Electronic mail from a user perspective: Problems and remedies

Olle Bälter

Interaction and Presentation Laboratory (IPLab)
NADA, KTH
S-100 44 STOCKHOLM
SWEDEN
balter@nada.kth.se

November 1995, 2nd printing

Abstract

The aim of this thesis is to describe problems encountered by users of electronic mail (email) and the development of an interface to electronic mail, CoMail. Characteristics and requirements of email programs are identified, mostly functional requirements from the users' point of view, but also some technical points. The opinions of the users are based on a survey with 28 email users at a university site, in depth interviews with twelve users at different sites, and formal and informal user studies of CoMail.

CoMail is one of the tools in the CoDesk environment. It has been developed with iterative design and is intended to be a user friendly direct manipulative graphical interface to electronic mail. The user studies especially point out the need for better support for organising messages, handling email addresses and a solution to the problems that characters such as the national åäö often are distorted. Notably it is important to find strategies for processing the incoming messages, so that the number of messages saved in the folder for incoming messages is kept limited. This can be achieved by better support to delete and undelete messages and an advanced address book that gives the user possibilities to delete messages that today are stored just in order to save the addresses. Suggestions for remedies are given for most of the identified problems.

Acknowledgements

First of all I would like to thank all the people that participated in the survey and the interviews for the valuable time they gave me, with little hope of receiving anything in return.

I am also grateful to:

My supervisors Yngve Sundblad and Kerstin Severinson-Eklundh for their scientific guidance and for improving my English during their thorough reading of drafts of this thesis.

My colleagues Björn Eiderbäck and Per Hägglund for many valuable discussions on object orientation, Smalltalk and improvements of CoMail.

My room mate Thierry Reignier for discussions around CoMail, implementing the waste basket, and courageous attempts of teaching me French.

Ann Lantz for an inspiring, fruitful and entertaining cooperation during a whole year. I have learnt a lot from you.

Lars Kjelldahl for putting up with running CoMail in a terminal window on a Macintosh, despite the slow performance.

Konrad Tollmar for valuable input and comments to the C++-version of CoMail.

The computing facilities group at Nada, especially Peter Svanberg and Patrik Fältström, that has provided me with many insights in the technical problems connected with email.

Finally I thank my Katarina for still being there with a smile when I dragged myself from the work all these late evenings and early nights.

Table of contents

1 Introduction

This thesis describes and analyses problems that users of electronic mail (email) encounter. The study is based on an email survey with 28 respondents, twelve interviews with email users, and user studies during the development of CoMail, our interface to electronic mail.

Email is used for person-to-person communication all over the world. Its use is growing rapidly and soon everyone with access to a computer will have access to email. New groups of users will put new demands on the programs that handle email. The increasing flow of email creates other demands.

The general question treated in this thesis is “What factors influence the use and usability of email?”. The main issues investigated are: impact of a large amount of messages, user interfaces to email, technical problems the users have with email, and storage of email. The purpose of this study is to find general ideas of possible improvements and extensions of CoMail and other mail tools.

1.1 Specific qualities of email

Email is used for asynchronous communication. In [Palme 1995 b] the social, ethical, economical and legal aspects of email are described as well as email standards, functions, techniques and research. In this thesis, email is characterized with five important features: It is *fast*, *non-intrusive, direct, computer processable,* and can be *equalising*.

Email is fast

The time to transmit an email message between sender and receiver is normally less than a minute between nodes (computers) with fix connection to the same local, wide-area, or global (Internet) network. To nodes that are connected via intermittent (a few times a day) phone modem calls, it might take a day.

Email is non-intrusive

The asynchronousness in email allows the sender to write the message whenever there is time. The receiver can read the message whenever the receiver has time. There is no need for the sender and the receiver to be close to a connection such as a phone at the same time. Since email spans across time-zones, cooperative work between people that are geographically or temporally separated is simplified.

Email is direct

Each email message is directed to an individual person or several persons. If the email message is correctly addressed, it will reach the intended addressees and no one else. Phones in offices are often shared. Typically 75% of the phone calling attempts fail [Bair 1979]. A fax machine can be shared by a hundred people. A person that moves around and uses different computers can still read his messages. Email fills a need not covered by other media: Messages can be sent to large groups within minutes [Palme

1995 b]. With TV and radio it is not (yet) possible to direct the information to selected receivers.

Email is computer processable

A text transmitted via a fax can be read, but cannot directly be edited with computer-based editing tools without extensive postprocessing with Optical Character Recognition (OCR) or retyping. When the text is transmitted via email, the text can immediately be handled in a text editor or word processor. Pictures can be handled by image editors. The growing market of multimedia demands audio and video in a computer processable form and with email the distribution can be done without the loss of quality that follows transformations to and from analogue media.

Email can be equalising

Earlier studies have shown that email can give people that have problems with expressing themselves on the phone new possibilities [Sproull & Kiesler 1991]. Sproull and Kiesler show that people may start to communicate with strangers, and that formal status may becomes less important, for further details see “Communication aspects” in section 3.2. According to Sproull and Kiesler, this can be explained by the lack of social norms for email communication and that the sender is less aware of the receiver (e.g. gender is not as obvious in email as on a telephone).

As a drastic example, the Zapatista rebels in Mexico use email and Internet to distribute their political statements. The Mexican government cannot control the flow of information once it has been sent into the Internet [Communications of the ACM 95:4].

1.2 Problems in using email

Informal interviews with users reveal that there are problems connected with the use of email. The main problem areas treated in this thesis are the following.

Problems caused by a large email flow

The more incoming messages, the more time has to be spent to read messages. It is harder to search among many stored messages than a few. It takes time to write messages and replies. Many mail tools can give a sound or a visual signal when new mail arrives. This can be disturbing and interrupt other tasks, in a similar way as a telephone does. A first guess could be that problems would increase when the flow of email increases.

What do users do when they are overflown with email? Do they give up or do they develop strategies to handle the situation? What kind of strategies? These problems have been studied by e.g. [Hiltz & Turoff 1985] and [Mackay 1988]. The main difference between their studies and this study is that email programs have changed a lot since 1985 and 1988, and that this study not only discusses the users but also the technical and the design problems.

Users’ problems with the interface

There are a large number of email programs on the market, old tools with a simple text based interface and more modern ones with a direct manipulative graphical interface. They differ in many ways but also have a lot of similarities. These differences will of course affect the use. How much can a graphical direct manipulative interface help compared to a pure command language interface? How does the interface affect the handling of email?

Users’ awareness of functionality

Some mail programs have a rich functionality, but are the users aware of how to use it? Do the email programs solve the users’ problems in an acceptable way or could more be done to help the user cope? What happens when a person changes from one email program to another? Are the users aware that they can send multimedia email?

Problems for users to organise their email

Inexpensive disk memory allow email users to store most or all of their email. When many messages are stored, the screen is no longer large enough to view all messages and the messages must be organised in some way. Is the support for storing, searching and retrieving adequate in the mail tools?

Problems that are specific for different work categories

Different categories of workers have different tasks, which may affect their use of email. At an academic department, the computing facilities staff typically receives much more messages than the research students and the researchers and from different sources. Are the messages handled differently by different groups of users?

1.3 CoMail

The purpose of CoMail was initially to provide CoDesk, our user tailorable environment for computer supported cooperative work, see section 5.1, with an interface to electronic mail. The development started with informal interviews of the secretaries and the computing facilities staff at NADA. The input from those interviews, and discussions with the IPLab group were used as a basis for the first version of CoMail.

With a working version of CoMail guest researches and interested colleagues visiting IPLab have tried CoMail and have provided valuable input. A formal user study has been made by two computer science students as a part of their Master’s thesis [Hellgren & Olsson 1995].

Three colleagues have used CoMail as their only mail tool for a period of between one month and six months. They found errors and their complaints and suggestions are the reasons for many improvements of CoMail.

The development of CoMail gave insight into some of the technical problems a developer of an email program must solve. This insight was used as input in the questionnaires and the interviews.

The iterative development of CoMail is still in progress. Most of the ideas and the conclusions from the survey and the interviews described in chapter four are not implemented yet.

1.4 Structure of this thesis

Chapter two describes some technical aspects of email and email programs, especially functionalities and programs used in this study. Chapter three describes related research about email that has influenced this study, including the development of CoMail. Chapter four describes a questionnaire survey and interviews with email users. Chapter five describes the development of CoMail and the collaborative environment around it. The thesis is concluded in chapter six with conclusions and suggestions for extensions.

2 Technical background

The focus in this thesis is on email from the user's point of view, but some technical background must be described. Email can be used in many different environments. In this thesis the focus is on email via the Internet. This chapter starts with a description of the technical practices that may affect the usability in 2.1 and continues with a description of MIME and multimedia email in 2.2. The email systems used in this thesis are described in 2.3 The chapter is concluded in 2.4 with a description of some functionality in email systems and different implementations of these functionalities.

2.1 Technical practices that affect email usability

Everything on a computer can be sent via email to another computer. Ordinary text written with the letters A-Z seldom causes any problems, but other characters, letters, applications, pictures and so on can. The A-Z alphabet works perfectly well in English-speaking communities, but almost all other languages have specific letters which are as important as A-Z. In some cases the differences are small and can be solved with e.g. accents, but in some cases similar letters have different sorting order (e.g. in Swedish the letter Ö is the last in the alphabet). The lack of certain letters can be compared to removing a letter from the English alphabet when using a computer.

These problems can be handled by international standards, see e.g. the Unicode documents [Unicode 1991, 1992, 1993]. For the eight languages and alphabets in the five Nordic countries, see [Nordic 1992]. In Swedish the problem is particular great with the letters å, ä and ö. A short description of common reasons for this follows.

Most computers today base their data on eight-bit units, called octets (or bytes). One bit represents zero or one. An eight-bit-byte can represent 256 different characters. However, historically, normally only seven of the bits were used for storing character data. Also, the computer software development was mostly done in USA, where the characters A-Z often are sufficient for text and names. This caused implementing of protocols for email handling in Internet under the assumption of use of the seven-bit coded character set ASCII, which contains only the letters A-Z.

In the nineteen eighties, several 8-bit coded character sets – some vendor-specific and other internationally standardised – were beginning to be widely used, e.g. in Europe. When such a coded character set is used together with email handling software, written according to the seven-bit specifications in the Internet standards, there is obviously a great risk of malfunction. One common example is data distortion due to zeroing of the eighth bit, see figure 2.1. The problem is already – in theory – solved, through rewriting of the Internet standards, but it will take long before all software has been adjusted to the new specifications.

Original string:	abåäö...
As octets with the coded character set ISO Latin 1, decimal:	97 98 229 228 246...
As binary numbers:	01100001 01100010 11100101 11100100 11110110...
After distortion - most significant bit zeroed:	01100001 01100010 01100101 01100100 01110110...
In decimal:	97 98 101 100 118...
Corresponding ISO-Latin 1 string:	abedv...

Figure 2.1: Data distortion of eight bit characters

Another problem is usage of different coded character sets in different environments. E.g. the character Ö (a frequent letter in Swedish, but also in e.g. German, Hungarian, and Turkish text) are in Sweden coded as the octet values 92, 133, 153, or 214, depending on the coded character set used.

If an email with such characters is sent to another computer, the result may be that these characters are displayed as another character at the receiver. E.g. an ‘Ö’ sent as 92 will be displayed as ‘\’ at the receiver. The octets arrive undistorted, but the receiving computer interprets them in the wrong way. The receiving computer must also be told how to interpret these characters. This problem can be solved for Internet users by consistent use of MIME (see 2.2).

Protocols

When an email message reaches its destination it is stored on a mail server. The mail server can be a designated computer, or a designated part of the disk space. For reading, the message must usually be moved from the sever to a storing space. There are three slightly different ways of doing that. Ordinary file movement (copying and deletion), the Post Office Protocol (POP), and the Internet Message Access Protocol (IMAP). All three move the message from the mail server. File movement and POP move the message (normally) to the home directory on a Unix machine or to the local disc on a personal computer, see figure 2.2. IMAP gives possibilities to move the message to a mail file server, where the message can be read from any computer in the net (by a user with access rights), see figure 2.3. IMAP gives possibilities to handle administration with the mail file server, and thereby can make searching faster.

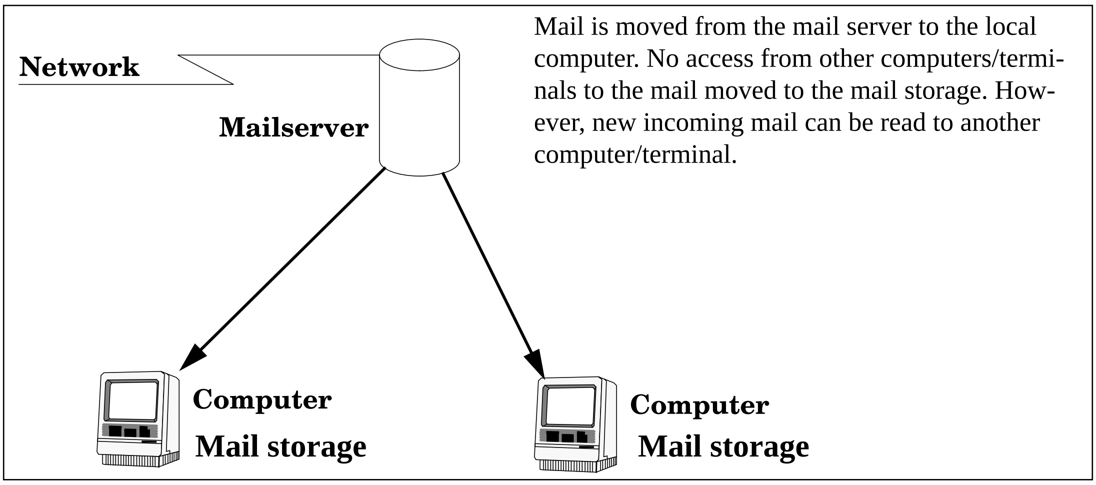

Network
Mailserver
Mail is moved from the mail server to the local computer. No access from other computers/terminals to the mail moved to the mail storage. However, new incoming mail can be read to another computer/terminal.
Computer
Mail storage
Computer
Mail storage

Figure 2.2: The Post Office Protocol, POP.

