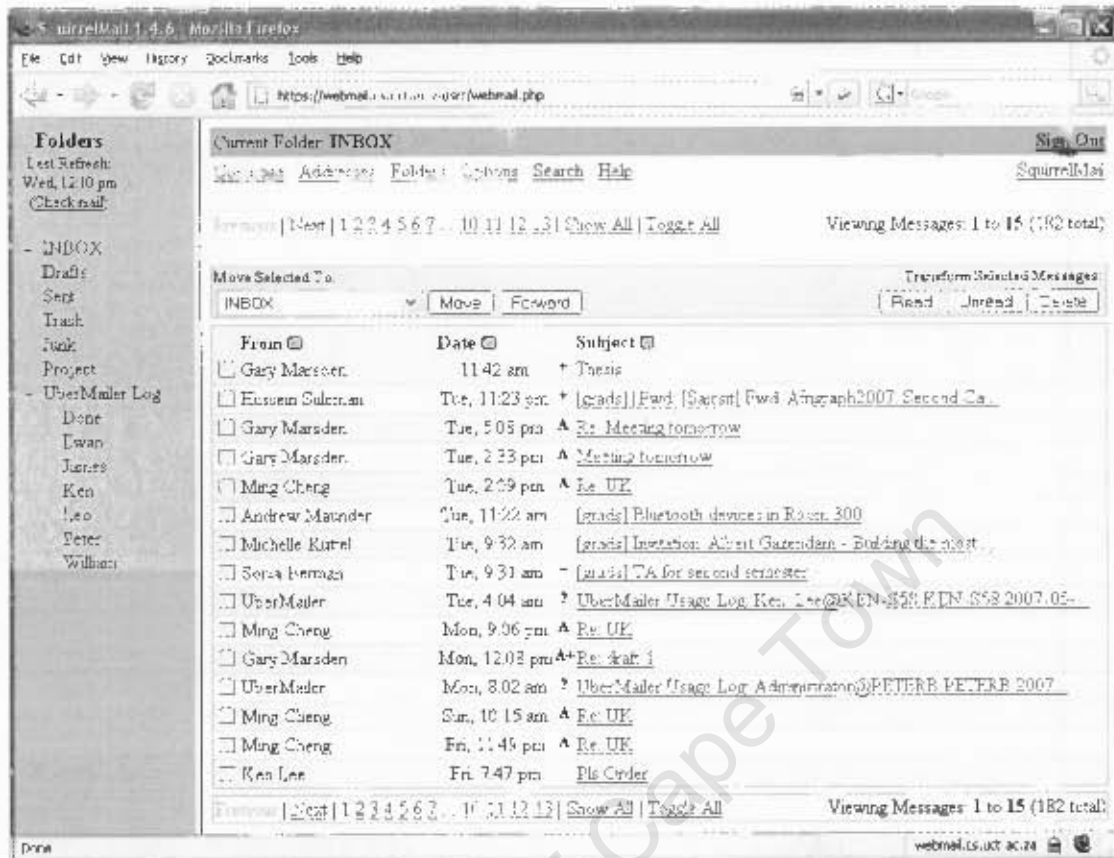


Figure 1: Web Interface for Email account

Using the SMTP server provided by ISPs can eliminate all the mail sending problems because the users will always be in the same subnet as the SMTP server. The disadvantage of this method is that users will need to modify their SMTP settings every time they change connections. The settings might include the SMTP server address, port, authentication, SSL, etc. It might be too complicated or annoying for the users to change every time they change connection.

In this study, we use the SMTP servers provided by ISPs to send mails for the users. When users change to a different connection, the mail settings for the new connection will be applied automatically. Users will not need to change any mail settings manually.

2.2.4 Mailing Rules

Mailing rules are often referred to as filters. Mail client programs use the preset rules to determine how the incoming mails should be organized and prioritized.

The rules can be used to filter out the irrelevant mail and to highlight mail from important senders, etc. The goal is to make it more efficient for the users to respond to the emails which require more attention. The rules can either be set by the users, or created through machine learning of analyzing users' behaviour [19].

The question is, what about the outgoing emails? There is research about automatically created rules, black lists and scoring systems for the SMTP server to prevent spam from relaying to other SMTP servers. Similar rules are also used by the SMTP server to flag certain mails to the users as possible spam. However, we have not come across any research related to applying mail rules for outgoing emails from mail client software.

The mail client programs treat all the outgoing emails as equally important and the outgoing emails are sent in the order the user sends them. The users are expected to manually send the mail at the time they want the mail to go out. It works fine when there is only one connection users are connected to, but when there are several connections users are connected to, they might prefer to send certain mail through certain connections because of the mail size, the connection speed, etc. It means that users will need to keep the emails in the draft folder in the mail client programs, and send the emails out manually when they have an opportunity to connect to the connection they want to use to send the mails. In this study, we implemented mailing rules for outgoing emails, and tried to find out how users think about the rules, and how they use them.

2.2.5 Mailing Costs

There are many different types of Internet connections available. The cost for connections can be charged by the time users spent on the Internet, or by the amount of data transferred across the Internet. When an email is sent using a different connection, the cost for sending the email can be different because the per MB cost for each connection can be different, and the time needed for sending the email is different because the connection speed is different.

For normal desktop users, they only connect to one network, thus the mailing cost stays the same. But for mobile users, they might send emails from a number of different connections such as home, work, 3G/Wi-Fi, etc which

implies that the cost and the time required for sending emails is always different. Because of these differences, users might prefer to have the bigger emails sent with the faster connections and to have the less important emails sent with the cheaper connections. But for the important emails, they will want to have it sent with the first connection available. The emailing rules we implemented for this study could help the users to reduce the mailing costs by routing the less important emails to the cheaper connections.

2.3 Similar Applications

What we are trying to implement for this research is a software system which changes a user's SMTP settings based on the Internet connection he/she uses, and to apply a different set of mail sending rules for every connection. There is software on the market which attempts to solve the mail settings problem. We will talk about these applications in the following section.

2.3.1 Applications for manual settings change

These are the software packages which simplify the steps for changing the SMTP server settings. None of these applications can modify the SMTP settings without users interacting with the application, and none of these applications provide the mail sending rules for the users.

Mobile Net Switch

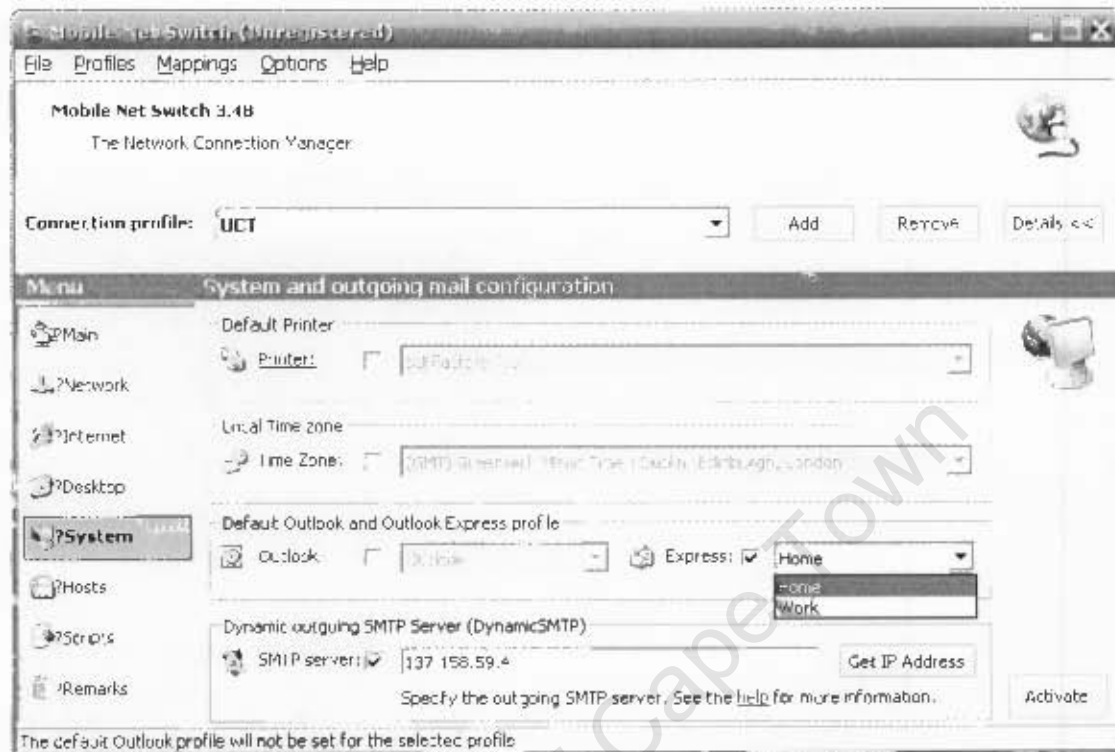


Figure 2: Mobile Net Switch

Mobile Net Switch [20] is a commercial software package for changing the network settings when the user moves to a different network (see Figure 2). This application can change the IP settings, network drive mappings, default printer settings, SMTP servers, etc. In this application, each set of connection settings is seen as a different profile. The user needs to create different profiles for different network settings. When users change location, they will need to select the profile suitable for the new location from the application, and all the network settings for the new location will be applied according to the profile.

Outlook Express SMTP server bulk changer

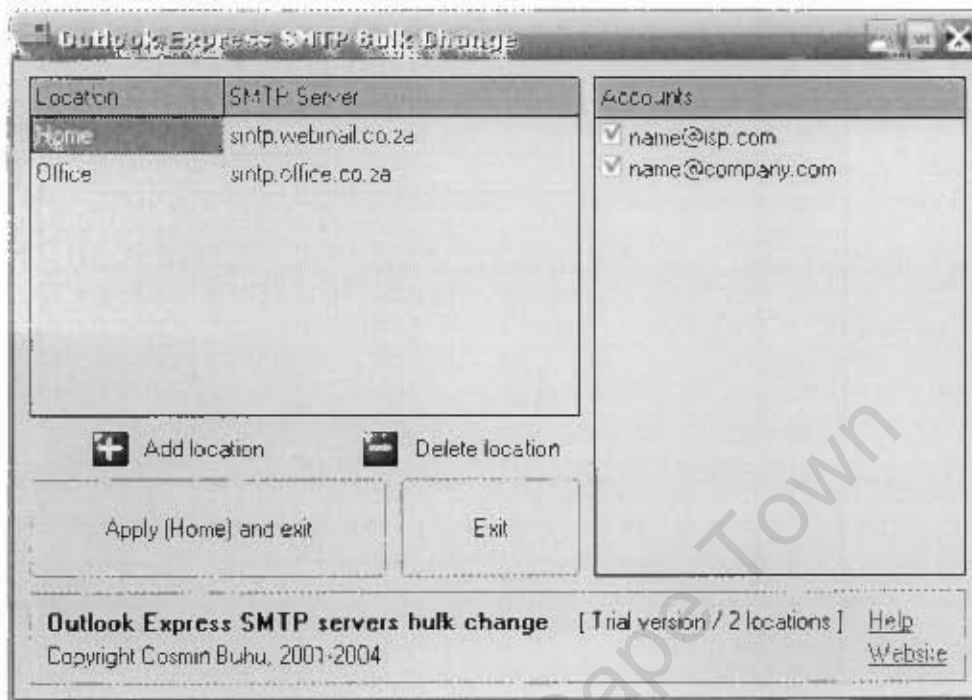


Figure 3: Outlook Express SMTP Bulk Change

As the name suggests, this application [25] is designed to simplify the process of changing the SMTP servers for Microsoft Outlook Express mail client (see Figure 3). In this software, there is a list of SMTP Servers entered by user and a list of email accounts in Outlook Express. When a user wants to change the SMTP server, they can select the email accounts they want to modify, choose the SMTP server and then apply the change. The biggest problem with this software is that it only supports Outlook Express, and users need to restart Outlook Express in order for the change to take place.