POP is mainly used on personal computers, when the user always handles the mail on the same computer. With the POP, a person's mail can only be transferred to one single computer. A person that is using two computers cannot directly read the email from more than one computer.

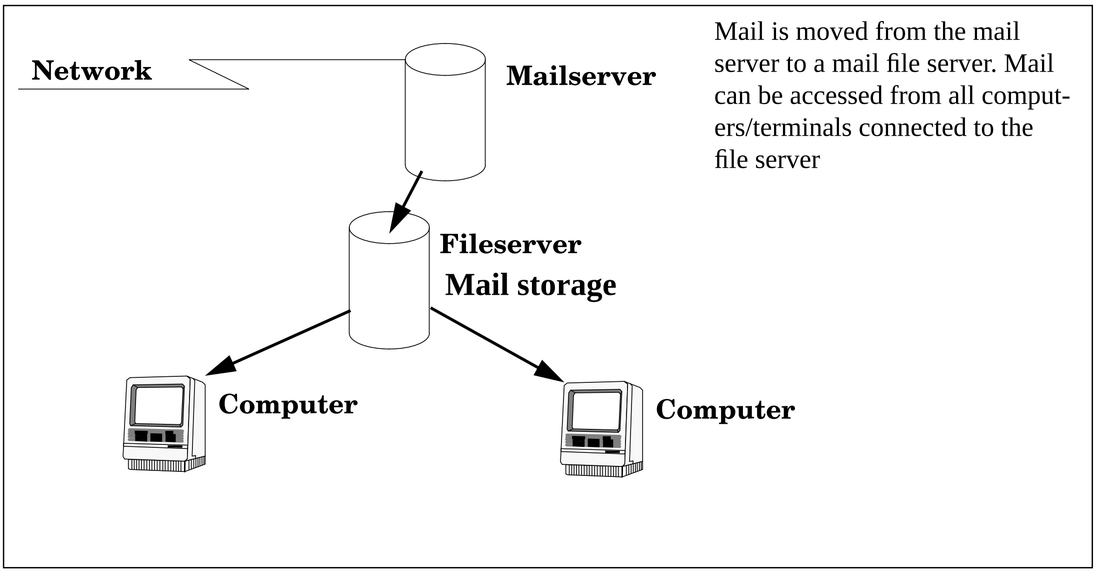

Network
Mailserver
Mail is moved from the mail server to a mail file server. Mail can be accessed from all computers/terminals connected to the file server
Fileserver
Mail storage
Computer
Computer

Figure 2.3: The Internet Message Access Protocol, IMAP.

IMAP can store the messages on a file server that can be accessed from many computers. This is more convenient when several computers are used by the same person.

Both protocols exist in several versions. The protocols used at the moment (1995) are IMAP2 and POP3. The protocol only defines how mail should be transferred. In order to use the protocols there must be a *protocol client* that supports the protocol.

The next version of IMAP (4) is interesting because it will support disconnected use (as POP already does). With IMAP4 it will be possible to connect to the server, fetch the messages, disconnect and start reading, writing, deleting and sorting the messages.

The next time the server is connected the written messages will be sent and the message structure will be mirrored on the server. This liberates the user from constantly being connected and still (in contrast to POP) allows connections to the server from different terminals.

IMAP4 also supports several users that can be connected to the same mail box at the same time. This means that a group can share a common address for mail that is directed to the group and not to an individual.

File and folder structure

Regardless of protocol, the email program can use different ways to store the messages. In an ordinary file system we distinguish between files and folders (directories, catalogues). A folder is a grouping object where files and folders can be stored.

In some email programs this structure is preserved with each message in a separate file. Many email programs have a structure where one file contains many email messages. This simplifies implementation of some functionalities, such as searching, for the email program constructor, but the user cannot in an easy way read the mail with any other program but the email program. In this thesis the structure with many messages in one file is named “mbox” after its name in “mail”, the “classic” text command based Unix email program.

Another problem is that even if two email programs both use the mbox format, they may not be able to read each others' mbox files. The reason is that when program A fetches new mail from the mail server, it adds tags to the message and uses these tags in further processing. Program B does just the same, but uses tags of its own type, which means that B cannot read mbox files from A and/or vice versa.

With a good interface the internal structure of how the messages actually are stored on the computer can be hidden from the user. The resemblance between a file and a message, a file directory and a mail folder can be preserved in the interface.

2.2 Multimedia mail

MIME – Multipurpose Internet Mail Extension is intended to be an “Internet standard” for how to structure email in a standardised way so that the receiving email program should know how to unpack the message and display it in the same way as the sender sent it. In July 1995, MIME had the status of “draft standard”, the final step to a standard is intended to be taken during the winter 1995 [RFC 1800].

MIME handles seven different formats: text, message, audio, image, video, application and multipart (combinations of the formats above). These formats make it possible for the receiver to know what the message contained when it was sent [Borenstein & Freed 1992]. In order to display complex contents, the receiving users’ email program must know how to interpret it.

For example: If the sender uses the coded character set “ISO 8859-1” (also known as ISO-Latin-1), with Ö stored as 214, the recipient’s email program must know what “ISO 8859-1” is in order to display the right character. If the sender uses MS-Word to write a document and sends it as a MS-Word document, the receiver cannot display it without access to MS-Word on the receiving computer. These problems are not solved by MIME.

Together with a user friendly interface, MIME can liberate the user from knowledge of the internal structure of the message, coded character sets and the receivers mail tool. The sender can focus on writing the message and the receiver on reading it, not how to transform it to and from different encoded formats.

MIME gives email users a possibility to send multimedia messages. Some new possibilities and advantages of multimedia messages compared to ordinary text email or fax messages are the following.

1. Formatted text is more pleasant to read than unformatted text (with proper formatting).
2. Pictures can describe things that are difficult or impossible to describe with text.
3. A shared fax is non-private, the message is only black and white, paper and print are often of bad quality, and the information is not (easily) computer processable.

MIME can be used to solve problems with characters outside A-Z for Internet users, but as long as there are non-MIME programs, there must also be possibilities in the mail programs to convert messages with distorted national characters to their original shape. As an example in Swedish, there are situations where åäö can be distorted to characters shown as e.g. {}\, or edv, or \214\212\232, or invisible control characters. For the around 70 different ways of (mis)representing Swedish letters, see [Jarnefors 1994]

As many new standards, MIME has problems of becoming incorporated in all communities. “Interoperability” is a conference organised by the Swedish Network Users Society (SNUS). At Interoperability 1995, only half of the ten participating email programs passed the test for MIME functionality in Sweden [Fältström 1995].

MIME is defined to enable the passing of email messages with embedded pictures, sound and video, but this multimedia mail will also add problems for users with dial-up connections. An uncompressed sound recording of someone reading a full text page in this thesis (approximately 2 minutes recording, ≈ 21 MB) will take about half an hour to transfer via an ordinary 14.400 baud modem. With a 2MB connection, it will take about 90 seconds. Sending email with embedded video demands, in practice, that the connection is made via a coax or fibre cable [Furht 1995]. The file size is of course also a problem when messages should be stored. Hard discs are large, but not infinite.

2.3 Examples of email systems

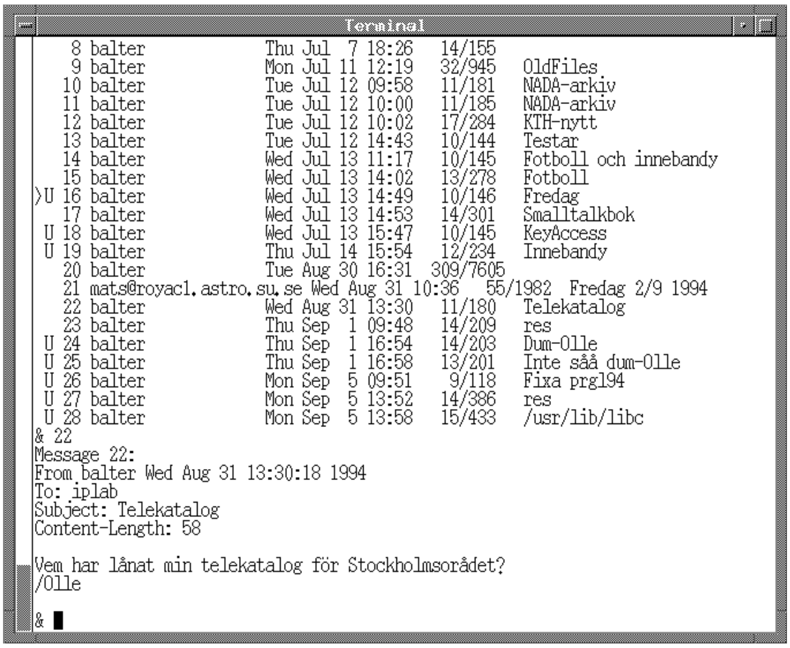
Today there is a large number of commercial, shareware and freeware email programs. They all share some basic functionality such as the ability to read, write and send email, but the differences in how to do that can be great. Here follows a description based on four interface categories with the most simple first and the most advanced last: pure text, text with menus, graphical and direct manipulative. There are other ways to group them, e.g. after multimedia possibilities, but the standard for multimedia is still so new that it is not so useful as a category.

The main differences between the interface categories are described below as well as some characteristic feature of the programs. Within each category examples of systems are given. Most programs below use IMAP. Eudora uses POP and Pegasus runs on a Novell server with similar functions as the IMAP protocol. After the name of the program the version number of the program used in this study is added.

Pure text

Pure text email programs require only a simple text terminal (or a text terminal emulating program) to run. There is no visual support for remembering commands. The number of messages visible at the same time is limited to one or to the screen size (if the messages are short, more than one may be visible). Two typical programs are mail and MH, both Unix based.

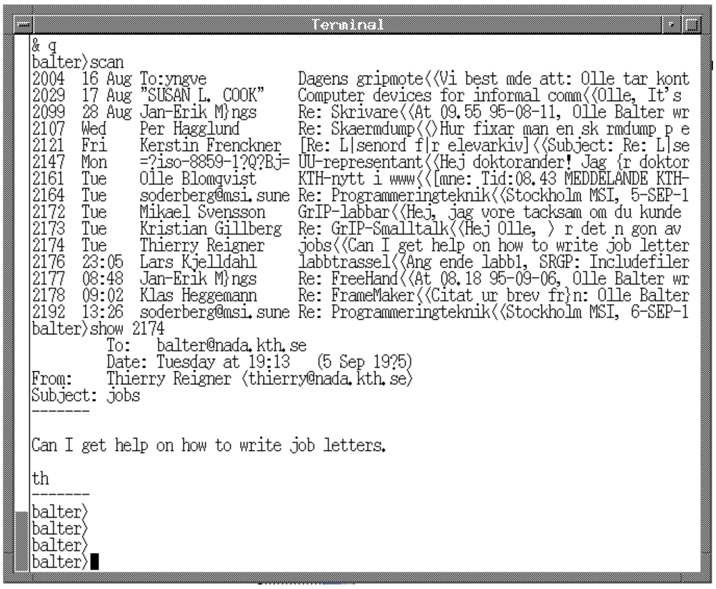
The first, Unix mail, see figure 2.4, is a program where a large amount of (mostly) one-character commands can be used [man Sun 4.1]. Main advantages of Unix mail is that it is simple, fast and does not need to store outgoing messages on a file, which gives possibilities to send email even though the file system is full or out of order.



```
    8 balter               Thu Jul  7 18:26   14/155
    9 balter               Mon Jul 11 12:19   32/945   OldFiles
   10 balter               Tue Jul 12 09:58   11/181   NADA-arkiv
   11 balter               Tue Jul 12 10:00   11/185   NADA-arkiv
   12 balter               Tue Jul 12 10:02   17/284   KTH-nytt
   13 balter               Tue Jul 12 14:43   10/144   Testar
   14 balter               Wed Jul 13 11:17   10/145   Fotboll och innebandy
   15 balter               Wed Jul 13 14:02   13/278   Fotboll
>U 16 balter               Wed Jul 13 14:49   10/146   Fredag
   17 balter               Wed Jul 13 14:53   14/301   Smalltalkbok
 U 18 balter               Wed Jul 13 15:47   10/145   KeyAccess
 U 19 balter               Thu Jul 14 15:54   12/234   Innebandy
   20 balter               Tue Aug 30 16:31  309/7605
   21 mats@royacl.astro.su.se Wed Aug 31 10:36   55/1982  Fredag 2/9 1994
   22 balter               Wed Aug 31 13:30   11/180   Telekatalog
   23 balter               Thu Sep  1 09:48   14/209   res
 U 24 balter               Thu Sep  1 16:54   14/203   Dum-Olle
 U 25 balter               Thu Sep  1 16:58   13/201   Inte såå dum-Olle
 U 26 balter               Mon Sep  5 09:51    9/118   Fixa prgl94
 U 27 balter               Mon Sep  5 13:52   14/386   res
 U 28 balter               Mon Sep  5 13:58   15/433   /usr/lib/libc
& 22
Message 22:
From balter Wed Aug 31 13:30:18 1994
To: iplab
Subject: Telekatalog
Content-Length: 58

Vem har lånat min telekatalog för Stockholmsorådet?
/Olle

& 
```

Figure 2.4: Unix mail with a list of messages and an opened message.

MH 6.8 is a Message Handling system originally developed at RAND Cooperation, then improved and widely distributed by the University of California at Irvine [Rose & Romain 1990]. MH is used directly from the Unix shell, see figure 2.5, and after each command the control is returned to the shell. The MH-commands have mnemonic names like scan (to scan the list of messages) and comp (to compose a new message). Each message is stored in its own file (not like the mbox format) which gives possibilities to handle the mail with other programs, and ordinary Unix-commands. MH has many possibilities to adapt the program to the user, e.g. defining what editor to use to compose or read messages.

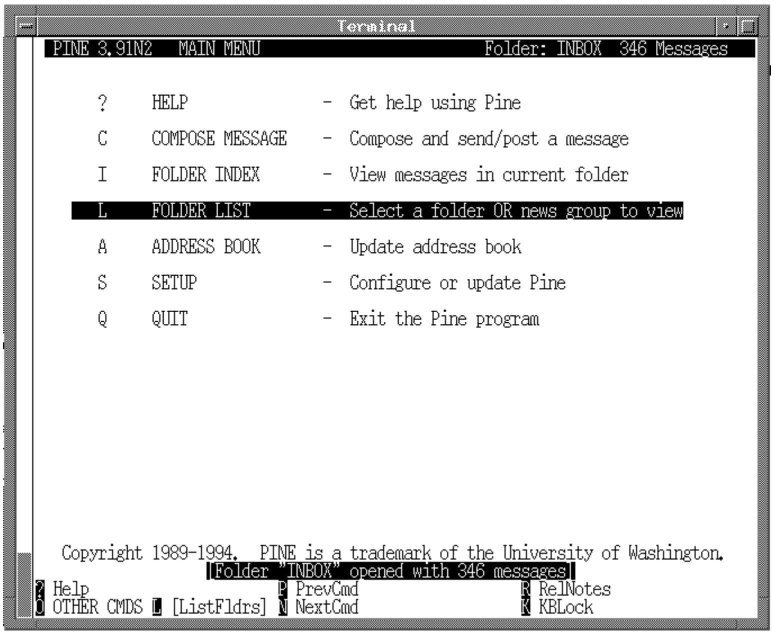


```
& q
balter>scan
2004  16 Aug To:yngve           Dagens gripmote<<Vi best mde att: Olle tar kont
2029  17 Aug "SUSAN L. COOK"    Computer devices for informal comm<<Olle, It's
2099  28 Aug Jan-Erik M}ngs     Re: Skrivare<<At 09.55 95-08-11, Olle Balter wr
2107  Wed    Per Hagglund       Re: Skaermdump<()Hur fixar man en sk rmdump p e
2121  Fri    Kerstin Frenckner  [Re: L|senord f|r elevarkiv]<<Subject: Re: L]se
2147  Mon    =?iso-8859-1?Q?Bj= UU-representant<<Hej doktorander! Jag {r doktor
2161  Tue    Olle Blomqvist     KTH-nytt i www<<[mne: Tid:08.43 MEDDELANDE KTH-
2164  Tue    soderberg@msi.sune Re: Programmeringteknik<<Stockholm MSI, 5-SEP-1
2172  Tue    Mikael Svensson    GrIP-labbar<<Hej, jag vore tacksam om du kunde
2173  Tue    Kristian Gillberg  Re: GrIP-Smalltalk<<Hej Olle, > r det n gon av
2174  Tue    Thierry Reigner    jobs<<Can I get help on how to write job letter
2176  23:05  Lars Kjelldahl     labbtrassel<<Ang ende labbl, SRGP: Includefiler
2177  08:48  Jan-Erik M}ngs     Re: FreeHand<<At 08.18 95-09-06, Olle Balter wr
2178  09:02  Klas Heggemann     Re: FrameMaker<<Citat ur brev fr}n: Olle Balter
2192  13:26  soderberg@msi.sune Re: Programmeringteknik<<Stockholm MSI, 6-SEP-1
balter>show 2174
         To:   balter@nada.kth.se
         Date: Tuesday at 19:13   (5 Sep 19?5)
From:    Thierry Reigner <thierry@nada.kth.se>
Subject: jobs
-------

Can I get help on how to write job letters.

th
-------
balter>
balter>
balter>
balter>█
```

Figure 2.5: MH list of messages and an opened message.

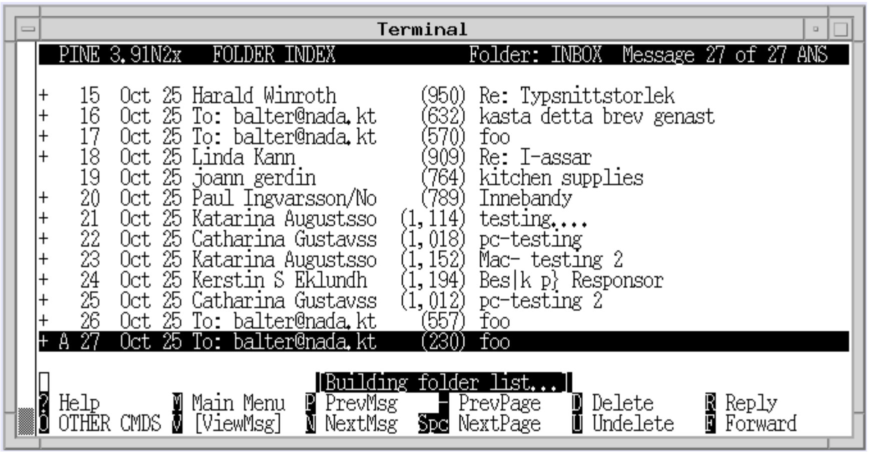
Text with menus

When menus are added to a program the user does not have to memorise commands, only recognise them. An example is Unix based Pine. Pine 3.87 was constructed at the University of Washington in Seattle and is intended to be easy to use for beginners. Pine uses only one window, so you cannot see both the list of messages and an email message at the same time or two messages at the same time. A (user invented) remedy is to have another terminal window (when possible) where another Pine process is run with a read-only view. The sending of a message can easily be temporarily postponed, to allow the user to view other messages. Commands are given with one key, sometimes in combination with the ctrl-key.



```
PINE 3.91N2   MAIN MENU                        Folder: INBOX  346 Messages

      ?     HELP               -  Get help using Pine
      C     COMPOSE MESSAGE    -  Compose and send/post a message
      I     FOLDER INDEX       -  View messages in current folder
      L     FOLDER LIST        -  Select a folder OR news group to view
      A     ADDRESS BOOK       -  Update address book
      S     SETUP              -  Configure or update Pine
      Q     QUIT               -  Exit the Pine program

  Copyright 1989-1994.  PINE is a trademark of the University of Washington.
                   [Folder "INBOX" opened with 346 messages]
? Help                      P PrevCmd                 R RelNotes
O OTHER CMDS L [ListFldrs]  N NextCmd                 K KBLock
```

Figure 2.6: Pine main menu. Note the command menu at the bottom.



```
  PINE 3.91N2x   FOLDER INDEX            Folder: INBOX  Message 27 of 27 ANS

+    15  Oct 25 Harald Winroth       (950) Re: Typsnittstorlek
+    16  Oct 25 To: balter@nada.kt   (632) kasta detta brev genast
+    17  Oct 25 To: balter@nada.kt   (570) foo
+    18  Oct 25 Linda Kann           (909) Re: I-assar
     19  Oct 25 joann gerdin         (764) kitchen supplies
+    20  Oct 25 Paul Ingvarsson/No   (789) Innebandy
+    21  Oct 25 Katarina Augustsso (1,114) testing....
+    22  Oct 25 Catharina Gustavss (1,018) pc-testing
+    23  Oct 25 Katarina Augustsso (1,152) Mac- testing 2
+    24  Oct 25 Kerstin S Eklundh  (1,194) Bes|k p} Responsor
+    25  Oct 25 Catharina Gustavss (1,012) pc-testing 2
+    26  Oct 25 To: balter@nada.kt   (557) foo
+ A  27  Oct 25 To: balter@nada.kt   (230) foo

                            [Building folder list...]
? Help        M Main Menu   P PrevMsg   - PrevPage  D Delete    R Reply
O OTHER CMDS  V [ViewMsg]   N NextMsg Spc NextPage  U Undelete  F Forward
```

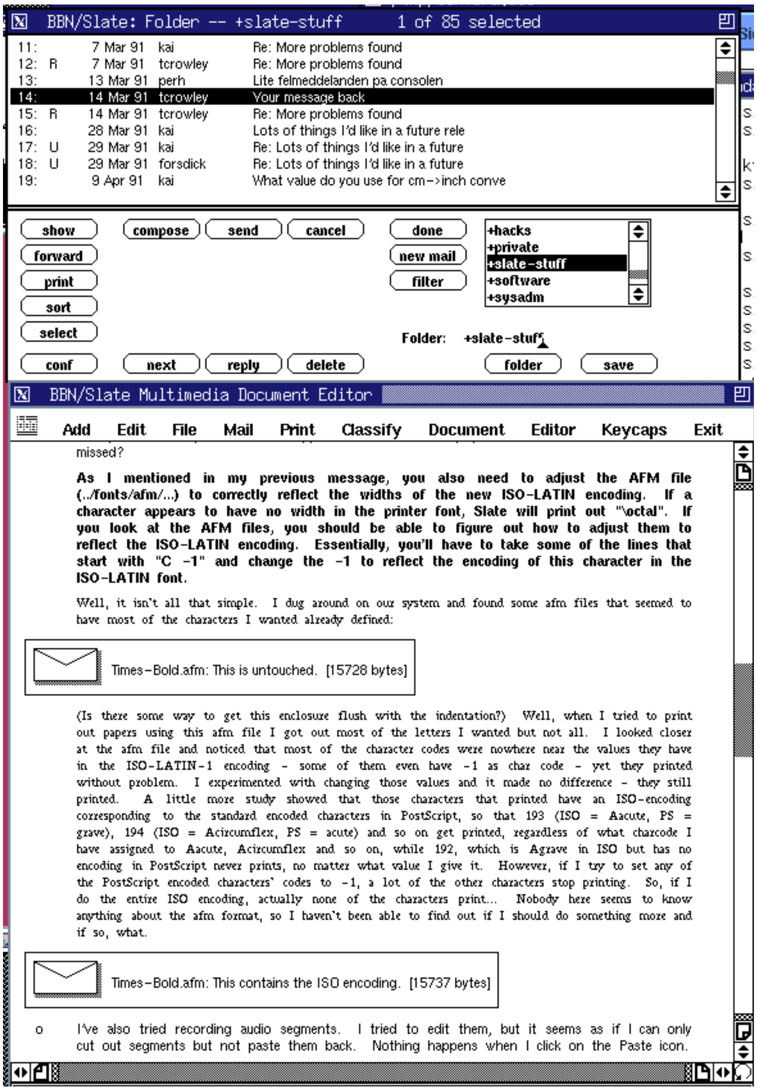
Figure 2.7: Pine list of messages

Graphical, non-direct manipulative, interfaces

With graphical support it is possible to have several messages open at the same time in different windows. Using a mouse can often be an alternative to typing commands. Buttons and fast-keys can be used for common commands. Examples are Mailstrom for the Macintosh, Slate for Unix and Pegasus 3.11 that runs on both Macintoshes, DOS and Windows machines. There is also a version of MH named xmh with a graphical interface and an extended version, exmh, with MIME-support.

Figure 2.8: Mailstrom with a list of messages and two opened messages

Slate 1.2 is a multimedia email program with conference possibilities. In contrast to other programs the editors for formatting text, sound, pictures and video are built-in. This means that a multimedia message can be read in Slate, without any use of other programs. Slate uses its own format BBN/Slate, but outgoing mail to receivers without Slate is automatically transformed into ordinary text.



```
11:      7 Mar 91  kai        Re: More problems found
12:  R   7 Mar 91  tcrowley   Re: More problems found
13:     13 Mar 91  perh       Lite felmeddelanden pa consolen
14:     14 Mar 91  tcrowley   Your message back
15:  R  14 Mar 91  tcrowley   Re: More problems found
16:     28 Mar 91  kai        Lots of things I'd like in a future rele
17:  U  29 Mar 91  kai        Re: Lots of things I'd like in a future
18:  U  29 Mar 91  forsdick   Re: Lots of things I'd like in a future
19:      9 Apr 91  kai        What value do you use for cm->inch conve
```

missed?

As I mentioned in my previous message, you also need to adjust the AFM file (../fonts/afm/...) to correctly reflect the widths of the new ISO-LATIN encoding. If a character appears to have no width in the printer font, Slate will print out "\octal". If you look at the AFM files, you should be able to figure out how to adjust them to reflect the ISO-LATIN encoding. Essentially, you'll have to take some of the lines that start with "C -1" and change the -1 to reflect the encoding of this character in the ISO-LATIN font.

Well, it isn't all that simple. I dug around on our system and found some afm files that seemed to have most of the characters I wanted already defined:

Times-Bold.afm: This is untouched. [15728 bytes]

(Is there some way to get this enclosure flush with the indentation?) Well, when I tried to print out papers using this afm file I got out most of the letters I wanted but not all. I looked closer at the afm file and noticed that most of the character codes were nowhere near the values they have in the ISO-LATIN-1 encoding - some of them even have -1 as char code - yet they printed without problem. I experimented with changing those values and it made no difference - they still printed. A little more study showed that those characters that printed have an ISO-encoding corresponding to the standard encoded characters in PostScript, so that 193 (ISO = Aacute, PS = grave), 194 (ISO = Acircumflex, PS = acute) and so on get printed, regardless of what charcode I have assigned to Aacute, Acircumflex and so on, while 192, which is Agrave in ISO but has no encoding in PostScript never prints, no matter what value I give it. However, if I try to set any of the PostScript encoded characters' codes to -1, a lot of the other characters stop printing. So, if I do the entire ISO encoding, actually none of the characters print... Nobody here seems to know anything about the afm format, so I haven't been able to find out if I should do something more and if so, what.