2.3.2 Applications for automatic settings change

Ideally, the settings for SMTP servers should be updated automatically when users move to a new location. But it never works that way in reality because there is no algorithm available to correctly detect all the different types of connections the user connects to. The following are some of the applications attempting to change settings when the user connects to a different connection. Still, none of these applications provides rules for the outgoing emails.

Autoroute SMTP

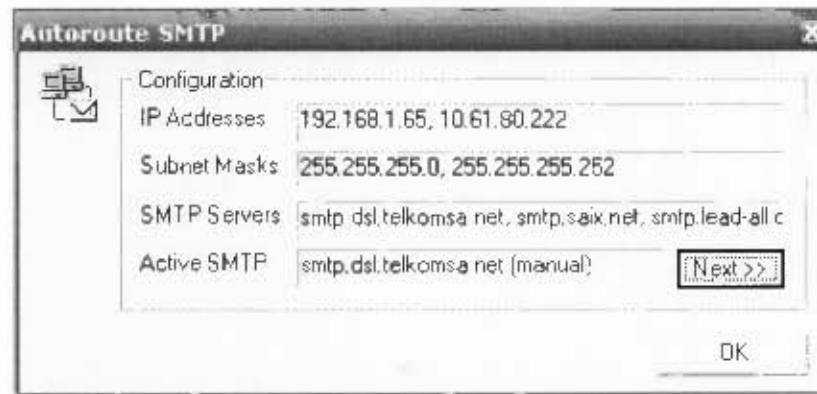


Figure 4: Autoroute SMTP

Autoroute SMTP [1] is an application designed to change the SMTP settings automatically when a user changes network (see Figure 4). How it works is when the user connects to a network, the software compares the user's IP address to the IP addresses of the list of SMTP servers entered by the user. The SMTP server in the same subnet as the user will be used for the specific connection. This application works well if it is guaranteed that all the SMTP servers the user attempts to use are in the same subnet as the user. However, this is not a given.

The problem with this application is that users will need to manually choose the SMTP server to use if there is no SMTP server in the subnet the user is in. Also, this application does not remember the SMTP server selections: every time the user connects to the connection where no SMTP server exists in the same subnet, they will need to go through the SMTP servers list to choose the right SMTP server to use.

IBM/Lenovo Access Connect

This is proprietary software [12] designed for the IBM/Lenovo laptop range of computers. This software is similar to the Mobile Net Switch software as it is also used to modify connection settings. The main differences in the features are that this software can be set to identify different connections by the MAC address for the router the user connects to, and it does not provide features for changing the SMTP Servers.

This is the only software we found which can detect different connections

correctly. The problem is that this connection identification method requires the MAC address of the router, but the MAC address information is not accessible over dial-up connections. Although this application does not change the SMTP settings, the method they used to identify LAN connections works effectively. Therefore, the connection identification method will be used in our research to detect LAN connections.

Easy SMTP Server

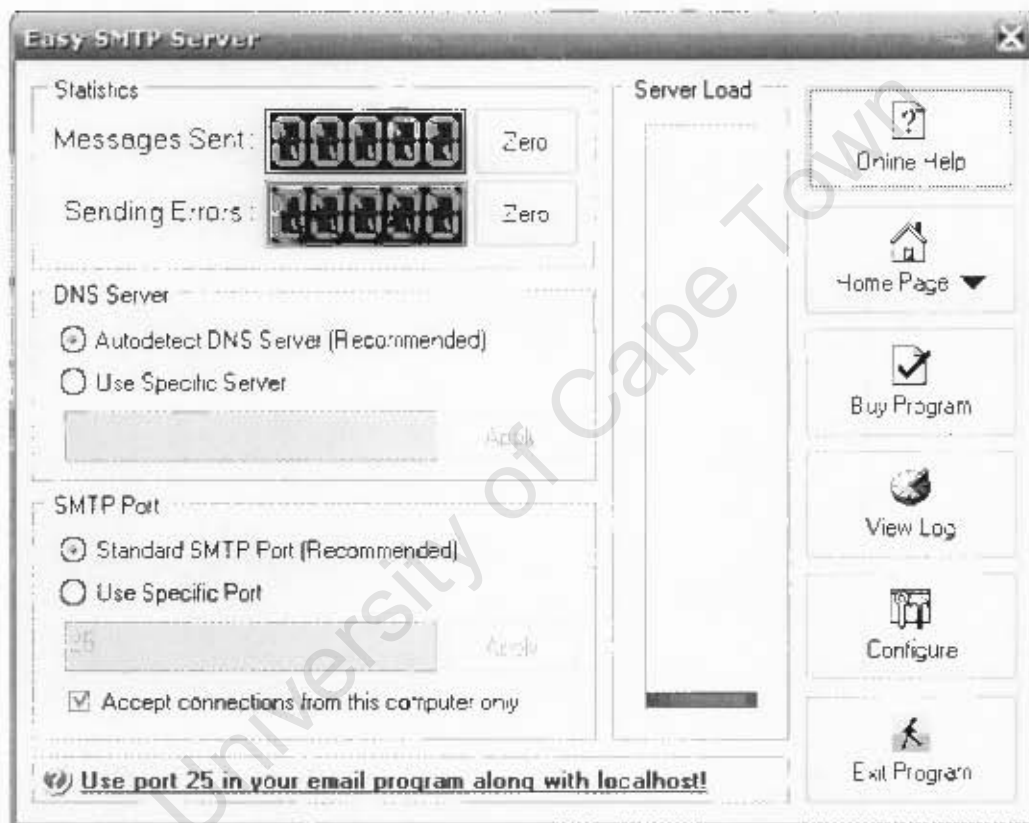


Figure 5: Easy SMTP Server

This application [6] runs a small SMTP server from the user's laptop (see Figure 5). Thus, no matter what Internet connection the user uses, he/she will never need to modify the SMTP settings. The problem with this application is that most SMTP servers use the Distributed Sender Blackhole Lists as discussed earlier. Thus, mails to those SMTP servers will be blocked, or never get delivered to the recipients.

SMTP Spotter

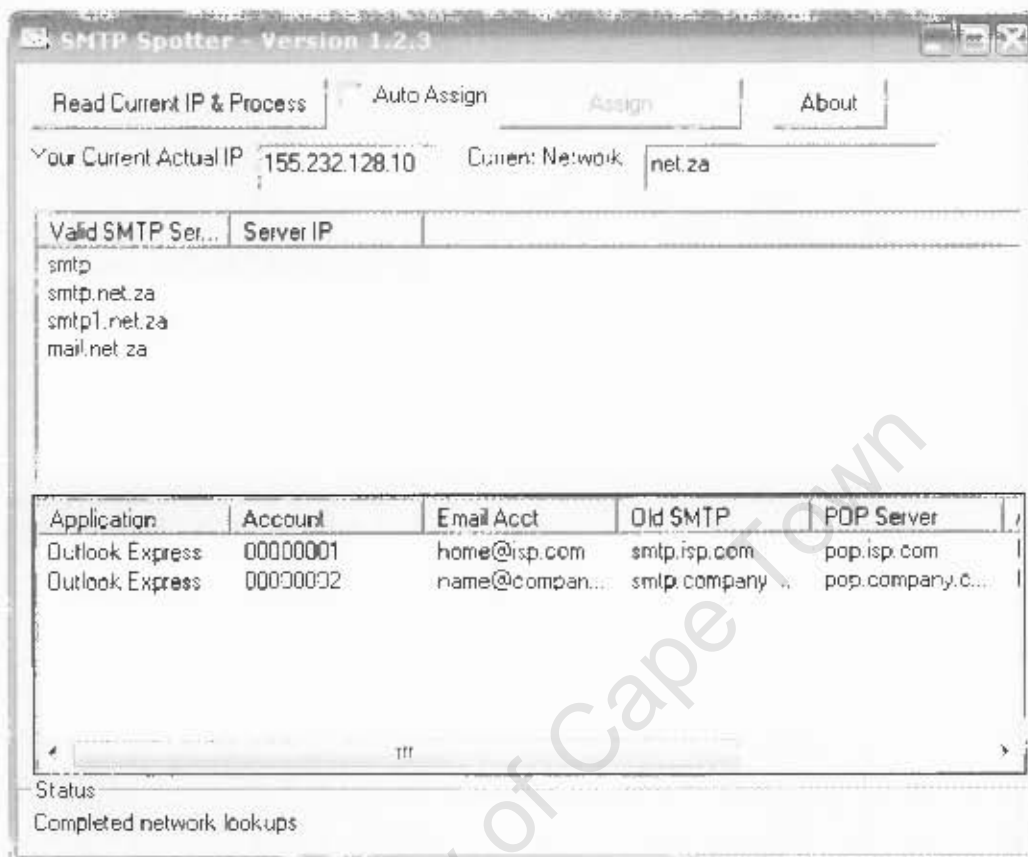


Figure 6: SMTP Spotter

This software [37] is also used to modify the SMTP settings on the mail client (see Figure 6). The special feature for this application is that it tries to figure out what the SMTP server address is. Users only need to execute this application, and it will figure out everything for the user. This application tries to get the domain name for user's IP address, then, based on the domain name, start to guess the obvious SMTP server addresses such as "smtp.domainname.com" and "mail.domainname.com". Then, the software tries to connect to these names it came up with, and see if any exists. It will set the SMTP server to the first server it successfully connects to.

This application will work only if the user's public IP address is a registered domain name, and that this domain has a SMTP server called something like "smtp.domainname.com" or "mail.domainname.com". If all the addresses fail, the application will fail too. The other weak point of this application is that this application only works with Microsoft mail client applications such as Outlook and Outlook Express, and the users' mail client needs to be closed before this

application is executed.