Times-Bold.afm: This contains the ISO encoding. [15737 bytes]

o I've also tried recording audio segments. I tried to edit them, but it seems as if I can only cut out segments but not paste them back. Nothing happens when I click on the Paste icon.

Figure 2.9: Slate with a multimedia message.

Direct manipulative interfaces

A graphical interface with mouse can include direct manipulation for dragging messages between mail folders or between the email program and the rest of the file system. The mail tools in this study that at least partly support drag-and-drop are Eudora and Sun's Mailtool V3.

Mailtool V3 is a Unix program that must be run under OpenWindows for full functionality. It handles drag-and-drop between the file system and the program, but not between mail folders. The possibilities to have several mail folders and message windows open at the same time are not available. Only one folder and one message can be open at the same time.

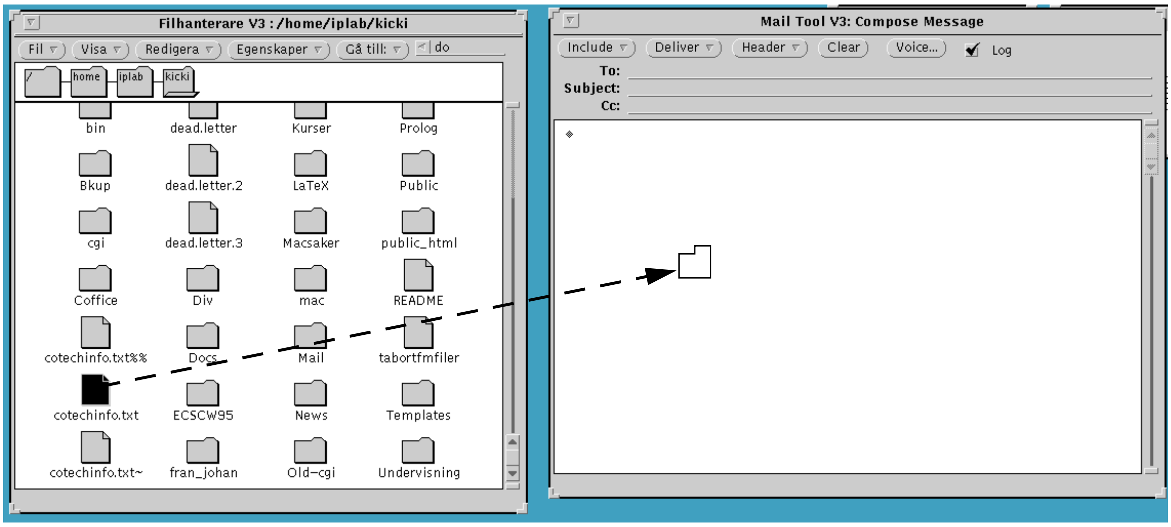


Figure 2.10: Mailtool V3, dragging an enclosure from the file system.

Eudora 2.1 runs on both Macintoshes and PC-platforms and has a similar look-and-feel as Mailstrom. Eudora has a queue of outgoing messages that can be sent all at once, so there is no unnecessary waiting when a message is sent and another should be written. This together with the POP-protocol simplifies use when a person is disconnected from the net. Messages can be dragged-and-dropped between mail folders within the program, but not between the file system and the program. Several messages can be open (in several windows) at the same time.

Figure 2.11: Eudora with two open folders and two open messages.

Conclusions

The studied systems display little difference between a text with menu and a graphical interface. The main difference in functionality is that a mouse can be used. A disadvantage of graphical interfaces is that text terminals cannot be used. A good interface for a user that uses different terminals would therefore have two modes:

1. A graphical direct manipulative interface with support for drag-and-drop both between the file system and the mail tool and within the mail tool. It should be possible to have several message and folder windows open at the same time.
2. A text interface with menus, with at least the most basic functionality.

2.4 Functionality

The functionality in the different groups of email systems are rather similar, but the necessary actions to perform a function differ. Examples are here taken from one mail tool in the text category (Unix mail), one tool in the text with menus category (Pine), and one mail tool in the direct manipulative category (Mailtool V3). Other mail tools used in this study having interesting features are also mentioned.

The tasks described are writing and sending a new email message, answering, forwarding, storing, overviewing messages, and handling of email addresses. The sorting and searching functionality is mentioned only briefly. For a more complete view of searching and sorting see [Reignier 1995 and forthcoming].

Writing and sending

In Unix mail an email message can be written and sent in two ways; by sending a file or “direct message writing”. The first case can look as follows (bold text is output from the computer):

```
computer> mail addressee <filename
```

Direct message writing:

```
computer> mail addressee
Subject: topic
cc:
This is the contents of the message.
.
```

There is no additional command to send the message. Note that the input of the message is finished with a single dot in the beginning of a line. In this write mode (that is during the writing of “`This is the contents of the message.`”), commands can be entered with a leading tilde (~) to e.g. start an editor with the message. The second possibility to write and send messages can be used even when the file system is out of order and is very fast for short messages.

In Pine, messages are written by entering the “Compose” window (by pressing “C” or selecting “COMPOSE MESSAGE” and pressing return in the main menu). The compose window has message header fields that the user fills in as a form (using arrow keys or the tab key to move between them). Addresses can be entered manually or by the T-command to enter the address book. Messages are sent by pressing ^X.

In Mailtool V3 the “Write”-button opens an empty mail window where the message and the address can be written. Despite the graphical interface, there are no possibilities to cut and paste. The message is sent by pressing the “Send”-button. Pressing the button with the right mouse button shows a pop-up menu which gives possibilities to send and then close the application, close the window, clear the window or send and leave the window as it is.

In Eudora it is possible to put outgoing messages in a queue and send them all at once. This has the advantage that the user does not have to interrupt the writing to wait for the messages to be sent (this can take anything between a few seconds to several minutes, depending on mail tool, the network, the number of recipients and the size of the messages). The user first writes all messages and then selects the “send now” menu alternative, and can continue with other tasks. The computer will then of course be busy in the background for a while.

Reply

In Unix mail, a reply to the sender is made by the command ‘r’ and the number of the message. A capital ‘R’ replies to the sender and all the recipients of the message. After the reply-command, the write mode is entered.

In Pine the R-command gives a choice of including the replied message and if there are several recipients to that message Pine asks whether the reply should be sent to the sender only or to the sender and all recipients.

In Mailtool V3 there are four different ways to reply: To the sender, to the sender and all recipients of the message, to the sender with the message included, and to the sender and all recipients with the message included. After the reply-command is entered, the write window opens.

In Eudora the reply command replies to the sender only. If the “alternative” key is pressed at the same time as the reply command is chosen, the reply will be addressed to the sender and all recipients.

Forward

In Unix mail messages can be forwarded by using the ~f-command and the number of the message. The message that should be forwarded will be included in a new message. After the ~f-command the write mode is entered.

In Pine the F-command can be used to forward a message.

In Mailtool V3 a message can be forwarded with the menu command “forward”. The write window is then opened with the message included.

Store

In Unix mail messages can be moved between mbox files by the ‘save’-command. After the save command a message list and the target mbox file should be entered. E.g.

```
save 17 42 oldmessages
```

will move message 17 and 42 to the file oldmessages.

Pine stores by default all outgoing messages in a folder for a calendar month. When a new month begins, the users is asked whether the two months old folder should be deleted or not. Incoming messages are stored in the folder ”inbox”, which grows and grows, also over month bounds, and had to be kept under control through manual deletions and movements of messages between folders.

In Mailtool V3 one message at a time can be moved by selecting a folder from the Mailfile menu to add the folder to the move menu, then selecting the message, and then select the target folder in the move menu. If more messages should be moved to an already added folder, they can be moved to any folder in the move menu by selecting the message and then a folder in the move menu. The only way to move several messages at the same time is to use the search with the “select all” option and then the move menu. There is no way to select more than one message manually or to deselect just one of the searched messages.

Overview of messages

It is sometimes valuable to see other messages at the same time as a message is composed or read. Preferably it should be possible to, at the same time, view several lists of messages, from different folders or even the same folder, and several messages, both old and new.

In Unix mail the numbers of visible messages and folders are limited by screen space, see figure 2.4. Several windows can be opened towards the same mbox file, but there is no support for handling inconsistencies between the two versions, (as in Pine below). The last saved version is the only saved version.

In Pine, only one message or a folder can be viewed at a time, but there is support for starting a read only version in another terminal window that gives possibilities to view in one window and edit in the other (Pine locks the mail file). Another possibility is to postpone the sending (^O-command) of a message while viewing other messages.

Mailtool V3 does not support the possibility to have several messages and folders open at the same time in different windows, there is only one folder window and one message window.

Address storage

Addresses can be stored as aliases or in an address book. An alias is a (short) name and an address or a list of addresses that should be connected to that alias. The short name can then be uses as an alias for the address(es). An address book is a list of addresses and names. Preferably other information can also be added to a person. Both aliases and address books can be created by writing the information by hand, or by extracting the information from a message.

In Unix mail there is no support for addresses, but aliases can be entered in the .mailrc file (that also contains a lot of other settings).

Pine has a “take” command that extracts the address from an opened message. This is a nice feature since the user does not have to write the name and the address of that person. However, the user has to write a “nick name” (an alias) for that user and that user’s address. Pine also supports alias names for groups of users.

Mailtool V3 has support for aliases, but all addresses must be entered manually. There is no support for an address book.

Sorting messages

Sorting is used to display a list of messages in a different order.

There is no support for sorting messages in Unix mail.

Pine has possibilities to sort messages by Subject, Sender, Date, Arrival date and Size.

Mailtool V3 has support for sorting by date, sender, subject, size, status and number by selection from a sub-menu.

In Eudora, messages can be sorted by clicking on the header line in the mail list view window (e.g. sender, size, and date). Messages can also be prioritised at five different levels and this priority can also be used for sorting. Eudora also has filtering possibilities where the user can define rules for what to do with messages, e.g move to other folders or change their priority.

Searching messages

Searching is used for finding a single message, or a group of messages, with some attribut(es). When a group of messages is found it is sometimes convenient to refine a search by applying another search-command to that group of messages. E.g. first search for all messages from a person and then search for just the messages from that person in the last ten days.

In Unix mail, it is possible to search for characters in the subject line, mail from a specified user and messages of a certain type such as deleted, new, old, read and unread messages.

`& /foo` displays a message list with all messages that have “foo” in the subject line.

`& balter@nada.kth.se` displays a message list with all messages from balter.

`& :d` displays a message list with all deleted messages.

There are no possibilities to refine a search in Unix mail or to combine any of the search alternatives.

Pine has a rich functionality for searching messages, but only for information in the header fields (not on the contents of the body). A search can be refined by applying another search criterion to a selection. Messages can be selected manually as well. Several messages can be selected with different criteria and commands can be applied to the selection. A disadvantage is that it is only possible to search in one folder.

In Mailtool V3 there is a search window where criteria can be entered for common header tags: to, from, subject etc. The matching messages become selected in the mail list window but to view the messages a button in the search window must be used. Once a message in the messages list is selected, all others are deselected. There are no possibilities to refine a search.

When a search is started in Eudora, the user is asked which folder to search in. After each folder is searched Eudora asks whether it should continue with the next folder, the search is however continued in the background. This enables the user to start a search and continue with other tasks, while still having full control over the search and the possibilities to e.g. skip a large folder where the searched message definitely not is.

3 Related research and development

In this chapter, the status of research on usability in electronic mail today is described. In some sections, the impact of the described research on CoMail is mentioned. The chapter opens with some advice on how systems can be designed in 3.1. A summary of relevant research in electronic messaging follows in 3.2 and the chapter is concluded with a description of some important experimental and pioneering systems in 3.3.

3.1 User centred design

User centred design focuses on the user's needs and abilities. There are many formal methods defined on how to design for the user, see e.g. [Gould 1988]. Many developers share the insight that in order to be able to make a good design a person, or a team, must not only have technical knowledge, but also knowledge in other disciplines such as psychology, sociology and design. The principles upon which a system is built affect the *usability* of the system.

Usability

A general term used today for expressing the quality of an interactive computer application is its *usability* [Nielsen 1993]. Usability is a complex concept for which some aspects can be measured in terms of time to complete tasks, number of mistakes, number of actions, and need for help, but mainly it is a quality. Dix, Finlay, Abowd & Beale (1993) express it:

> "The primary objective of an interactive system is to allow the user to achieve particular goals in some application domain, that is, the interactive system must be usable."

Nielsen defines five attributes of usability:

- Learnability, a system should be easy to learn so that a user quickly can use it productively.
- Efficiency, once the user has learnt to use the system, it should be productive to use the system.
- Memorability, a system should be easy to remember so that casual users can recognise how to use the system even after a period of time.
- Errors, catastrophic errors should not occur at all, other errors should be few and the system should provide means to recover from errors.
- Satisfaction, the users should enjoy to use the system.

The international standards authority, ISO, is drafting a standard (ISO 9241) to define usability for office environment work stations.

To fully understand the design principles that follow, some terms must be described first (in the email context):

Metaphors are used quite successfully to teach new concepts in terms of ones which are already understood. Similarities between a known domain and the new one are noted and operators from the known domain are transferred to the new one [Dix et. al. 1993]. An example is the desktop metaphor of the Macintosh which made computer usage easier for many people.

Direct manipulation is incremental action on visible objects in the interface with a rapid feedback. Actions are possible to reverse, every user action is a legal operation, and text commands are replaced with actions to manipulate the visible objects directly [Shneiderman 1983], [Dix et. al. 1993]. An example is moving a message from one folder to another by grabbing it, dragging it to the other folder and dropping it into the folder.

Feedback is important so that the user always knows what the computer is doing [Dix et. al. 1993]. Feedback can be divided in three different types:

1. Information that shows that the computer has understood the user's instruction.
2. Information that shows that (and how) the instruction has been executed.
3. Information that shows that the execution may take time.

An example is selection of messages from a specified user. When the selection is performed, the program can e.g. give information at the bottom of the screen "Performing selection, sender=balter". A watch is shown to indicate that this will take some time (still better: how long it will take), and when the selection is made the text at the bottom of the screen changes to "Selection, sender=balter".

Portability allows users that are shifting between different computer platforms to use the same program, or at least use the same data. An example is persons with a Macintosh at home and a Unix machine at work being able to handle mail at both places.

Consistency is use of identical terminology in prompts, menus, and help screens, and similar commands in similar situations. An example is that the delete button could be possible to use for deleting characters as well as messages and folders. This applies on several levels, there should be consistency both between, and within applications [Dix et. al. 1993].

Passive *adaptability* is when the program is user tailorable. An example is when the user can decide e.g. whether to use a Swedish or an English interface. Pro-active *adaptivity* is when the program itself adapts to the user [Dix et. al. 1993]. For example, if a person constantly deletes all messages from a person without opening them, the email program could ask if it always should delete messages from that person. Such adaptivity is difficult and must be implemented carefully. It can be handled by "intelligent agents" [Maes 1993] that draw conclusions from the way the user handles messages and tries "intelligent" guesses on how the user e.g. wants to have messages organised.

Undo is an important functionality. With the possibility to undo the latest action(s), the user will be more apt to explore the possibilities of the program and therefore learn more about it and maybe even better ways to work with the program [Shneiderman 1987].

User centred design

This section describes principles that were useful as a guidance for the design of CoMail in order to improve usability, to motivate people to use it, and for the testing of CoMail.

Heckel (1982) writes about 30 elements of friendly software design and 8 counter-productive instincts that are typical for engineers. These counter-productive instincts are very obvious when email programs are studied. Two examples:

- We structure for internal organisation. –A user does not want the program’s internal structure to intrude into the external interface. But it does. Mail tools using the mbox file format do not remove a message when it is moved to another folder. Instead the message is marked for moving. To actually move it, another command (expunge or similar) must explicitly be performed. The reason is that it takes time to remove a part in the middle of a file (the mbox file) and append it (or insert it) in another file.
- Our knowledge constrains our vision. –A user is focused on the task he is trying to do, and not the mechanism for doing it. More than once people have replied to all the recipients of a mailing list instead of just the sender (that was intended).

Norman (1988) gives many interesting examples of good and bad design of the things we see and use daily. Doors can be opened by pulling and pushing. A good design is to make the pushing obvious as an “affordance” by only providing a plate to push on and no handle. A bad design is to provide a handle that invites to pull even on the pushing side. A good design invites the user to do right, a bad design deludes the user to do wrong or in worst case even to destroy things.

Looking at the world in Norman's way, a world of terrible design is revealed around us and new examples can be found everyday:

- Bicycles leaning towards a house with a sign stating “No bicycle parking”. Five meters away there is an empty cycle stand. Illiterate bicyclists? No, the stand is too narrow to hold the broad tires of their mountain bikes!
- Beautiful straight line pavements meeting in right angles and a track of dirt across the corner of the grass field inside. The pavements was not built where the users wanted to go!

Norman suggest two principles to cope with these problems:

1. Make relevant parts visible.
2. Give each action an immediate and obvious effect.

Janlert (1991) expresses the same ideas:

> “The most basic demand on a user interface is that it should be designed so that the interaction becomes understandable for the user”.

Poltrock and Grudin (1994) have made the observation that successfully overcoming technical hurdles is far from a guarantee of success. They have found eight challenges for the developer of a successful system. Five of these were relevant for the development of CoMail:

1. *Who does the work to make an application succeed and who benefits from it?*
   Or as Cockburn and Thimbleby (1992) put it: "*All additional work must be motivated by personal benefit*". There are always problems with mail formats and preferences when a person switches from one mail tool to another. In order to appreciate CoMail, the user must find it easy to try CoMail, therefore all the necessary software installations should be made by those that have an interest of having their tool tested, instead of writing an instruction and burden the user with the installation.
2. *Obtaining a critical mass of users.*
   This was no problem since everybody at the university site already used email. However, ideas that required that CoMail was used in both ends, could not be extensively tested by "real" users, since the group using CoMail was small.
3. *The social, political and motivational currents that software must navigate.*
   Among the research students at the laboratory, those that developed in Smalltalk were both interested and motivated to use CoMail. A success for CoMail could result in a more wide-spread use of Smalltalk at the laboratory.
4. *Providing the flexibility required by variable work routines*.
   This can be achieved by allowing several ways to solve a task.
5. *Designing infrequently used features to be unobtrusive but accessible when needed.*
   In CoMail the search functionalities were hidden in the interface, to not disturb the novice user.

Twidale, Rodden and Sommerville (1993) have drawn several conclusions from their development of the Designers NotePad. Although users find it difficult to articulate new requirements to support their areas of expertise, they can be most effective (and vociferous) in saying what is wrong or missing from a provided system or prototype. Petroski (1992) observes:

> "Only a few men have sufficient integrative ability to invent form of any clarity, (but) we are all able to criticize existing forms".

Twidale et. al. continue with defining the advantages of using authentic tasks rather than invented tasks to test a system:

- It is naturally much more engaging for the user.
- The users ideally focus on their problem rather than the system, which is how the final system will be used. If the interface is clumsy and fails to allow this focusing the developers can spot what needs correction.
- Real world problems are different from artificial tasks: they are messy, do not fit into neat classifications, always contain exceptional features, are open-ended, contain elements of ambiguity and have a history and a future (so the system must cope with existing documents, ideas etc. and provide some product that can be used in later design stages or activities).

According to Twidale et. al. only when the most glaring errors and omissions are identified and remedied is it possible to spot the more subtle ones that are nonetheless crucial in their influence on usability, see figure 3.1.

More subtle errors and omissions are smaller. They are unlikely to be detected until the less subtle ones have been spotted and rectified. Simple test cases might not involve them and not detect them.

Figure 3.1: Illustration of the gradualist approach to developing and testing: using simpler cases first [Twidale et. al. 1993].

During testing of a system, there are always some general and particular suggestions of improvement. Twidale et. al. believe it is more productive to heed the particular, especially at the early stages of design. A user performing a task sees the need for an improvement with a hands-on experience of both the system and the task. General suggestions are particularly common from people observing a demonstration of the system but not using it to solve a problem of their own. When the user with hands-on experience notices problems with the application, the problems are genuine in a working situation. The general suggestions have often little to do with genuine problems.

Twidale et. al.'s findings leads to the conclusion that iterative design is very useful when constructing a system that relies on interaction with the user. Similar techniques with iterative design has been used for a long time e.g. at Apple [Houde 1992]. Houde describes how prototypes are created in MacroMind (nowadays Macro Media) Director (an animating and scripting environment), that enables rapid production of simulations of interface ideas. As the design progresses C++ is used to create prototypes to test more fully formulated solutions.

The small group of CoMail users contributed to the development with genuine, hands-on problems. They worked with CoMail as their only mail tool for several months. Their views, comments and complaints were extremely valuable for the development and to find the bugs. A new version of CoMail was released once a week during this period.

3.2 Studies of electronic mail

A lot of research has been done on email. In this section, the research papers relevant to this thesis are described shortly. The papers are here loosely organised into five areas, which are partly overlapping. The areas are communication, filtering, mail format, interface and studies of email users daily work. Each area begins with a description of the area and ends with the conclusions drawn from these papers and how this is related to the development of CoMail.

Communication aspects

Email is for many people a rather new medium. The speed and the asynchronousity implies that the communication process may be different from other media from a social point of view.

Sproull and Kiesler (1991) have studied the introduction of electronic mail into organisations and describe the changes of communication and the consequences of the new technique to communicate. The consequences can be grouped into social and technical factors. They conclude that it is very hard to predict the consequences, especially socially.

In the beginning, the technical consequences are most apparent, especially the increased efficiency. The efficiency increase is mainly due to the following factors.

- Speed of communication
  Email is as fast as a telephone within a company and there is no waiting for the sender of a message. No waiting for the receiver to pick up the phone and answer, or hang up an ongoing phone conversation.
- Asynchronous communication
  The sender can write and the receiver can read at different times. The sender does not even have to know where the receiver is at the moment. Email can be read from many places.
- Ad hoc communication
  The ease of communication supports informal meetings. Short discussions can be handled via email messages.
- Accelerate the flow of information
  It is easy to spread the same message to several recipients and forward incoming information to others.

The changes in social systems surface more slowly but some are:

- People communicate with new people
  It is easy to stay in touch even for people that meet seldom in person. Distance has less importance.
- Dependents change
  When people communicate with new people they receive information from new sources and the old, maybe formal, ways of achieving information become less important. Assignments can be given directly by email without passing any superiors. Groups may be distributed and still have a rich communication.

- Formal status becomes less important
  Many things are less apparent in email: gender, looks, intonation, position and status. Message passing that in a formal organisation always should go through the head of a department can be short-circuited.
- The way people behave and think may change
  When information comes from new sources, people may start to think in new directions. E.g. when using the short-circuited message passing mentioned above.
- More work is needed to achieve consensus
  When the formal status becomes less important it becomes harder to use status to enforce decisions, everyone can have their say.

According to Sproull and Kiesler, the content of electronic communication is more honest and straight forward because the sender is less aware of the receiver. For face-to-face communication there are social norms and instructions, implicit and explicit. Electronic communication introduces a new social situation. Norms are few and not accepted or known by all the users.

One main advantage of electronic communication is evident when a group is separated by distance and/or by time. A disadvantage (?) is that more discussions are necessary to achieve consensus, it is hard to convince people of new opinions via email.

Severinson Eklundh (1986) studied the dialogue process in communication via the COM system, and discussed the nature of email dialogues compared to spoken communication. A problem brought up in this study (see also Severinson Eklundh, 1994) is the fact that heavy users often do not deliver a response to a message within a reasonable amount of time. This may have the effect that the waiting recipient may try to attribute the silence to some particular reason. This can be frustrating for the recipient since the real reason for the late or missing answer is overload and not lack of interest.

In the COM study Severinson-Eklundh (1986) discusses the need for a context to email dialogues. Almost half of the users reported that they would often or very often check if a message had been received. Due to the COM functionality of providing possibilities to follow dialogue threads, the possibilities to include a message in a reply was not used at all.

In a recent paper, Severinson-Eklundh (1995) discusses the need for a context to email dialogues. This can be achieved partly by including the previous message in a reply. Another way is to have support in the mail system for following the dialogue threads. In many systems today, messages have to be searched for manually.

Conclusions

Email is more than just a way to communicate via computers. Email will cause more communication between people that already communicate, and contacts between people that have not communicated before. This increase in information can cause problems for the receivers and the speed of email communication puts demands on the receivers to answer quickly on email messages. The interface and the functionality will have a strong influence on the way the users work and use email.

Use and need of filtering

Filtering is a process where rules are defined for how to prioritise, sort and delete messages. Theses filtering rules can be applied both to sort incoming messages in different folders before they are read, and to messages already read and stored in different folders.

Jeffries and Rosenberg (1987) present a suggestion for an interface to handle filtering of electronic mail. They have studied 18 users and found that with their form-based language, non-programmers could produce filtering instructions at the same speed as programmers could do with a procedural language. Programmers using the form-based language were even faster.

Westergren (1989) also points out the need for filtering in her study of a PortaCOM installation. The message systems should support dialogue structure better and at least support filtering and sorting of incoming messages. Messages should be possible to limit in time (if the user does not read a message within a certain time, the message could be erased, e.g. a message about a possible lunch meeting is not interesting in the afternoon). Westergren also points out that the user should be able to list areas of interest and direct and receive messages that could be interesting for a person with those interests. The system should also support deleting of messages that the user does not want to read. Shet and Maes (1993) report similar needs in their paper “Evolving Agents For Personalized Information Filtering”.

Mackay, Malone, Crowston, Rao, Rosenblitt and Card (1989) have developed and studied the Information Lens system and found that useful rules can be created based on the fields present in all messages (e.g. To:, Subject:, From:). People used these rules to prioritise messages before they read them and to sort messages after being read. Delete rules were used primarily to filter out messages from low-priority distribution lists. Mackay et. al. quote an interview with a user:

> “these two (filtering) rules changed my life”

The problems with filtering are discussed in [Malone, Lai & Fry 1987], see also the description of the Andrew Message System here in section 3.3. Malone et. al. describe the problems with excessive filtering (the rules can be used to filter out messages that are personally addressed, users of such a rule could become less responsive to information from other people in the organisation), imperfect finding (people may have difficulties knowing what they want and do not know that they want to see a message until they have seen it) and conflicts of interest (an advertiser that has a message that most people would filter out may send messages with a popular topic to trick people to read them).

In [Killander WWW 1995] an effort to develop intelligent agents for filtering messages is described. The intelligent agent tries to draw conclusions from the way the user reads and sorts messages; and after a while tries “intelligent” guesses on how the user wants to have the messages organised.

Conclusions

These studies indicate that there is a need for filtering and it will probably grow with the number of incoming messages. Filtering can be based on the header fields in email messages and rules can be defined in such a way that non-programmers can use them. Filtering has been left out in CoMail while waiting for the user's opinions on filtering in the questionnaires and interviews.