X-Ray Mail Assistant

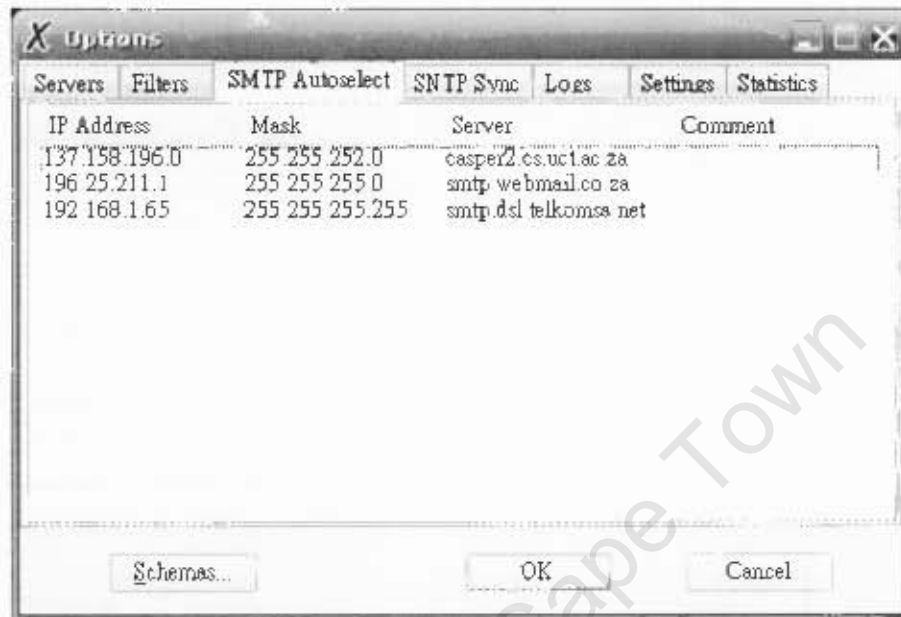


Figure 7: X-Ray Mail Assistant

The X-Ray Mail Assistant [44] is a mail filter application which scans and edits the mail headers for both incoming and outgoing emails (see Figure 7). The SMTP server selection is an additional feature to this application. Users will have a list of SMTP servers stored in this application. The application allows the users to select SMTP servers manually, or to have the SMTP server paired to the user's IP address for SMTP auto select. When the mail client programs try to connect to the SMTP server, it will be routed to the selected SMTP server to send emails.

The main problem with this application is that most of the routers assign IP addresses in the range of 192.168.*.* to the users by default. Thus, it might not be possible for the program to identify connections correctly. If the connection identification does not work, the SMTP server routing will often fail to connect to the right server.

All the applications we discussed have their pros and cons. In this research, we will integrate the techniques we found useful from these applications into our prototype. The other thing we discovered is that none of the applications have rules for the outgoing emails. We will have the outgoing mail rules

implemented in our prototype for users to evaluate their usefulness.

2.4 Summary

In this chapter, we discussed location-aware computing, emailing and applications designed to solve emailing problems. Emailing is a problem for users who constantly move between networks because their mail often got blocked by different SMTP servers. There are many different software applications on the market that try to solve users' problems, but most of the time the software does not work properly. There are email rules users create to manage their incoming mail, but none of the rules can be applied to the outgoing emails. The focus of this research is to apply user-centred design techniques to identify problems users have with sending emails from different locations, and to design a program which is useful and usable to help the users eliminate their emailing problems.

University of Cape Town

3 Methodology

According to Hewett et al. [10], "Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." For the field of computer science, the focus of HCI is on interactions between the users and the machine. In order to improve the users' interaction experience, designers need to understand users' needs, design a system based on their needs, and to evaluate the design in order to improve it. There are many different HCI techniques, each serve difference proposes and has its advantages and disadvantages. In this project, we will focus on the techniques and the use of the techniques rather than the theories behind them.

3.1 Understanding Users' needs

It is important for system designers to know what the users' needs are before they can design a system which will be able to solve the problems users have. There are different techniques employed by usability experts to understand users' needs. In this research, the techniques we have chosen to follow are Contextual Inquiry [11] and Diary Study [9].

Contextual Inquiry in this instance will be in the form of a one-on-one interview held at user's work place which focuses on observations of ongoing work. The user will be asked to demonstrate how he/she performs a specific task which the designer is trying to simplify/replace. The interviewer will then ask the user the reasons behind certain actions which interest the researcher. The session will continue until all the tasks are completed. The information captured during the session is then used to inform the design of the final system. The advantage of this technique is that this technique gives the researcher the opportunity to question the reason behind the user's action when it is unclear. The disadvantage is that this technique needs to be performed in the users' work environment.

Diary Study is a technique used to discover users' needs by analyzing their activities for a period of time. We propose to use it in order to gain insight into

the users ongoing interaction with our software – the software can only be properly evaluated in a longitudinal study. During the diary process, users record their activities to the medium provided by the researcher. The researcher then studies the activities recorded to identify the issues users have with regard to specific tasks. A solution for the issue is then developed based on the findings. The advantages of a Diary Study are that it is relatively cheap and requires fewer resources compared to direct observation, while achieving similar results [9]. Also, it is good for discovering issues users might not be aware of. However, this technique is time consuming and less controllable because it is difficult to ensure the users will complete the entries at the appropriate time.

There are other techniques to find out users' needs such as Artifact Walkthrough [2] and Focus Groups [30]. Tasks at the workplace are normally focused around documents. Artifact Walkthrough is a technique where the users explain the life of an artifact (often documents) to the researcher. When the users go through the creation and the use of the artifact step by step, the researcher will have an opportunity to understand the process of the specific task, thus discovering the users' needs [34].

The Focus Group technique is a group discussion between a group of users and a skilful facilitator who leads the session to discuss issues for a specific task. The conclusion of the discussion will then be used to guide the design of the new system. The advantage is that the discussion group can brainstorm the ideal system which is best suited to all the users in the group. However, it might also lead to a 'groupthink' situation because users want to please others in the group [15]. Groupthink is a situation where all the participants unanimously agree with no dissent. It happens because the participants know that being opinionated often lengthens the discussion, and upsets other participants.

For the first iteration of this study, we will use the Contextual Inquiry technique to discover users' needs. It is more suitable than Artifact Walkthrough because sending emails is a task which does not involve the creation of any artifacts. It is more suitable than focus group because it does not require as many users to perform the session and users do not have to be present simultaneously. Furthermore, because the targeted user group for this study is very specific, the interviewee for the Contextual Inquiry will be the experts in performing the

tasks we are trying to analyse. The Diary Study technique will be used to analyse the log file from the first experiment to improve the software prototype.

3.2 Prototyping

Prototypes are the artifacts used by the researchers to demonstrate the design ideas to the users, to see if it is suitable for the users' needs [42]. There are two different types of prototypes, low-fidelity prototypes, such as sketches of a user interface by pen and paper, and high-fidelity prototypes, such as a beta version of the software system. The benefit of prototyping is that it provides an opportunity for the researchers to test the design ideas before implementing them. The comments and critiques from the users will help the researchers to adjust and improve the designs.

Low-fidelity prototypes are easy to build and the designers can use the paper based interface of the software to talk users through different scenarios. It can be used to express the user-interface without misunderstandings sometimes caused by verbal conversation [35]. The disadvantage for this type of prototyping is that it is very rough and users might not get the feeling of interacting with a proper software system.

On the other hand, high-fidelity prototypes are more responsive to users' interactions. There are two different categories of high-fidelity prototyping; vertical prototyping and the horizontal prototyping [17]. The vertical prototype tests a specific feature in the system. The horizontal prototype shows the users most of the features, but none of the features are actually functional. The high-fidelity prototype is good at showing the users what the final version of the software will look like, and what features it provides. However, it is more time consuming, and it can give users the wrong impression that the implementation of the software system is almost completed.

In this research, the only way to get meaningful evaluation results is by giving users the software to use for a long period of time. This approach necessitates high quality prototypes. The technique we employed is Iterative Implementation. The process will start with an alpha version [26]; it is the fully functional version of the software for day-to-day internal use. The users use the software and give feedback to the designer based on the users' experience.

The feedback might be software bugs, design issues, etc. The designer will then make appropriate changes according to the feedback. A revised version of the system will be released after the alpha version. This process will be repeated until the final release of the software system.

3.3 Evaluation

The usefulness and usability of any software system is best evaluated by its users. There are many different techniques to get users' feedback for a specific design. Since we are going to apply the Iterative Implementation technique, we will focus on those evaluation techniques which are designed to evaluate interactive (high-fidelity) prototypes. During the evaluation, we are trying to find out if users understand what this application can do for them and if they will trust this application and use it to send their emails. The techniques we decided to use are the Conceptual Model Extraction, Log Analysis and interviewing.

Conceptual Model Extraction [20] is a technique used to evaluate the design of screen elements. It reveals user's interpretation of the user interface. During the evaluation session, the interface sketch or the screen shots of the software will be presented to the users to find out what they think each element does, and how certain tasks can be performed with the interface. The result of the evaluation sessions help the researchers to understand how users think the interface works, and thus design the interface best suited to the users.

The Log Analysis [14] technique is a branch of direct observation where the researcher studies the log file after users use the program for a long period of time. This technique gives the researcher an opportunity to understand what users think by studying the way users interact with the prototype. The difference between this technique and direct observation is that for direct observation, the researcher needs to be present when user tries to use the application to complete tasks. The appearance of the researchers could easily bias the experiment.

Interviewing [15] is a technique which the interviewer and the user have a one-on-one discussion after the user has used the prototype. The interviewer will have a list of questions for the user to answer and the user answers the

questions in their own words. Interviews can produce qualitative results for the design team about what users like/dislike; how they feel about the design; what they struggled with, etc. The downside of this technique is that users might ignore or forget about something worth mentioning to the interviewer, which leads to missing information.

There are other evaluation techniques such as Questionnaires [29] and Heuristic Evaluation [23]. Questionnaires is a technique which provides users with a list of set questions to answer. The benefit of Questionnaires is that they have the potential to reach a wide audience, but it is not as flexible as Interviews. For this study, we are focused on a specific group of users, which is not a big audience, and some of the questions we have for the users are based on their behaviours as recorded by the Log Analysis which is unique to each user. Thus the Interview technique will be more suitable for this study.

Heuristic Evaluation is a technique in which usability experts analyse the interface of a software package and give a list of problems in the design. This technique is good at discovering the faults in the design, but a faultless design is not necessary the design users want to use. In this research, we are trying to find out if users will use this application to send all their emails, thus direct observation techniques such as Log Analysis will be more suitable than Heuristic Evaluation.

3.4 Conclusion

This research will employ the relevant methods discussed in this chapter, and result in a program designed according to users' needs. The program will assist mobile users by simplifying the tasks of sending emails from different locations. It determines what connection settings to use, and what mails to send using what connection. In the following chapters, we will talk about the problems we tried to solve, the procedures we followed to solve the problems, the results we uncovered, and the analysis we performed.

4 Iteration One

This project was aimed at solving emailing problems for mobile users. Thus it is important for us to find out what problems the users have. What we undertook in this iteration was to use Contextual Inquiry to find out users' needs, and to come up with an initial solution addressing those needs. In this chapter we will discuss the results of the Contextual Inquiry and the design of the alpha version of the software.

4.1 Questions

The following are the questions we addressed in this iteration:

Target users

- Who is the target user for this study?
- How do we identify the target users?