Studies of use of mail formats with extensions

Electronic mail messages consist of two parts: A list of headers and a body. The body is used for the actual message. The headers define e.g. who the messages should be sent to (the To: field), the topic of the message (the Subject: field). Some of these headers are mandatory, but most are elective. Widely accepted in the Internet are the headers defined in [RFC 822], but new headers can be added. When new headers are added, other mail tools will not understand the headers, but in an experimental system this can be used to demonstrate the possibilities.

There is research on both what can be done with the existing email headers (e.g. those in [RFC 822]) and how these headers can be extended to support new functionality. An idea is that email can be used to distribute programs within a message and when the message is opened, the program is started. This interactive mail is interesting, but since these interactive messages have many similarities with computer viruses, the security problems are severe.

The headers can also be used to confirm delivery of messages. However, even though it would be convenient for the sender of an email to know when the receiver has read it, it is not always true that the receiver thinks the same. The receiver may not want to reveal that he has opened a message that he did not have time to read. There are systems that have possibilities of delivery confirmation, e.g. COM [Palme 1995 a].

Goldberg, Safran and Shapiro (1992) write “when electronic mail is used for computer-mediated conversation, users often find it hard to maintain dialogue continuity”. Their solution is “Active Mail”, a system that uses email to distribute interactive messages. Active Mail supports ongoing interactions between sender, receiver and future participants through direct (non-email) communication channels. This is done by starting other processes that can handle other means of communication. They also discuss how their system can be used to schedule meetings in a better way than ordinary calendar programs.

Cockburn and Thimbleby (1992) report about the Mona system. Mona uses the header information in RFC 822 format to establish two paired link types with each message:

- previous and next message by the same user
- the inferred cause(s) and response(s) of a message

The previous and next relationship is based on the time of sending. The cause and response is based on the arrival time of a message to a group of recipients (a similar relation can be based on the contents of Subject field and Re:). This information is used in Mona to build a web of the conversation structure that can be displayed graphically. In their 1992 system the rules that defines the cause and response links were fixed, but they point out that a Prolog file could be used to tailor the behaviour.

In [Anderson & Gillogly 1976] computational email, email with embedded programs, is mentioned. Borenstein takes up this idea in [Borenstein 1992]. Borenstein discusses the advantages of building Computer Supported Cooperative Work (CSCW) applications on top of email and the security problems with embedded programs in email. Borenstein solution to the security problem is “ATOMICMAIL”, a LISP-based language with all (?) possible security leaks removed. An alternative for sending email with embedded program could be to use the programming language Java that is developed with these security aspects in mind [Java WWW].

Conclusions

These studies indicate that it is both desirable and possible to use the established standard headers to enhance the support to follow dialogues in email tools, such as in the Mona system. Adding new formats within the MIME-standard is also possible, but requires that the receiver uses a mail tool with abilities to interpret these formats. In an experimental system, adding new headers can be used for test and demonstration purposes, but to study if these new formats are useful, an established large base of users is necessary to achieve a critical mass.

Interface paradigm experiments

Some researchers have studied the email interfaces and made paradigm comparisons between direct-manipulation and command based interfaces and studied the time to perform tasks and the number of errors.

Jones, Brock and Brassard (1990) have studied 28 users with three different user interface paradigms: Command line, Menu and Direct Manipulation. In this thesis these interface paradigms are named “Pure text”, “Text with menus“, and “Graphical” respectively in section 2.3.

Jones et.al. studied four different themes fundamental to email systems:

1. Visual Navigation
   Many users found the steps in both the command line and menu interfaces tedious. The programs could not provide a simultaneous view of the index while reading mail and this was disruptive to their work. The direct manipulation interfaces avoided this problem by providing a simultaneous display of the mail index while users were reading or sending messages. Through the use of multiple windows and the ability to point and click, the way in which users work with mail was extended by the interface.
2. Organisation
   Users felt overwhelmed by the amount of mail they received, and were uncertain as to the structure of the organisation (folders) they had created. The direct manipulation mail system provided an advantage by making the organisational structure visible, but the users also demanded a search function that searches across all folders.

3. Integration
   Users requested greater integration between email and other processes such as the electronic calendar.
4. Customisation.
   Users wanted to customise their environments to suit their work style, e.g. mark printed mail as read and shut off confirmation prompts.

The direct manipulative interface was superior for the two first themes, but also created new problems with synchronisation between windows when several windows were opened at the same time.

Benbasat and Todd (1993) have done two experiments which examine the effects of iconic and direct manipulation interfaces on the performance of casual users using an electronic mail system. Results indicate that there were no advantages associated with iconic representations compared to text-based representations of actions and objects. The subjects were introduced to the system and then performed a task twice in succession. After a week they performed the task a third time. Subjects working with direct manipulation interfaces completed their tasks faster than those with menu based interfaces. However, this difference in time was not significant when the task was repeated a third time, indicating that the benefits to direct manipulation might diminish after a learning period. No interface was better than the others in terms of reducing error rates when interacting with the computer system.

An explanation to why no advantages were associated with iconic representations compared to text-based representations is pointed out by [Ives 1982]:

> “While an icon may be worth a thousand words it is not always the thousand words the designer had in mind”.

Akin and Rao (1985) studied errors and time to perform certain tasks. They concluded that the more knowledge or resources that are necessary in performing a given task, the less efficient the interface is. The fewer commands used to perform a standard task, the more efficient the system. While the total number of commands used by expert users to perform these tasks is more or less the same as that of regular users, the pool of commands used exclusively by experts is significantly greater in numbers than that of regular users. Experts have a large “vocabulary” of commands but still use roughly the same number of commands to perform tasks.

Akin and Rao suggest that the studied system, RdMail, should be improved with:

- Ability to manage visual data on the screen through a window, to scan short headers, simultaneously with other text or graphical information, while composing text.
- A graceful editing capability built into the mail system, rather than moving into an edit mode.
- Greater facility to survey message contents without displaying all contents of messages files, i.e. the subject field and the first lines of a message.

Conclusions

These studies suggest that direct manipulation could be useful for sorting messages in folders. Icons should be used with caution and preferably for common tasks. Mostly the common operations should be supported with buttons. It is an advantage to be able to open an unlimited number of messages or folders at the same time, but then support to close them all at once becomes desirable. At least part of the contents of the messages should be visible in the list of messages. The users' need for different ways of organising messages should be investigated. The traditional list of messages could be complemented with a possibility to use two dimensions as in the Macintosh Finder. Support for handling addresses in the mail tool is desirable. An ideal mail tool should be possible to run on several platforms.

Studies of users daily work with email

Some research has been done on how the users of electronic messaging systems perceive their medium. These studies have been focused on the users point of view in contrast to the earlier mentioned reports that often concentrate on what can be done technically.

Nava Pliskin (1989) points both at the advantages "a wonderful dream come true; to be in touch, daily and for free, with dear colleagues" and the problems. She bases her conclusions on her own experience and divides the problems into four categories:

1. Addressing difficulties
   When communicating with people outside your own site you often run into problem with their addresses. Even though an email always contains a return address, that address is too often wrong. There is no catalogue with email addresses, like the phone book. Mail computers in the net change name, sometimes without notifying persons and computers that still try to use them. A similar function to the phone message "The number has changed, the new number is #" would mean a lot.
2. Unreliability issues
   There is no possibility to know whether a message has reached its destination or not, until the addressee responds to it. Messages that for some reason cannot be delivered are usually returned to the sender, but this is not always true. Some systems have an automatic acknowledgement of delivery, but Pliskin reports that even that is unreliable. Acknowledgements have been sent without any delivery.
3. Medium limitations
   From the users point of view it is impossible to send anything but ASCII-text and there is no possibilities to send a signature (e.g. on a contract).
4. Interface problems
   Users often have to retrain when their mail box is transferred to another combination of host computer and network.

Pliskin's list of desirable improvements includes: Simplification of address codes, automated address directory compilation, interfacing to other media and automatic message tracing. The part with interfacing to other media has in theory been solved now, at least partly, with MIME.

Hjalmarsson, Oestreicher and Waern (1989) report similar problems for the six users participating in their study: to know whether the message has reached its destination or not and the lack of address catalogues. Other results are that when “elementary users” (two secretaries) run into a problem, they prefer to ask an expert for advice rather than using the help functionality in the system. Both the paper documentation and the on-line help were said to be difficult to handle and to understand in most situations. The advanced users (two technicians) used the documentation more.

The study also indicates that “different users have different requirements” and that elementary users could not explain their needs or express their requirements. Some requirements which the users did not think of spontaneously were revealed through explicit questions during the interview. Hjalmarsson et. al. suggest that this may be due to the lack of knowledge about what to demand, but it can also be due to the habit of adapting to whatever technical tool is provided, and not demanding anything from the tool itself.

Hjalmarsson et. al. suggest solutions to some of the problems: A modular design in which various subtasks can be performed in different ways and that the end-users should be involved in the design process.

Some of the users of email are overwhelmed by the information flowing into their email boxes and conference systems. This overflow phenomenon has been studied by Hiltz and Turoff (1985). According to them, this overflow causes the users to

1. answer only parts of the incoming mail,
2. answer less correctly than they otherwise would have done,
3. store information and answer it when they have time,
4. systematically ignore some of the characteristics of the incoming information,
5. finally leave the system.

They conclude that overload, within the context of an organisation, is essentially a behavioural phenomenon. Individuals must learn screening skills and develop shared norms about sending behaviour so as not to impose unwanted material on others. It makes more sense to address the inappropriate behaviour through social norms and sanctions than to obscure the problem with software.

Mackay (1988) has interviewed 18 of 60 people at a research laboratory as a pre-study of an introduction of Information Lens at that laboratory. In her study it was found that the feeling of being overloaded varies a lot among people, independent of the number of messages they send and receive. In Mackay's study one person felt that 36 distribution lists were few, while another person thought that 20 were a lot. One person thought that it was possible to control the email situation when receiving 75 messages a day, while another person felt overloaded with 23 messages a day. The users that feel out of control are often, according to Mackay, the persons whose work do not demand immediate response to mail, but feel that they cannot stop themselves from reading anyhow. Persons who feel overloaded had some of the following characteristics:

- subscribe to many distribution lists,
- try to read all of their mail but do not always succeed,
- save hundreds of messages in their inbox,
- often do not reach the bottom of their inbox,
- want to save a large part of their mail,
- maintain many folders on different topics where mail is saved,
- have difficulty finding messages.

Mander, Salomon and Wong (1992) have studied how users organise their papers, messages and other physical objects of information in real life. Their conclusion is that users like to group items spatially and often prefer to deal with information by creating physical piles of paper, rather than immediately categorising it into specific folders. Computer users are confronted with large amounts of information, but currently are often provided only with a hierarchical filing system for managing it.

Conclusions

These studies indicate that it is desirable to have support for handling email addresses built into the email program. Organisation of messages is essential and the mail program should preferably provide several ways to organise. The overload phenomena could be studied in the questionnaires and interviews. There is an Internet standard draft for support of tracing delivery of messages, but with the same objections as for introducing new formats above (page 30), this was put aside in this work. Help should be built into a system from the beginning. In that way errors, mistakes and bad interfaces in the help system become obvious as soon as a user has to go somewhere else for help. Questionnaires are not enough for evaluation of user demands on programs, more things will be revealed in interviews. The hierarchical way of organising messages is not the only possible way.

3.3 Experimental and pioneer systems

The difference between the email systems mentioned in 2.3 "Examples of email systems" and those that are mentioned here is that the systems here in some way differ from, or are extensions of, traditional email programs with new ideas. Most of the ideas are new ways to organise messages, filtering and support for group communication. Most of these systems have conferencing possibilities as well.

COM

KOM started as a conferencing system in 1978 at Försvarets Forskningsanstalt (FOA, Swedish Defence Research Agency). English releases were named COM. It has a similar structure as the later USENET news, but supports private and secret meetings. All messages in COM are stored in a database. The COM-system support dialogues by allowing the user to go to the previous (and next) message in the same dialogue, or simply skip a whole branch of messages in an uninteresting discussion. Therefore it is easy to follow a dialogue even if you enter a discussion after a long time [Palme WWW1 1995]. The pleasant feature of following dialogues is still missing in many of the modern email systems. Messages in COM were stored for approximately 4 months, before they were permanently removed.

The COM system, and later the PortaCom, was used by the Commission of the European Community as a central information exchange hub for several European cooperation projects. This system is known as EuroKOM and is run at the University College of Dublin [Palme 1995 b].

A new version of KOM with a graphical interface will be released late 1996 and will be running on PC's, Macintoshes and Unix machines. THe new KOM version will have a text interface as well and will have support for a common address book for the site, as well as private address books. Messages can be erased automatically after a folder specific time. Additional recipients can be added to, or subtracted from, a message even after the message is delivered [Palme 1995 a & b].

Palme has also written a checklist available on WWW with desirable email and computer conferencing features. The list includes features which, according to Palme, are important to users, but seldom provided by existing email systems. Many of these features are in the area of better overview and organisation of the message data base [Palme WWW2 1995].

Andrew Message System

Borenstein and Thyberg (1988 and 1991) reports about “Messages”, a high-end interface to the Andrew Message System (AMS). “Messages” is developed at Carnegie Mellon University in Pittsburgh, USA. “Messages” runs under several window management systems and has built-in multimedia possibilities. It has a set of features called “active messages” that in addition to the message also has information that directs a particular interaction with the user. Examples are voting messages, return receipt requests, enclosures, subscription invitations and redistribution messages. For the receiver the active messages appear as messages that magically bring up dialogue boxes and ask questions using mechanisms that are easy to understand.

AMS transparently supports messages which include text, pictures, animations, spreadsheets, equations and hierarchical drawings, while supporting text-only commmunication with low-end machines.

In “Messages” it is possible to use a LISP-like language called FLAMES (Filtering Language for the Andrew Message System) that can be used to automatically sort or classify new mail when it arrives.

The Andrew Message System supports group communication by bulletin boards. The bulletin boards can be private, public, official, administrative or any combination. There is also an advanced customisation kit not intended for beginners.

During the development of “Messages” a help system “Advisor” was built upon the features in AMS to handle questions from users. An attempt was made to automatically filter these messages to the persons responsible for different categories (“mail” for those that only answered mail questions). However the filtering was no success.

More than 50% of the messages ended up in the miscellaneous category, as unsortable, because the users with problems had problems with specifying their problems. The computerised filtering was replaced by humans. All help questions were directed to a group of students that only answered simple questions and forwarded all other questions to a person they knew would know the answer.

“Messages” was in 1990 used weekly by more 5300 people to read bulletin boards and is reported to be very popular among its users [Borenstein & Thyberg 1991].

Information Lens

The Information Lens has received a lot of publicity and many of the ideas have been incorporated into commercial products. The system is designed to support people in managing their electronic mail and other electronic messages. There is an advanced filtering possibility available, designed both to screen users from junk mail and to find messages of interest, even though the messages were not directed to the user originally. The filtering system supports cognitive, social and economic filtering. The cognivtive filtering is an intelligent match between needs of a user and contents of a message. The social filtering is based on personal and organisational relations. The economic filtering is based on message size [Malone, Grant, Turbak, Brobst & Cohen 1987].

The user can let Information Lens organise email semi-automatically. The user defines rules based upon the contents of email and the email is stored in different folders based upon these rules.

In Information Lens it is possible to define message templates. These semi-structured messages can be used for the filtering and for more advanced tasks such as communication with other applications.

Another concept in Information Lens is the “anyone server” where users can send out a message to an undefined group of users, the subset of the users that have defined rules that match the properties of the message. The idea was that it would support users with similar interest to communicate without necessarily knowing each other [Bannon 1993].

Studies by [Mackay, Malone, Crowston, Rao, Rosenblitt & Card 1989] of Information Lens show that users without significant computer experience can define rules and use them. The templates are not necessary to define rules. Rules are used both to prioritise messages before reading them and afterwards to sort them. Delete rules are primarily used to filter out messages from low-priority distribution lists, but not to filter out personally addressed messages.

Many of the users did not find the system of use in their working environment. People often wanted to determine themselves when to run the set of rules [Mackay 1990]. This was possible in Information Lens, but not intended from the beginning.

Object Lens

Object Lens is the second generation of Information Lens. Object Lens has an interface that integrates hypertext, object-oriented databases, electronic messaging and rule-based intelligent agents. According to [Lai & Malone 1988], Object Lens is a knowledge-based environment for developing cooperative work applications. As the name “Object” implies, Object Lens is object-oriented. Objects can use inheritance, just like objects in object-oriented programming languages.

Oval

Oval is based upon the Object Lens system and can be used as an end-user programming tool. To test Oval it has been used to implement copies of cooperative work systems such as gIBIS, Coordinator, Lotus Notes and Information Lens. The name Oval stands for Objects, Views, Agents and Links, which are the four key building blocks of the system.

Objects represent things in the world such as people, messages and meetings. Views summarise collections of objects and allow users to edit objects. Agents perform active tasks for people such as moving, mailing and deleting. Links represent relationships such as between a message and its replies or people and their superiors.

The test implementations worked fine, but there are problems with maintaining large numbers of linked objects [Malone, Lai & Fry 1992].

4 Survey and interviews with email users

For finding factors that influence the use and the convenience of electronic mail it is indispensable to study email users. This chapter describes the results of the survey based on the University questionnaire in appendix A.2, the Work situation questionnaire in appendix A.3 and interviews made with twelve persons, a follow-up on the two questionnaires or the Final questionnaire in appendix A.4.

The purpose of the study is to identify problems that are specific for different work categories, problems caused by a large email flow, users problems with the interface, users technical problems, and problems for users to organise their email.

This study is a parallel to [Lantz 1995] , that studies email use at an industrial site. Both studies are parts of research on heavy mail users in the NADA/IPLab project “Communication patterns in CSCW”, financed by the SAMT programme, a joint effort by AMFO and NUTEK.

The survey method is described in 4.1 and the results in 4.2. A comparison of this survey and [Lantz 1995] is made in 4.3. Conclusions from these surveys are made in 4.4. The interview method is described in 4.5 and the interview results in 4.6. The chapter ends in 4.7 with conclusions from the interviews.

4.1 Survey method

The questionnaire in [Lantz 1995] was slightly modified (see “Industrial questionnaire” in appendix A.1) to form a new questionnaire (see “University questionnaire” in appendix A.2). The reasons for the modifications were that some questions could be formulated more clearly to reduce misinterpretations. The modified questionnaire was sent by email to 32 users in a mailing list at a laboratory at a university site.

The research laboratory at the university site is a multi-disciplinary research group consisting of 32 people from areas such as computer science, linguistics, psychology and anthropology. The personnel consists of senior researchers and teachers, research students, programmers and administrators. Email is used both for communication within and outside the university. Many kinds of computers and email programs are used.

A total of 28 people answered the questionnaire non-anonymously. The last 4 are people that are currently not working at the university site. The users that responded represents both novice users, experienced users and users with a very high load of email.

When most of the questions in the questionnaire had been evaluated, a set of six additional questions was sent by email to the university mailing list. These questions can be found in “Work situation questionnaire” in appendix A.3. The reason for this questionnaire with extra questions was that we could not clearly identify some possible causes of problems with the first questionnaire and during the evaluation of the answers these questions were raised.

General problems in handling email

The main question is whether the respondents have perceived problems with their email. We asked in question 12 (U refers to the University questionnaire):

> U12. How can you handle your electronic mail today?

The users were asked to select one of the following alternatives.

> 1) To handle my email has never been a problem.
>
> 2) I have some problems with my email, but I have not solved them yet.
>
> 3) I have had problems with my email, but I have found strategies to handle it.
>
> 4) I am close to not being able to handle my email.
>
> 5) I am close to not being able to handle my email, despite having tried different strategies.
>
> 6) I cannot handle my email.

There were six different answers, but to form groups with a significant number of answers, these six alternatives are reduced in this evaluation: “No problem” (alternatives 1 and 3), “On the limit” (alternatives 4 and 5) and “Problems” (alternatives 2 and 6). The last group is occasionally split into “Problems” and “Severe problems”.

Problems are here defined as what the subjects themselves have defined as problems when answering this questionnaire. This is a weak point in this study. The subjects may have interpreted this question in different ways. Some of the respondents contradict themselves in other questions. Even though the respondents answered that they did not have any problems in question 12, they reported problems in other questions.

The result of question 12 will be used in most of the following sections. Often in attempts to determine what causes problems. Another question that is referred often in the following sections is the free form question 21:

> U21. What are the three major problems that you have with email?

4.2 Survey results

The answers of the University questionnaire and the Work situation questionnaire were statistically evaluated. When frequencies are mentioned in this thesis their mean value and standard deviation are written within parenthesises (m#, s#). In some cases, the standard deviation is too large to be interesting and therefore omitted. Percentages less than 10% (three people) are normally not mentioned.

4.2.1 Professions

Different professions have different work tasks and this may affect there use of email and the amount of problems. We asked two questions related to the profession:

U1. What is your profession?

_Researcher

_Teacher

_Research student

_Secretary

_Other:

U2. Describe your main work task.

Professions found at the university site are Researcher/Teacher, Research student, Programmer, Secretary and Student. The two last groups were small so they are here joined into a group called Other. The answers are presented in figure 4.1 together with the answers to question 12.

Profession	#	%	No problems	On the limit	Have problems	Severe problems
Researcher/Teacher	9	32	1	2	4	2
Research student	12	43	5	0	7	0
Programmer	4	14	4	0	0	0
other	3	11	2	1	0	0
TOTAL	28	100	12	3	11	2

Figure 4.1: Professions at the university site and their level of problems

Inspection of the table 4.1 shows a tendency to more (severe) problems for the Researchers/Teachers than the other groups and no problems for the programmers, but the material is too small for statistically significant conclusions.

The group of programmers can be interesting anyhow: why do they not have any problems at all? Are they unwilling to admit that they have problems or is their knowledge of computers always enough to handle situations that would be a problem to other users?

Totally at the university site 43% claim that they have no problems with email, 46% that they have problems. Many subjects answered “no” when they were asked about whether they had problems, but reported problems anyhow when asked more specific questions.

4.2.2 Size of email flow

When the flow of email increases, more time must be spent to read, answer, delete or store messages. A large flow may be a cause for problems. We asked five questions about the incoming/outgoing messages (ordered after their appearance in this thesis: question 6, 7, 10, 16 and 5):

> U6. How many email messages have you received and sent respectively in the last 7 days?

The results are shown in figure 4.2.

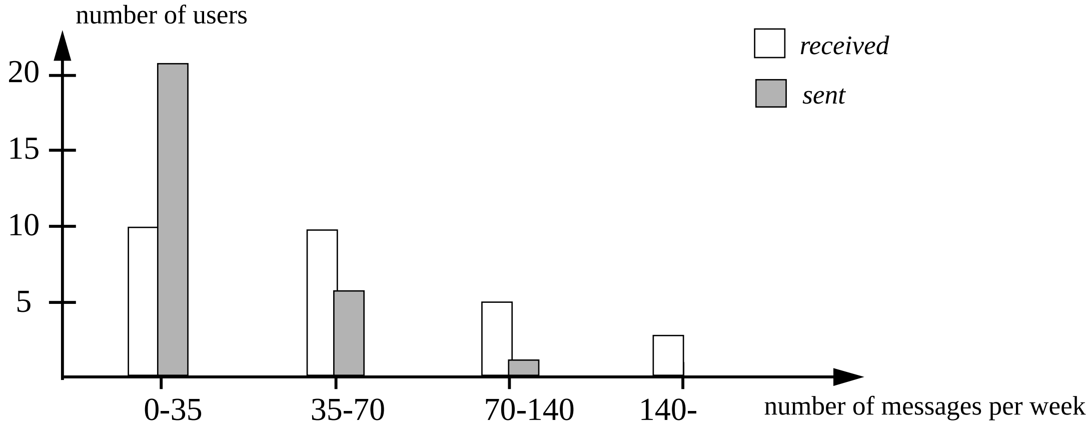


Figure 4.2: Number of received and sent messages in seven days

Most people receive less than 10 messages a day and send less than 5 messages a day. The extreme high was a person that received 1000 messages a week (>140 a day) and another that sent 100 messages a week (>14 a day). The person that received 1000 messages a week sent only 5 messages a week.

> U7. How large amount (in percent) of your email is private (that is addressed to you and not to some distribution list)?

The result is similar to a normal distribution around 50% and shown in figure 4.3.

Figure 4.3: Percentage of incoming messages that are private.

Many email systems have a folder where all incoming messages are stored, unless they are moved or deleted. We asked in question 10:

> U10. How many messages do you have in your incoming mail box (mbox, inbox or similar)?

The result is shown in figure 4.4. Most respondents have more messages in their inbox than can be displayed on a computer screen. Especially the MH users have many messages in the inbox, see figure 4.12. The extreme high was 3500 messages.

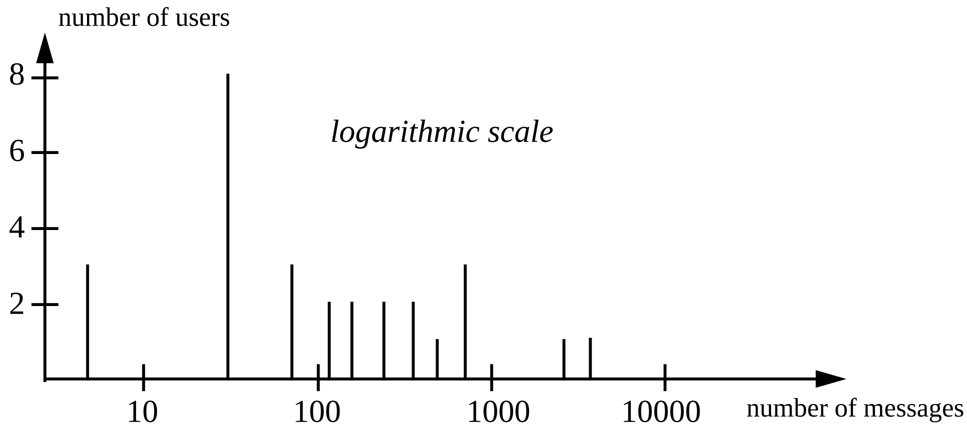


Figure 4.4: Number of messages in inbox.

The respondents received between 20 and 1000 messages in a week (m99). The amount of private messages ranged from 4 to 90 (m32, s23), the percentage of private messages from 1 to 100% (m53, s26). Each person sent between 5 and 100 messages in a week (m28, s21). The inbox, the folder where a person's incoming messages are stored, contained between 5 and 7235 messages (m410). There is a positive correlation (0.6, with 95% confidence interval) between the number of messages sent and the number of private messages received, see figure 4.5. Those that send many messages, receive many messages.

	Correlation	95% confidence interval Lower/Upper
Number of sent messages vs. # of private messages	0.6	0.3/0.8

Figure 4.5: Correlation between number of sent messages and number of received private messages.

There is no observed correlation, however, between perceived problems (question 12) and the number of incoming/outgoing messages or the number of messages in inbox. These results are similar to [Mackay 1988] and [Lantz 1995]. Our extremes were a person with four incoming messages a day and severe problems and a person with 280 incoming messages a day that had never had any problems.