Users' problems

- Do users have problems sending emails from different locations?
- Is there any dissatisfaction users have in regard to the way their outgoing emails are handled?
- What concerns do the users have in terms of emailing costs?

Using Additional Program

- Will the users want to use an additional program to handle their outgoing emails?
- What features do the users expect from such program?

4.2 Target Users

The target users of this study were computer users who use laptop computers, send at least 20 emails everyday, and sending emails from at least two different locations. It could be sales people, trainers, consultants, directors, etc. During the first iteration of the study, the researcher identified seven users who

fit the profile, as a sample set for this study. The samples were identified using a convenience sampling [39] method. As the name suggested, they are the colleagues, family members and friends who fit the profile, are willing to participate and are easy to access. Users' computer knowledge ranged from computer novice to computer experts. Table 1 shows the list of users we identified for the Contextual Inquiries:

| Users initial | Gender | Computer skills | Occupation | No of conn. used | Email corresp. per day |
|----------------------|---------------|------------------------|----------------------|----------------------------|-------------------------------|
| AY | Male | Advanced | Investor | 3 (two offices and 3G) | 40 |
| CR | Male | Expert | Consultant | 2 (work and 3G) | 35 |
| GS | Male | Advanced | Financial planner | 4 (work, home, client, 3G) | 45 |
| KG | Male | Novice | Training facilitator | 2 (work, hotel) | 20 |
| PB | Male | Expert | Supervisor | 4 (work, home, hotel, 3G) | 30 |
| RR | Male | Novice | Trainer | 2 (work, Sentech) | 20 |
| SR | Male | Advanced | Account manager | 2 (work, 3G) | 25 |

Table 1: Target Users

4.3 Procedures

The researcher first performed a Contextual Inquiry with each individual in the sample group. Most of the sessions took place in users' work place. During the session, the researcher asked users to demonstrate how they send emails from different locations, and to gather users' opinions on sending emails in terms of cost, speed, and urgency. The feedback from the contextual inquiry session was analysed to get information like: do the users fit the profile; what is their knowledge of the technical details; what are their emailing behaviours; what are their needs and expectations? The results of the Contextual Inquiry were used to form the user requirements for the system.

An alpha version of the emailing program was designed and implemented after the Contextual Inquiry sessions. The user requirements were treated as the guidelines for the design of the emailing program. When the alpha version of the program was complete, a demonstration of the program was performed to each individual in the sample group to get their feedback. The demonstration sessions were used to show the users the features and the interface of the program, and how the program works. Comments from the users on the interface and the software complexity, features' usefulness, etc were then collected and analysed.

4.4 Contextual Inquiry Results

This is the summary of the Contextual Inquiry results. During the sessions, seven users (see Table 1) were interviewed. The following is the feedback for each question:

User's understandings for different type of mobile Internet connections (3G, Wi-Fi, GPRS, EDGE, and HSDPA)

- Connections such as 3G, Wi-Fi and HSDPA were familiar to the users because they heard of them often from the network providers' advertisements.
- There are two users who never heard of EDGE. Only three users understand what EDGE is for. The rest of the users had heard of EDGE, but were not sure what it is for.
- Three users confused GPRS with GPS (Global Positioning System) while the rest of the users identified it correctly.
- For the connections types familiar to the users, they know which one is fast and which one is slow, but they cannot tell the actual speed of the connections.

How user sends emails from different locations

- Most users use their office connection and mobile connections (3G/iBurst [13]/Sentech [36]) to send emails.
- Four users make use of web-based email interfaces if the SMTP server cannot connect.
- Three users still use a dial-up connection as a backup connection when they are out of the mobile connection coverage.

Problems user has for sending emails from different locations

- Four users are aware that they need to change the SMTP server before they send emails. They find it quite annoying.
- Two users had their mail client setup by their IT staff so their email accounts correspond to the connections the users use. The only difference in the email accounts is the SMTP server address. When the users send emails, they simply select the account appropriate to the connection they are using. Users are aware that the correct account needs to be selected before they send emails or else the email will fail to send, but they are not clear about the reason behind it.
- Users do not like the web-based emailing interface. They use it only because their SMTP server does not work.

Complaints about how outgoing emails are handled

- Three users complained about missing emails. It is mainly caused by their antivirus software which scans the mail while it is sending out.
- Two users said they want to be able to find out if mail has arrived at its destination.

How user sends emails from new location

- Users will try to send emails using their wireless broadband connection (3G/iBurst/Sentech). They also try to use dial-up connection.
- Some users also use other people's machines (Internet café, clients' machine, etc) with web-based interface to send emails.

Experience of changing connections before sending emails and reasons for it

- Some users change to faster connections to send emails because they have big mails to send.
- Some users change connections in order to reduce their bandwidth/bundle usage on certain connections.

Emailing costs and ways to reduce such costs

- Average costs for emailing is about R100 per month.
- Most users never attempt to reduce the cost.
- Certain mails can be delayed to save cost, but users need to have control of what to send and what to delay.

Mailing queue: Sending order; expiration time; when to send it

- Some users want the mail to go out according to its priority.
- Some users want the mail to go out in the sequence they send it because users already reply to mail according to the mail priority.
- Mail should go out when there is a connection available. But users want to be able to decide what to send and what to delay.
- Users want to have some kind of reminder to tell them about pending mail.

Using the software

- All the users are interested in exploring additional software.
- Some users want to be able to see what the program is doing, and have control over everything.
- Some users want the software to make decisions for them; they do not want to see any additional interface.
- Users want the program to send out emails in the order they want.
- There should be a reminder for pending mail (if set to display)

4.5 The system

In this section, we are going to discuss the software system designed for this study. The design of the program is based on the result of the Contextual Inquiry sessions and the focus was on two things, how the connection settings should be managed, and how the outgoing emails should be handled. A list of guidelines which the program was trying to achieve was drawn up based on the findings:

- The program should prompt the users to setup a connection to assist in making the email sending decision.
- The program should still work on certain connections if users decide to skip (or disable) the connection setup.
- The SMTP server should be automatically applied for each connection. For connections with no SMTP server, users should be able to try to send mail directly to the receiver's mail server.
- There should be a set of rules per connection that users can set. (e.g. do not send mails bigger than 500KB; send all the emails immediately with this connection, etc)

- Users should be reminded about pending emails.
- It should be possible for the users to arrange emails for sending (by FIFO, Priority, etc).

Based on the guidelines above, we designed the software to handle the outgoing emails for the users. The system consists of four different modules: Connection Manager; Mail Handler; SMTP Sender; and User Interface. The Connection Manager is used to identify connections users connect to. The Mail Handler is used to receive and store mail. The SMTP Sender is used to connect to SMTP servers to send emails. The user interface displays information from different modules, and allows users to change settings. See Figure 8 below.

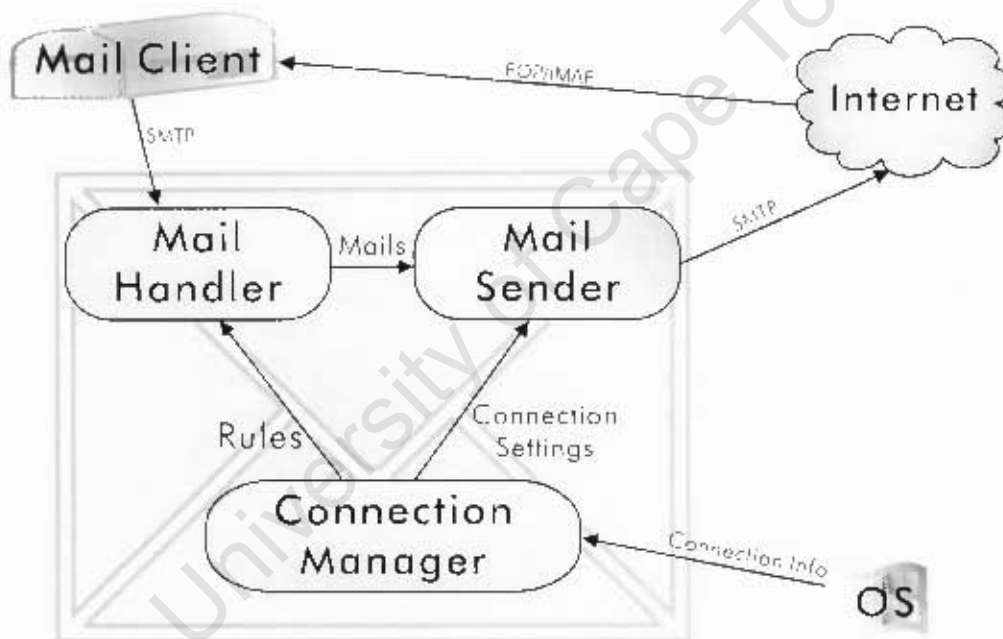


Figure 8: The different modules in the program

4.5.1 Connection Manager

The Connection Manager is the module used to identify different connections. It checks if the user is connected to the Internet, and which connection the user is using. It also provides the connection information of the current connection to other modules which request it. The Connection Manager is also used to store the connection information entered by the users, and allows users to view the connection list, and edit/delete any connection information.

Identify Connections

When the user connects to the Internet via a router, we use the Media Access Control (MAC) address of the router to identify the connection. The MAC address is a unique identifier that appears on all the network devices. It is used to determine where each IP packet comes from, and goes to. The MAC address is the perfect mechanism for identifying the connection when the user is connected to a router. However, this method does not work when the user is connected to a user-established connection. The user-established connections are the connections such as dial-up, PPPoE, VPN, GPRS/3G, etc. When the user connects to this type of connection, the MAC address is always 00-53-45-00-00-00, which cannot be used to identify connections.

The solution we provided to this problem is to identify connections by name for the user-established connections. Connection name is the name of the connection given by the user to identify different connections when a user creates the connection initially. It can be the company name, if the user uses this connection to connect to the company VPN, or it can be the name of the ISP when the user uses it to connect to the Internet, etc. After the connection is created, it will be listed in the “Connect To” list found in the Start Menu. We have noticed that the connection names to the user-established connections are unique per user’s machine. Thus, when the user is connected to these types of connections, we can use the connection name to uniquely identify the connection. With the two methods (MAC address and connection name) working together, we can properly identify any kind of connection the user connects to.

Connections List

The connection information is vital for this program, and it is provided by the users. It needs to be accessible and easy to use and modify at any given time. We decided to store this information in XML format because it is portable and human-readable. The information we store in the connection information file is the following:

- Connection Identifier – This is for the program to identify the connection.
- Connection Name – This is used for the users to identify the connection.
- Connects to Internet – This is a flag to tell if this connection connects to the Internet.
- SMTP Server – This is the SMTP server address specific to this

| | |
|------------------|---|
| | connection. |
| Speed – | This field stores the speed for this connection. |
| Cost – | The cost for using this connection (per MB and per Hour) |
| Mail Size Rule – | This stores the threshold for the allowed mail size, and if this rule is enabled. |
| Priority Rule – | This stores the lowest allowed priority for mails, and if this rule is enabled. |
| Cost Rule – | This stores the maximum allowed cost for sending the email based on the mail size, connection speed, and the connection cost. This field also specifies if the rule is enabled. |

4.5.2 Mail Handler

The Mail Handler is used to handle all outgoing emails. There is an SMTP server running in the Mail Handler to capture emails from the mail client. The emails captured by the SMTP server are stored as text and there is a mail list storing all the emails currently on hold by the program. There is also a mail filter used to filter mail by the rules defined in the Connection Manager.

SMTP Server

For this application, we need a simple SMTP server to interface with the mail client to receive emails from it. We used the C# Email Server (CSES) [3] created by Eric Daugherty to act as the SMTP server for this project. It is an open source Email server which provides a SMTP component which accepts incoming mail and allows programmers to process it afterward.

Once the Email is accepted by the SMTP server, it is stored in a location specified by the user. The filename of the email is the date and time the mail is sent. We used the Coordinated Universal Time (UTC) to milliseconds to make sure each email will have a unique file name. An entry to the email information is then created and kept in the program.

Email List

Like the connection information, the email information is also stored in an XML file. This information is removed from the XML file once the mail is sent. The

following is the information stored in the Email list XML file:

| | |
|-------------|---|
| Filename – | This is the path to the email file. It is also used as the identifier to each individual email. |
| Subject – | This is the subject for the email. This is used for the users to identify the mail. |
| Size – | This is the size of the email. This is used by the mail rules to determine if the mail should be sent or not. It is also used for the mail orderings. |
| Priority – | This is the priority of the email. This is used to determine if the mail should be sent or not and for send orderings. |
| Send Time – | This is the time the program accepts the mail. This is used for send orderings. |

Mail Filter

There are two different scenarios where the email sending process will be commenced: it can be triggered at a time interval set by the users, i.e. every 5 minutes, or when the SMTP server receives mail. It is also necessary to have an active Internet connection detected by the Connection Manager for the mail sending process to commence. When the email sending process commences, the program will attempt to send emails stored in the application. Before the mail is sent, it will go through a process to sort the emails by the sending order set by the user. It can be ordered by the size, priority or the send time of the email. After the emails are ordered appropriately, the mail will be examined to make sure it passes all the rules set to filter emails. The rules will be provided by the Connection Manager for the current active connection, and only the mail allowed by all the rules will be passed onto the SMTP Sender to send out.

4.5.3 SMTP Sender

The SMTP Sender is the module used to send emails. It can send emails with or without a working SMTP server. When there is an SMTP server available for the connection the user connects to, the SMTP Sender will act as a mail client to send the emails to the SMTP server. We call this the SMTP method. When there is no SMTP server to use, the program will act as a SMTP server to connect to the recipient's SMTP server to send the email directly. We call this the direct method. The main difference between these two methods is that the

second method tries to figure out what SMTP server to use based on the recipient's email address, while the first method uses the same SMTP server for all the emails.

Mail Exchange

When a domain name is registered with the Domain Name System (DNS), there will be at least one Mail Exchange (MX) registered with it. The MX specifies the mail server for the mail going to the domain. It is normally used by the SMTP servers to relay emails. When the program uses the direct method to send mail, it will acquire the MX address from the DNS server for each mail recipient, and connect to the MX to send the email to the specific recipient. If there are ten recipients for the email, the process loops through the same procedures ten times to send the email to every recipient.

4.5.4 User Interface

The user interface (UI) is the only module visible to the users. Users use the UI to set their preferences, to monitor emails kept in the application, and to manage their connections. In the main screen, we decided to use the tabbed forms to reduce the number of forms users will see. The following are the forms that exist in the program: the main form, the selection form and the new/update connection form. For the purpose of this program, the close buttons on the forms does not close the application. The user will need to use the menu item in the system tray icon to exit the program. The program was designed this way so that it will not be closed by users accidentally when they only want to close the form.

Notice and Information

When the program is executed, it will only be shown in the windows system tray as different icons. The icon will show the status of the program, as in Table 2 below. The user can click on the icon to display the main form, or right click on the icon to display the menu for the program. The menu allows users to display a specific tab, to send pending emails without waiting for the time interval, or to exit the program.





| Icons | Meanings |
|---|---|
|  | There is no email pending in the outbox, and there is no active connection available. |
|  | There is no email pending in the outbox, and there is an active connection available. |
|  | There is mail pending in the outbox, and there is no active connection available. |
|  | There is mail pending in the outbox, and there is an active connection available. |

Table 2: The icons used in system tray

When there is a notification, such as settings change or sent emails to display to the users, it will be displayed for several seconds as tooltip messages in the system tray as in Figure 9. Since the message is displayed as a tooltip message, it will not take attention from the program the user is currently using, thus users will not feel bothered by the notify message.

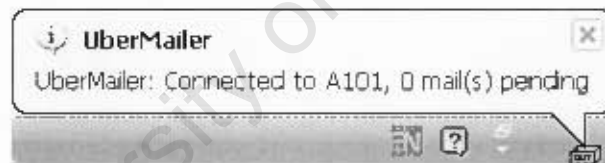


Figure 9: Tooltip Message

The Main Form

This is the main interface to the program. The main form contains five different tabs: General, Options, Connections List, Mails List and About. In the General tab, shown in Figure 10, users can see the status of the application and the number of emails sent during the session. The Refresh button refreshes the general page; the Close button hides the form.

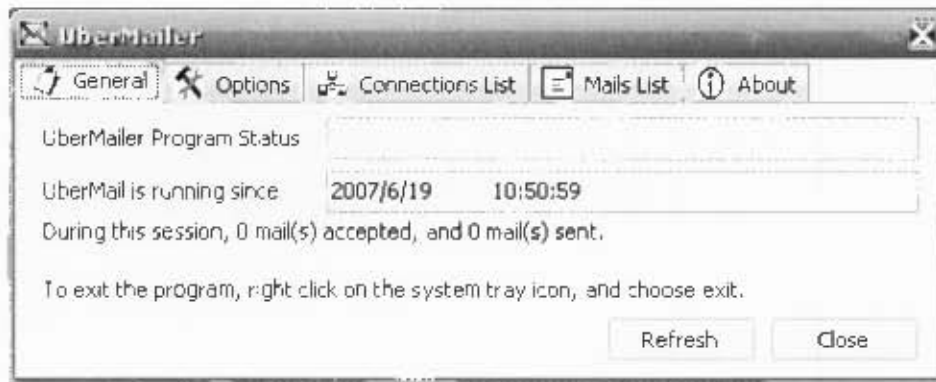


Figure 10: The General tab in the Main form

In the Options tab, shown in Figure 11, users can set the directory in which the emails will be stored, the order in which outgoing emails will be sent, the interval which the program tries to send emails, and the option for users to start the program when the operating system boots up. The Save button saves the change to the Options tab, and the Cancel button is used to set the setting back to the previous settings. The user's settings are stored in the settings.xml file. The only exception is the "Start with Windows" selection which is stored in the registry as a necessity to start the program with Windows.

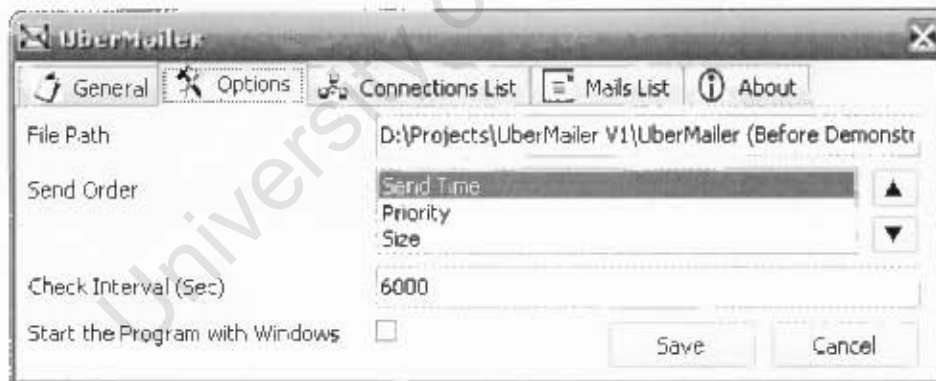


Figure 11: The Options tab in the Main form

The Connections List tab contains a list of all connections the user has ever connected to, as shown in Figure 12. Users can modify or delete selected connections. The "Identify Current Connection" button is used to detect the current active connection. When it is pressed, the current connection will be highlighted. The Mails List tab (Figure 13) stores a list of emails still waiting to be sent out. With the selected emails, there are buttons allowing the users to view the recipients; send the mails ignoring the rules; or to delete the mails. The About tab (Figure 14) contains the author and version information for the

application.

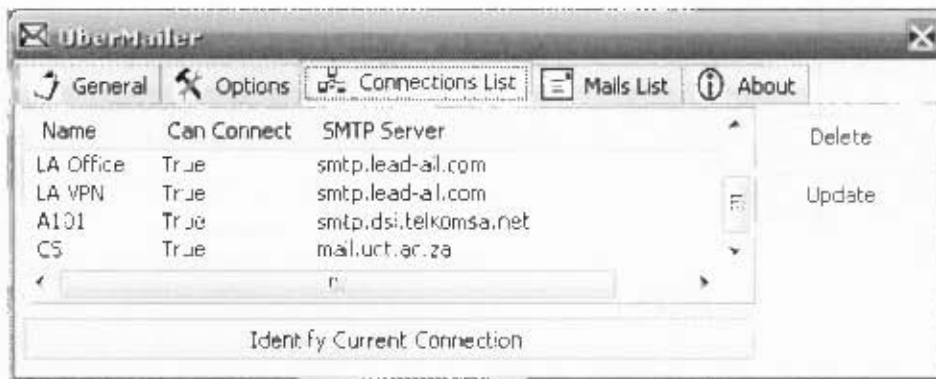


Figure 12: The Connections List tab in the Main form

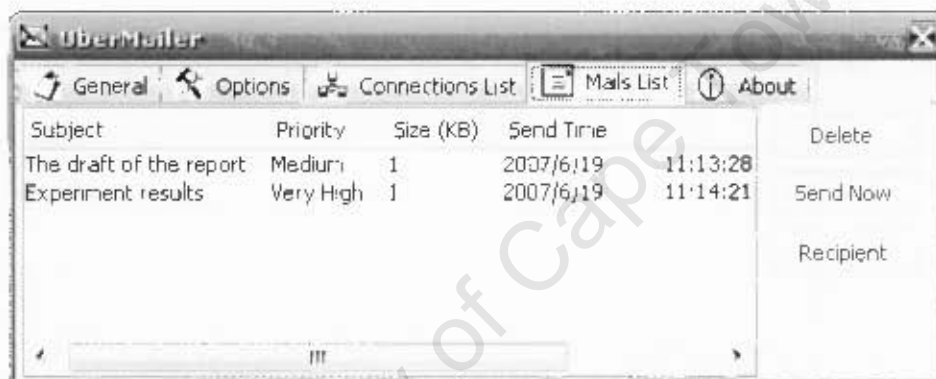


Figure 13: The Mails List tab in the Main form



Figure 14: The About tab in the Main form

Selection Form

The Selection Form is used when it is necessary for users to make decisions about ambiguous situations. The form can have up to four different options (see Figure 15). The selection form is designed to be a generic form which can be used for any kind of selections required by the program. But for this iteration, it is only used when a new connection is detected. When a new connection is

detected, users can choose to set up the connection; to temporarily send mail without setting-up the connection; to discard the notice for the selected connection; or to snooze the new connection notice, using the cancel button.



Figure 15: The Selection form

New/Update Connection Form

When a new connection is detected and the user decides to set it up, or when the user updates a selected connection from the main form, the new/update connection form will be displayed, as shown in Figure 16. This form is used for the users to set up connections. In the connection form, the user can set all the information and the rules related to the selected connection. The information displayed in the connection form is the information stored in the Connection List xml file. The only exception is the Connection Identifier. It is hidden from the user, because it is not necessary for users to view such information.



Figure 16: The New/Update Connection form

4.6 Demonstration Results

In order to evaluate this iteration, we demonstrated the alpha version of the software to the group of users with whom we conducted the contextual inquiries sessions. During the demonstration, the Conceptual Model Extraction technique was used to find out how users interpret the interface. It makes sure the program interface is clear enough for the users to use so future assistance from the researcher should not be needed. The feedback from the demonstration of the alpha version is as follows:

- Based on the Conceptual Model Extraction results, all the users are able to understand the interface which includes the menus and the messages displayed by the program. However, most of the users believed the close button on the main form should close the program, instead of just hiding it from the users.
- In the options window, users expect the “Save” and “Cancel” buttons to close the form, instead of just saving the changes or reloading the settings from the system.
- Users believed that the tabs in the interface should be reordered. The mail tab contains the most crucial information which users want to see (is the mail sent, which mail is still trapped in the Outbox, etc), thus it should be the first tab users see. The options tab, which is the least frequently used tab, should be placed just before the about tab.

These alterations will be made to the next release of the software.

4.7 Iteration Conclusion

In this iteration, we performed the Contextual Inquiry with the users, and developed a program to simplify the procedures for sending emails from different connections. After the implementation of the program, it was demonstrated to the users to make sure the interface is intuitive enough for the users to use.

The result of the Contextual Inquiry shows that all the users chosen fit the criteria of the target users. When users are travelling, they either send mails with a wireless connection (3G/iButst/Sentech) or they use the web-based

interface on other peoples' machines. All the users are keen to participate in the experiment, and are willing to use the additional program to handle their mail as long as they can control everything the program does.

During the demonstration sessions, we noticed some strange behaviour. Users complained about how a certain part of the interface works, but appeared to have no problem working with it. For the icons displayed in the system tray, two users complained that the "no connection" icon confuses them because it is not clear enough. But in the mean time, they had no problem telling the researcher that the icon meant there was no connection. Another user argued that the interface for arranging the sending order in the options tab was not easy for users to operate, but he had no problem arranging the send order according to the researcher's request. This behaviour illustrates the need to conduct observation-based evaluation, rather than relying on just interviews, when conducting this type of research.

Alterations to the program were done based on the users' feedback and the enhancements the users suggested. The alteration process is covered in the second iteration of this study, and is discussed in the following chapter

5 Iteration Two

In the second iteration of this study, the design of the software was altered according to the demonstration results we gathered during the demonstration sessions. This version of the program was used for the first long-term experiment which lasted six weeks. The aim of this iteration was to find out if the altered design of the interface suits users' needs.

5.1 Questions

The following are the questions we addressed in this iteration:

Modification to the program

- What modification was needed according to the feedback?

The use of the program

- Did the application solve the problems users have in sending emails from different locations?
- Would the users be able to use the application without any difficulties?
- Were users willing to rely on this application to send all their emails?

5.2 Procedures

The program was altered and tuned according to users' feedback during the demonstration sessions. There was also an activity logging module implemented in this version to support Log Analysis of the interaction. The log file was used to analyse users' activity. The first release of the program was installed on the users' laptop computers for the experiment. Most of the users in the sample group used the emailing program for six weeks. The activity log file was analysed by the researcher to understand what options the users use more frequently and how they were used.

Interviews were conducted after the log analysis to find out the users' experiences with the program. The interviews investigated topics like: Do

users find their way through the program? Does the program solve the problem for them? Do they find the program useful? Do they use the program all the time? Users' abnormal behaviours discovered during the log analysis were also questioned during the interview sessions. The results for the interview sessions were used to implement the final version of the software system.

5.3 Alteration to the Program

During the demonstration, users made recommendations on how the program should be altered to make it easier to use and more suitable for its purposes. Most of the changes took place in the program were visual changes. In the following section, we will discuss the modifications in detail.

5.3.1 Interface Enhancements

The following are the interface enhancements made to the program. These changes were made in order to increase the usability of the program:

Close Button

- The close button in the main form was used to hide the program. However, most users believe the button should be used to close the program instead of just hiding it. The main form was modified in such a way that there is now a minimise button and a close button in the title bar. The minimise button will hide the program in the task bar, and the close button will terminate the program. Because this program needs to be running at all times to ensure users' emails will be sent properly, a warning message will be displayed when users attempt to close the application.

Tab Orderings

- Based on users' suggestions, the Mails List tab was moved to the beginning of the tabs, as the default tab users see when the main form is loaded. The Connections List tab and Options tab became the second and third tab in the form. The General tab and the About tab were merged and displayed as the last tab in the main form because users care least about that information.

Mails List Tab

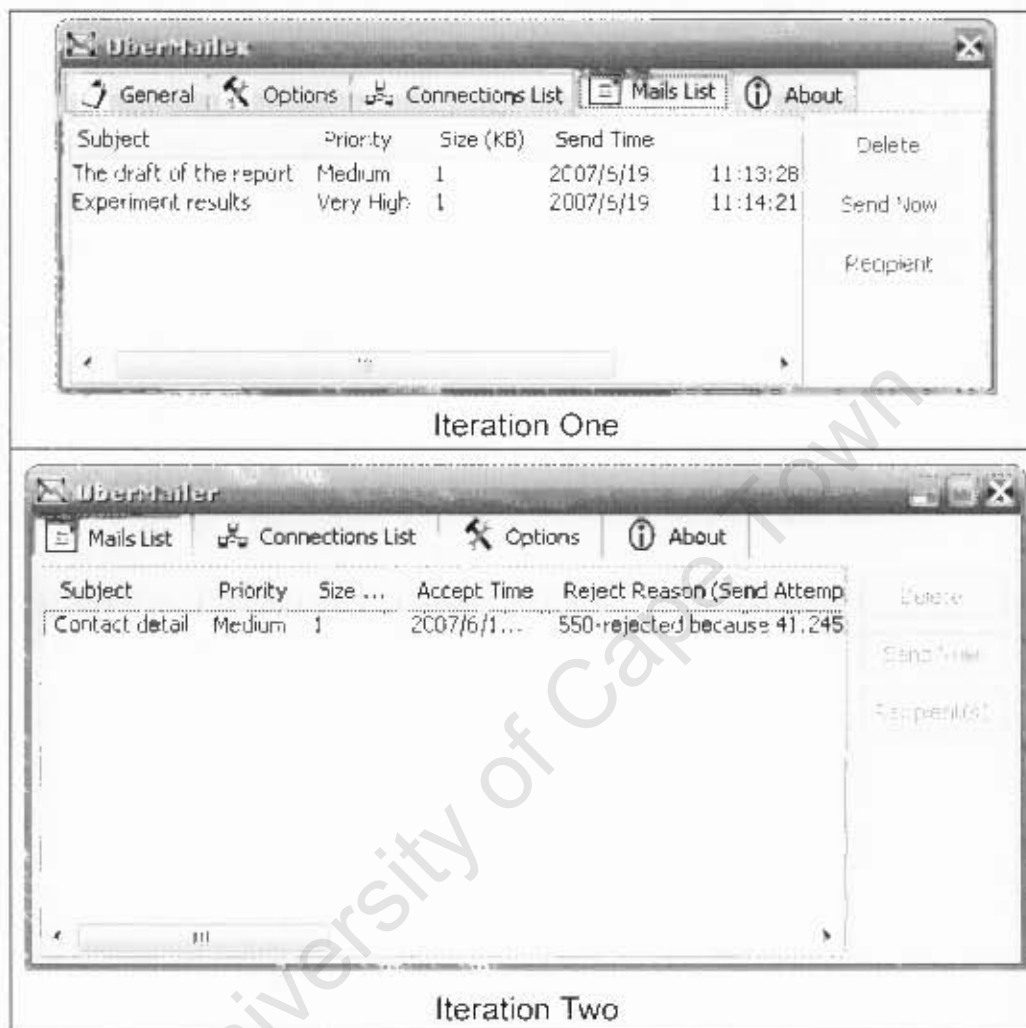


Figure 17: The Mails List tab for iteration one and two

- The "Send Time" field is renamed to "Accept Time". Users argued that the name "Send Time" implies that the mail is sent, and the time stated in the field looks like the time the mail was sent out. Thus this field was renamed to "Accept Time" because the time displayed in this field is the time the mail was accepted by the program.
- In the alpha version of the program, the mail list allows only single selection for sending or deleting emails. The list selection is now modified to allow multiple selections. Users can select different emails in the mails list with the same method they select files in the Windows Explorer.
- A new field "Reject Reason and Send Attempts" was added to the mails list. It is used to display the number of attempts the program has made to send out the specific email, and the error message for the mails failed to

send out.

- When an email is deleted by the user from the mail list, there is no record of the deleted mail. Thus in this version of the program, a notification email will be sent to the user to notify them about the deleted email when an email is removed by the user.

Connections List Tab



Figure 18: The Connections List tab for iteration one and two

- The button "Identify Current Connection" is changed to "Select Current Connection" because what this button does is to highlight (select) the current connection. Thus the new name is more appropriate for its feature.
- The "Can Connect" field in the connections list was changed to "Connects to the Internet". It was renamed because the old name misleads the users to believe that the connections (SMTP servers) marked as "Can Connect" are the ones which the users can use for the current connection. This field

actually means that the connection is setup by the user as one of the connections which connects to the Internet.

Options Tab



Figure 19: The Options tab for iteration one and two

- The label "Check interval" is changed to "Send interval". The value in this field determines how often the program will try to send out mails still kept in the program. This change makes this field clearer to users.
- The Send Order field was not very clear to the users. A description was added to the selections to eliminate possible confusion. The description will make the values easier to understand, to prevent possible confusion.
- In the new version of the program, the Save and Cancel buttons were both removed from the options tab. In the new version, when the user leaves the Options tab (move to other tabs, minimise the form, or close the program), a confirmation message will be displayed to find out if the user

wants to save the change or not.

About Tab

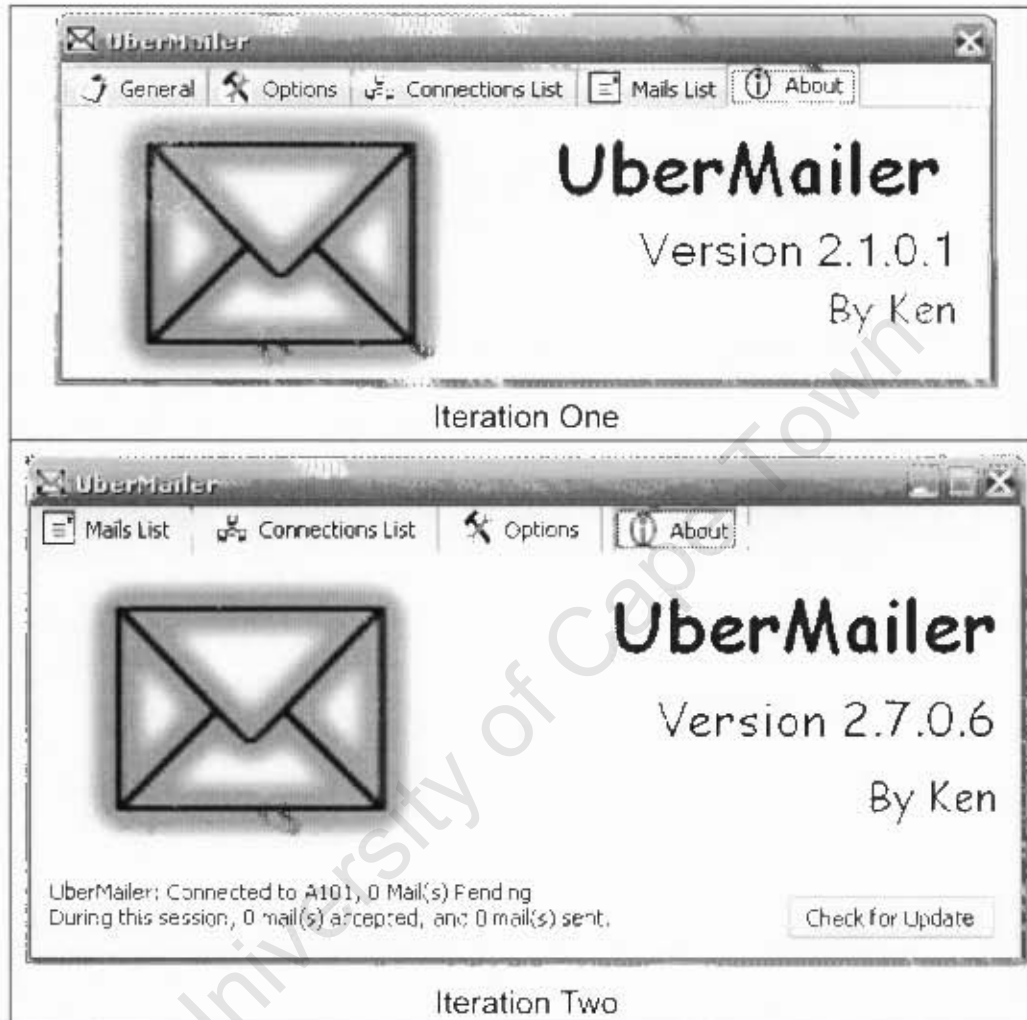


Figure 20: The About tab for iteration one and two

- The Refresh and Close buttons that used to be found in the General tab were removed from the program. The Close button is not needed anymore because users can perform the close action from the title bar of the program. For the Refresh button, the program now refreshes the status of the program automatically, which makes this button redundant.
- A new button "Check for Update" is added to the program. This button is used to connect to the server to check for new versions of the program. This feature will be discussed in detail later.

New/Update Connection Form

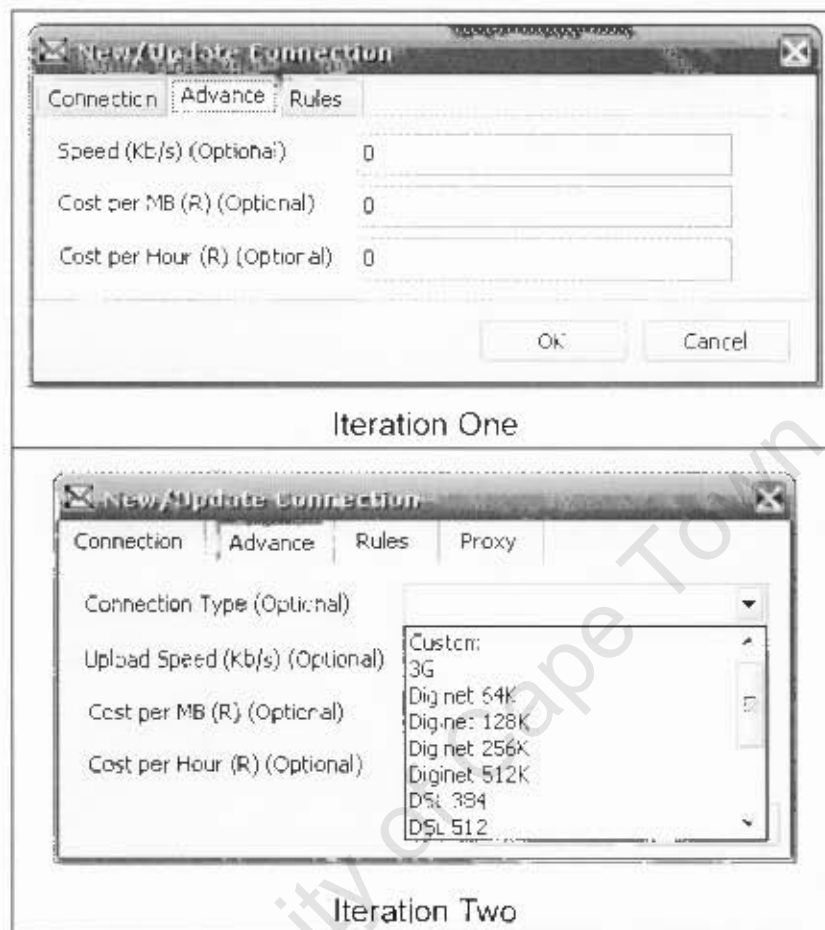


Figure 21: The Connection Types in the New/Update Connection

- In the New/Update Connection form, users can now select the connection type when they update connections. When a connection type is selected, the speed and cost information are entered automatically. This enhancement was made to simplify the connection settings. Users do not need to know detailed information for the connections anymore.

5.3.2 New Features

Program Updater

To ensure the experiment will run smoothly, an updater was added to the program to check for program updates. The updater connects to the Internet to compare the software version between the local copy and the copy available on the website. If the version number is different, the new version on the website is downloaded to replace the local copy.

The file downloaded from the website will be a compressed file. In order to make sure the downloaded file is not corrupted, the md5 checksum for the file is evaluated. If this checksum does not match the checksum recorded on the website, the update process will be terminated, and the user will be notified about the error.

Once the file is downloaded and md5 checksum matches, the updater will terminate the main program, replace the old files with the new ones, and the program will be executed again. By then, the update process is completed.

Log Recorder

We have decided to log users' activities during the experiment so we can perform log analysis. Thus, a log recording module was added to the program. The Log Recorder records the activities in the program such as connection change, mail sending attempt, options change, etc. The log records the time the activity occurred, the type of the activity, and the description for the activity. The log file was sent to the researcher as an email on a weekly basis.

Proxy Server Changer

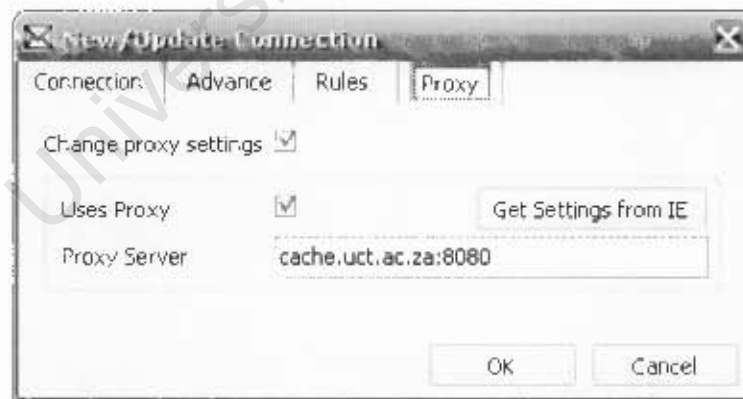


Figure 22: The Proxy settings

Since this program can be used to detect different connections, we added the feature to change the proxy settings for Internet Explorer and Mozilla Firefox browsers according to the connection the user connects to (see Figure 22). We were curious to find out if the algorithms we used to detect different connections can be used by other applications. It was later proved useful for users who need to setup proxy servers for different connection since this

program can change the proxy settings for the users automatically when they connect to a different connection.

5.4 The Experiment

For this research, we performed long term experiments in order to evaluate the program in a realistic context. In this section, we will look at the objectives of this experiment, and the procedures followed to perform the experiment.

5.4.1 Objectives

- To determine if the program can solve the problems users have in sending emails from different locations.
- To determine if the users will be able to use the program without any difficulties.
- To determine if users are willing to rely on this program to send all their emails.

5.4.2 The Procedures

The application was first tested by a group of mobile users. This group of three users were the pilot group for this experiment. The pilot group used the application for one week to make sure the application is stable enough to undergo the actual experiment.

For the actual experiment, the application was installed on five users' laptops (which include the pilot users) and the application was set to start up when the users turn on their laptop. Users were notified about how to set up their mail client. They were also shown how to bypass this application in case any of the users decided to stop using the application at any point in the experiment. From the study of the application demonstration, the users thought the interface of this application is quite self-explanatory, therefore only a short tutorial was given to the users about the program.

The log file was sent to the researcher every week to analyse users' behaviour. By studying the weekly usage pattern, it was possible to determine how users use this application. This includes how often the users interact with the

interface, what settings users prefer to use, do users rely on the application to send their emails, how long is email normally trapped in the application, etc.

The users used the application continuously for six weeks. Thereafter interview sessions were conducted with all the users in the experiment. The interview sessions were also used to discuss users' activities recorded by the log file, and asked for users' explanation for all any abnormal behaviour.

5.5 Experiment Results

To evaluate the results of this experiment, two techniques were employed. In this section, we will look at both the log analysis findings and the feedback from the interview sessions.

5.5.1 Log Analysis

In this section, we will discuss the findings from the log analysis. We also include the feedback from the interview sessions where it is related to the specific log findings:

- All the users do connect to more than one connection (2-9). Thus all the users chosen are valid users. Users send between 15 and 150 mails per month (including the log files).
- The difference between the number of emails the application receives per month and the number of emails the application sends out per month is under two emails in most cases – this difference is because users decided to delete those emails. The biggest single difference was nine mails in a month and was due to the fact that the user entered the wrong SMTP address. The difference rate dropped as this problem was corrected.
- The users open the application to check the Mail List after they send emails from their mail clients. The users do this in order to see the mail appear in the Mail List and disappear again. This behaviour appears for the first 5-10 days during the experiment. As we try to clarify this behaviour during the interview sessions we found the following:
 - When users first start using the application, they want to find out what

the application does, what features it has, and how it manages emails.

- The users stopped doing this when they think they know enough about the application and believe that the application does work properly.
- Although this application enables the users to send mail asynchronously, the users still tend to send mails only when they are online. The log shows that some users connect to the Internet before mail was received by the application. On average all of the users send less than one e-mail when they are offline every month. The interview sessions revealed the following:
 - Users either do not send mails offline, or they only send very few mails offline.
 - When users do send mails offline, they let the application handle it, so the mails will go out automatically.
- When the users are connected to a new connection, the application will prompt them to setup the connection. Some users know what SMTP Server to use, and enter it during the connection setup. Some users do not try to figure out which SMTP server they should use for different connections. They either put in a SMTP address which they know works for another connection, or they just dream up a SMTP address. In rare cases, users do leave the SMTP address blank, to let the application send mails directly to the receipts' mail server. What we have discovered during interview sessions about users' SMTP understanding and how they setup unknown connections were the following:
 - Although some users do not know the reason why some SMTP servers might be connection specific, they are all aware of this fact.
 - Most users do not know that they can leave the SMTP server field blank, to send mails directly. The application should inform the users about this feature.
- According to the beta version demonstration, all the users believe the software should close when the "Close" button is pressed. But all users seem to press the "Close" button by mistake. However, this behaviour became rare as the experiment proceeded because users interact with the application less frequently. What we have found from the interview

sessions was the following:

- The “Close” message is not annoying, but it will be better if users can set the default behaviour for the button.
 - Otherwise, the application should just hide when the user clicks “Close”.
- The users use the “Send Now” button to send mail. This happens when the mail is busy sending out or when the mail cannot send out (an SMTP server is not accessible). In most cases, the use of “Send Now” was not necessary. In no cases did it happen because users wanted to bypass the rules. In the interview sessions, we asked the users what they think the “Send Now” button is for, and in what situation do they use it:
 - Most of the users know what the “Send Now” button does. However, they used it because they do not know what the application is doing.
 - Therefore, the users should be informed about the mail/application status at all time, in a way that is clear and easy to understand.
 - Mailing rules is one of the key features for this application. However, only one user uses the mailing rules. This user uses the rules to block mails with low priority and mails whose size was greater than 50K on his GPRS and dial-up connection. Thus we questioned the users during the interview sessions to find out their thoughts about the mailing rules:
 - Most users do not know about the mailing rules, because they did not even know they exist – despite being told about them initially.
 - The mailing rules is not one of the more obvious settings to the users – the interface should be changed so that the users will be more aware of it.
 - Most users think the rules are useful, and they want to use them in future.

5.5.2 Interviews

We used the interview sessions to find out what the users think about the program, and how they think the program can be improved. The following contains the most common feedback for each question during the interview sessions:

Users' opinion toward the application, their likes and the dislikes:

- The application does work properly, and it solves some of the emailing problems.
- Users like the fact that the SMTP server is changed according to different connections they connect to.
- There is nothing that users dislike.

Troubles users have while using the program:

- One user has problems connecting to the SMTP server provided by his ISP Company. However, he never used that SMTP server prior to the experiment.
- The rest of the users do not have any major issues when they use this program.

Users' application improvement ideas:

- Most users suggested that a list of SMTP servers be implemented for the users to select the right ones to use.
- Because this program should be run at all times, the way the "Close" button works should be changed to close only the user interface of the program instead of closing the actual program.
- The users should be more informed about what the application is doing, and the status of their outgoing emails.

The use of the application after the experiment:

- Most of the users are willing to carry on using the application.

5.6 Iteration Conclusion

In this iteration, we updated the program based on the feedback we got from the previous iteration, and then performed an evaluation with this updated version of the program. From the evaluation, we found that the users believe this program works for them in solving the problems involved in sending emails from multiple locations. However, there are several things that can be improved in the program.

The log file revealed that all the users connect to more than one connection. For the first week of the experiment, users opened the application to see how

the program interacts with the mail clients and SMTP servers. As the users started to understand what the application does, and trusted the application, they stopped checking on the program.

During the interview sessions we were able to confirm some of the users' behaviours logged by the log file which seem strange to us. We confirmed that users do not send mails when they are offline. They prefer to compose/reply to emails when they are online. The opinions users have toward the program are that the program solves some of their mailing problems because it selects the right SMTP server to use. They are willing to continue using the application after the experiment.

University of Cape Town

6 Iteration Three

This is the final iteration of this project. We tried to solve the issues users picked up during the first experiment by redesigning the program according to their feedback. The final version of the program was then subjected to another evaluation similar to the first. Users were then interviewed to find out if the new design is an improvement.

6.1 Questions

The following are the questions we addressed in this iteration:

The program

- What should be changed to make the program better?
- What do the users like/dislike in the previous version?

Users' experiences

- Do the users find this interface better than the previous ones?
- Is there anything users like better in the older version?

6.2 Procedures

The implementation plan for the final version of the program was drawn based on the results of the previous experiment. The program was then modified according to the plan, and released to the users to go through another six weeks of user experiments. At the end of the final experiment, users were interviewed again to find out their experiences of the new version.

6.3 Alteration to the Program

From the previous iteration, we got feedback on users' experiences. This feedback was analysed to discover any shortfall in the program. Based on the complaints related to the program, we revisited the program design for this

iteration.

6.3.1 Mail Sending Status

When users are online and there are emails in the mail list, the users believe that the mails are busy sending in the background. If the mail is stuck in the mails list for too long, users think that an error occurred during the mail sending process, but they are not sure what happened and what they should do to solve the problem. Better indications of the mail sending status will keep users informed. Users are more comfortable when they know what is happening with their emails. In order to make the email status indication more clear to the users, the following alteration to the program was made.

Mail Sending Progress

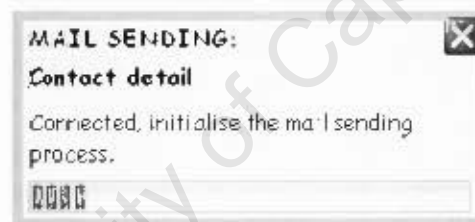


Figure 23: The Mail Sending Progress form

In this iteration, when the program connects to the SMTP server to send emails, a notification window, as in Figure 23, will show to reflect the mail sending status. The notification window also includes a progress bar to show the progress of the outgoing email. The progress bar is used because it provides a better visual representation of the mail sending speed and percentage completion. By looking at the speed the progress bar increments, users will have a better idea about how long it will take for the email to be sent out. Also, users are familiar with progress bars because they are used in the mail clients for sending/receiving mails, and are also used in Windows when users copy or move files.

Mails List

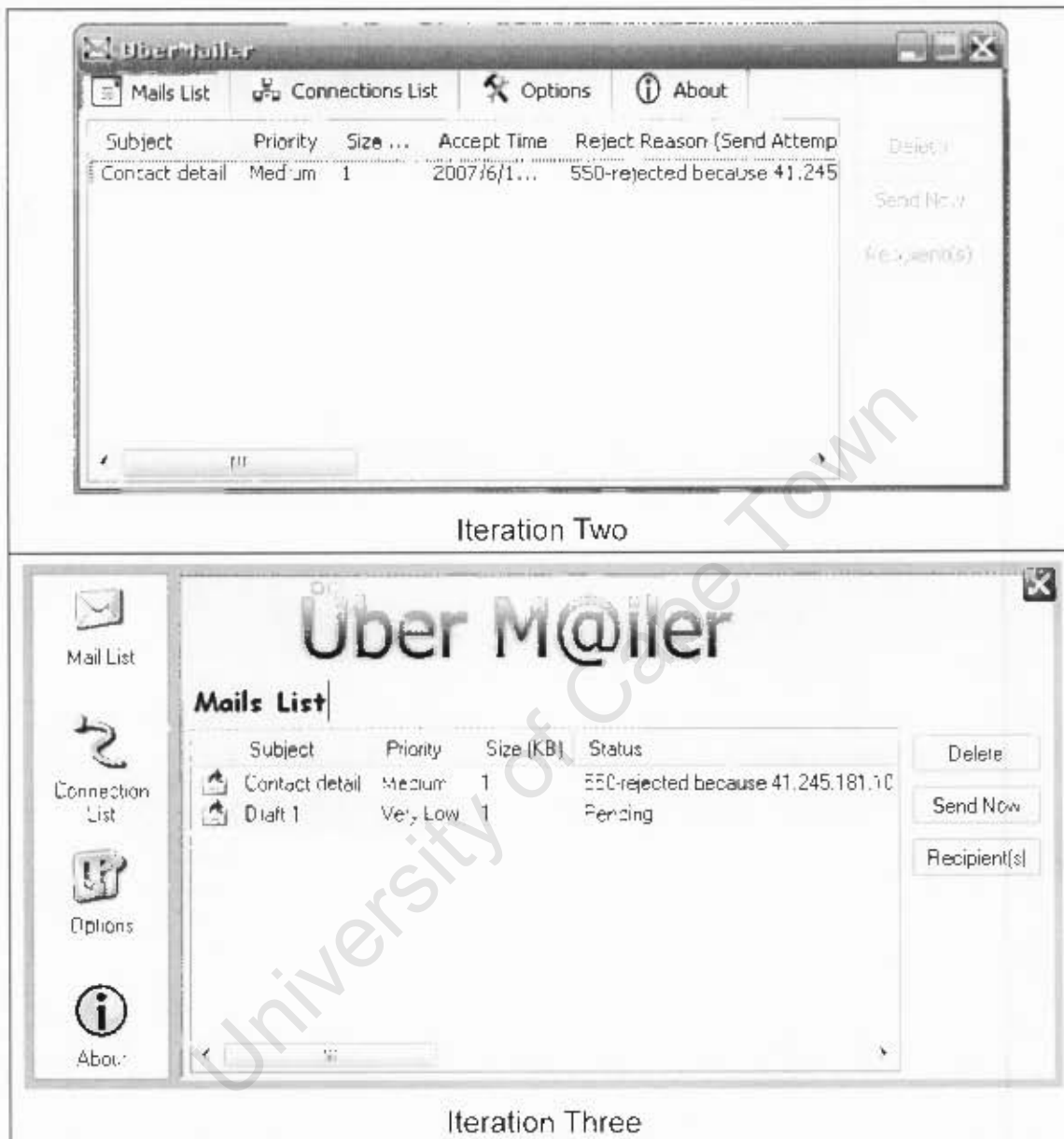


Figure 24: The Mails List for iteration two and three

In the Mail List, a graphical representation of the blocked emails, queuing emails and emails failed to send out was added, as in Figure 24, to make it easier for the users to check the status of their outgoing mails. The status field for each mail will now show the status of the mail (sending, pending, etc) instead of only the error messages and the sending attempts as in the previous iteration.