A t-test shows a significant relation (P-value 0.026) between problems and the quotient between the number of incoming messages and the number of messages in the inbox, see figure 4.6. Since a user cannot do anything directly about the number of incoming messages (except for leaving some mailing lists, but the relation is significant in relation to the number of incoming private messages as well, P-value 0.039), this means that it is important to regularly “tidy up” the inbox in order to keep the number of messages as low as convenient.

	Mean	Standard deviation
No problems	4.34	5.19
Have problems	0.79	1.41

Mean value for the quotient between the number of incoming private messages and the number of messages in the inbox. An unpaired t-test of the null hypothesis (no difference in mean value between the groups) gives a P-value of 0.0262 (2.62% probability that the mean difference occurs by chance).

Figure 4.6: Problems vs ratio between incoming messages and the number of messages in the inbox.

This may be difficult for the many users that use the inbox as a “to do” list, and are overwhelmed with work. Therefore we asked in the Work situation questionnaire, see section A.3, questions about the work situation (W1, W2 and W6):

> W1. Do you feel that you have time to do what you should in your work?
>
> Use a scale from 1 to 5, where 1 is no, you do not have time at all and 5 is yes, you have time to do everything.
>
> W2. Are you often stressed and in the last minute?
>
> Use a scale from 1 to 5, where 1 is always stressed and 5 is never stressed.
>
> W6. Who decide what you should do in your work?
>
> Use a scale from 1 to 5, where 1 is someone else decides everything and 5 is you decide everything.

However, no significant differences could be found between the group that perceive problems and the group that does not perceive problems; and their rating of available time, or their rating of the amount of self determination in their work. There is a weak correlation (90% confidence interval) between a high level of perceived self determination and a high level of perceived stress. A conclusion is that lack of time is not the only factor that affects the differences shown in figure 4.6. Therefore it is important to support the user in limiting the number of messages in the inbox

When people read ordinary paper mail they often sort the messages, and may e.g. throw away the junk mail, or read messages with hand written addresses first. Do the email users handle their email in the same way? We asked this in question 16.

> U16. In which order do you read your mail (several alternatives possible)?
>
> _ Presentation order
> _ Private messages first
> _ Distribution lists first
> _ Select based upon the sender
> _ Select based upon the subject
> _ Select after size of the message

The messages were read in presentation order by 71%, the private messages were read first by 29%, and the distribution lists by 11%. The sum is more than 100% since many of the respondents use several strategies. Many of the respondents use different strategies depending on how much time they have. If time is short, many read only the private messages, unless they are waiting for a special message. In that case they use the subject or the sender to select what to read first (depending on if they know who would send that important message or if they know what the subject would be).

The respondents who read private messages first or read incoming mail according to subject receive more messages than the average (P-value 0.04). The number of private messages is not correlated to the order in which the respondents read mail. This indicates that more support to sort the incoming email might be useful.

Accessing the mail tool

When a new message arrives every fifth minute 24 hours a day, the receiver might find it disturbing to have the mail tool is running continuously. Reading incoming messages may interrupt other tasks. We asked the respondents about their frequency of accessing the mail tool.

U5. How often do you enter the email system?

_ Continuously. I read the mail when it arrives.
_ Several times a day.
_ Once a day.
_ Several times a week.
_ More seldom.

Answers are presented in figure 4.7. The mail system was accessed continuously by 54% and several times a day by 36%. Some subjects mentioned that they use the mail system at home as well, and the access rate is much lower when they are at home (less than once a day).

Access interval	Continuously	Several times a day	Once a day	Once a week
Number of users	15	10	2	1

Figure 4.7: Access interval

Filtering

One way to ease the burden of incoming mail is to filter it. The only tools in this study that support filtering today are Eudora and Slate. They allow the user to define rules that move, prioritise or delete messages based upon the contents in header fields (e.g. From, To, Cc, Subject, Size). We asked if the respondents would like to use such a function in question 15.

U15. If there was a possibility to automatically sort arriving messages (in different folders, e.g. one for a mailing list, one for messages from a certain person), would you use that?

Answers are presented in figure 4.8. According to the answers, arrival sorting would be used by 75% if it was available, regardless of level of problems (question 12). An interesting comment is that the only person that already had the possibility to do arrival sort did not use it. However, that person received only 25 messages a week.

	Yes	No	Do not know
Number of users	21	2	3

Figure 4.8: Hypothetical use of arrival sort.

4.2.3 Organisation of email

After some time of email use, the number of stored messages grows. When many messages are stored, the screen will no longer be large enough to view all messages and the messages must be organised in some way or deleted. With many messages stored there will be a problem with overview. We asked six questions about organising email (question 9, 11, 14 and 18, in the University questionnaire and question 3 and 4 in the Work situation questionnaire):

U9. How do you save the email you receive (several alternatives possible)?

_ All in the same folder.
_ A folder for each month.
_ Sort in folders after subject
_ Sort in folders after sender.
_ Delete everything.
_ Delete all messages after they have been take care of.
_ other:

Messages were stored in one folder by 50%, a folder a month by 14%, according to subject by 36%, according to sender by 18%. Some used the “other” alternative and declared that they deleted most of the messages (21%), or stored selectively in chronological order (14%). One person stored important messages in separate files (this person had very few messages). The strategy for storing email is not correlated to whether the subjects have problems or not (question 12).

U11. How many email messages have you saved in different folders or files?

Answers in figure 4.9. The respondents had between 25 and 7200 messages stored (m1500).

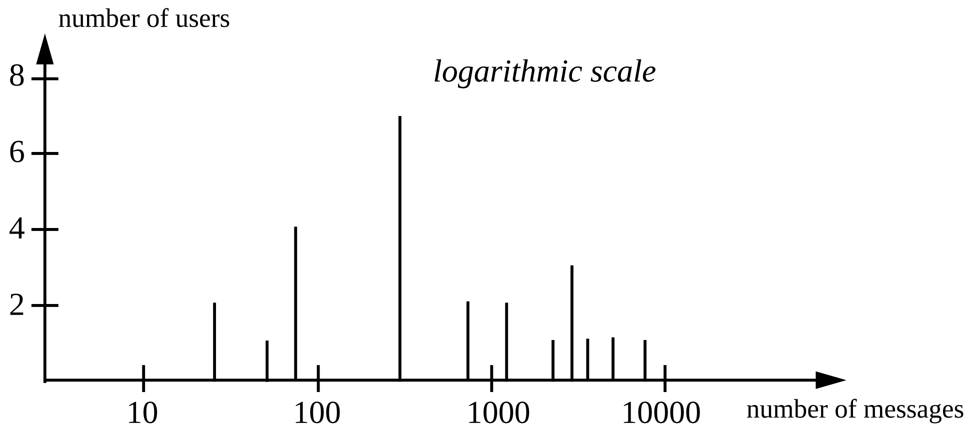


Figure 4.9: Number of saved messages

In question 14 we asked about situations where problems occur. The situations are related to organisation and to other tasks.

> U14. In which of the following situations have problems occurred (several alternatives possible)?
>
> _ When I shall receive information on what to do in my work.
>
> _ When I communicate with others.
>
> _ When I should order services from others
>
> _ When I want to find relevant information for my work
>
> _ When I should exchange documents (text files) with others.
>
> _ When I should save messages
>
> _ When I try to find saved messages
>
> _ When I try to delete messages
>
> _ Other:

Of these factors, organisation account for most of the problems; 18% of the respondents had problems when they were searching information, 25% when saving messages, 43% when searching messages and 39% when deleting messages. When asked what the three major problems with email were (question 21) 36% mentioned sorting, 25% mentioned searching. In fact 100% of those that said that they perceive problems (question 12) have problems with searching and sorting (question 14 and 21). An additional 33% of those that do not perceive problems (question 12) have problems with searching and sorting (question 14 and 21).

When asked what the three major features with email were (question 22) 14% mentioned that automatic storage of messages was one.

Persons that are good organisers in the “real” world might be good to organise on the computer as well and vice versa. This might affect the amount of problems so we asked in the Work situation questionnaire:

> W3. What order do you have on your papers? (ordinary paper mail, reports etc.)
>
> Use a scale from 1 to 5, where 1 very bad order (many unidentified piles) and 5 very good order (all piles are possible to identify, everything is sorted in files).
>
> W4. What order do you have on your files on the computer? (documents, programs)
>
> Use a scale from 1 to 5, where 1 very bad order and 5 very good order.

There is correlation between good order in the subjects' papers and fewer messages in their inbox and fewer messages saved, see figure 4.10. There is no correlation between the respondents perceived order on their papers and on their files.

The respondents rated their order of files on the computer higher than the order of paper in the real world (m3.9, s0.8, vs m3.1 s1.2). Those that perceived problems rated their file order lower than those that did not perceive problems (P-value 0.0002), see figure 4.11. No such differences could be found between perceived problems and the recipients rating of the order in their room of ordinary paper files (in fact, the rating of the order in the room was somewhat higher for the group with problems but this was not statistically significant). This indicates that order of files and problems with email are correlated. Better support for organising could decrease the level of problems perceived by the user.

	Correlation	95% confidence interval Lower/Upper
Paper order vs. # of messages in inbox	-0.57	-0.80/-0.18
Paper order vs. # of messages saved	-0.75	-0.89/-0.47
Number of messages in inbox vs. # of messages saved	0.77	0.50/0.90

Figure 4.10: Correlation between paper order and number of messages in inbox and saved messages.

	Mean	Standard deviation
No problems	4.5	0.5
Have problems	3.4	0.7

Mean value for the respondents rating of their file order, where 1 is very bad order and 5 is very good order. An unpaired t-test of the null hypothesis (no difference in mean value between the groups) gives a P-value of 0.0002 (0.02% probability that the mean difference occurs by chance).

Figure 4.11: T-test for File order (on the 1-5 scale) vs problems or not.

Delete

There is seldom any advanced support for deleting in email programs, but the need is great. Of those that said that they had perceived problems (question 12) 60% had problems when deleting messages (question 14 and 21). An additional 17% of those that had not perceived problems (question 12) had problems when deleting messages (question 14 and 21). Deleting messages is in many systems the *only* way of reducing the number of stored messages. There must be support both to delete and to undelete. When people feel that they can delete a message and still have the possibility to undelete a few days later (if they find that a message should not have been deleted), they will probably delete more. In the email systems known to us, it is only possible to undelete during the same session.

Finding old messages

For people with many stored messages, finding a certain old message can be difficult. The search can be done by reading through the list of messages until the searched message is found, or by first performing a *selection*. A selection can in the simplest case be a different sort order, e.g to sort all messages according to sender. A more advanced selection function may allow the user to e.g. select all messages that have a sender with a certain name. The selection can be presented in a separate window and finer selections can be made from that first selection. We asked the respondents how they found the selections possibilities in their mail tool.

> U18. Do you think that the possibilities to perform selections are adequate in your mail tool?

The possibilities to perform selections were reported as inadequate in their mail tool by 57%. Mailstrom and Mailtool V3 were rated better than Unix mail and MH, but Mailstrom and Mailtool V3 still caused complaints from 40-50% of the users.

4.2.4 Email address handling

According to [Pliskin 1989] and [Hjalmarsson et. al. 1989] email address handling is an important factor to improve the email programs. There are internal email address books for companies and the SUNET catalogue that covers all universities in Sweden, but no complete address book of all email users. In many places there is a “whois” service that can be asked for the email address of a person. We asked how the respondents handle their addresses today:

> U8. How do you handle email addresses of people that you communicate with (several alternatives are possible)?
>
> _ Use old messages.
>
> _ Use an electronic address book.
>
> _ Use a paper address book.
>
> _ Remember the addresses.
>
> _ Use another program that can search for addresses.
>
> _ Other:

To retrieve addresses 79% use old messages, 32% use their memory (remembered), 25% use an electronic address book, 18% some form of paper storage (internal phone book, list of students, business cards). Those who use paper storage have a tendency to be over represented in the group with problems, 78% of those who use paper storage had problems. Of those who use an electronic address book, 71% also use old messages to retrieve addresses. This implies that there is something inadequate with the address books.

Many subjects use the reply function to avoid searching for addresses, but when the number of people that you communicate with via email grows, this will cause problems. Messages stored just to keep the return address will be too many to survey and take too much disk space.

When asked what the three major problems with email were (question 21) 21% claimed that address retrieval was one.

4.2.5 Mail tool interface aspects

The type of interface to the email handler, see section Examples of email systems, will affect the use and maybe the amount of perceived problems. The choice of mail tool determines what kind of interface the user will have. A Unix mail user will only have a text interface, a Mailstrom user will have a mainly graphical interface. Since some mail tools are only available on certain computers, the choice of computer can affect which interfaces the user can choose between. Some users use a Macintosh, but handle their email with a Unix program (or vice versa). Such cross-platform usage may cause problems. We asked three questions related to the interface:

> U3. What type of computer do you use mostly?
>
> U4. Which mail tool do you use?
>
> U19. Do you think that a direct manipulative interface would be an advantage when you handle email (to drag messages to folders, attach documents to email messages)?

At the university site 75% of the personnel use Macintosh, 50% use Sun and 11% use PC Windows. At least 36% use more than one type of computer. We did not ask for use of more than one computer, but the subjects mentioned that anyhow. All researchers/ teachers except two use Macintosh, while the research students use Sun or both Sun and Macintosh.

Of the Macintosh users 52% have perceived problems while only 36% of the Sun users have perceived problems. The difference is not statistically significant.

A direct manipulative interface is perceived to be useful for handling email by 86% of the respondents. There is no significant differences between the views on direct manipulative interface regardless of computer type used or the type of mail tool.

Only 18% had at least a partly direct manipulative email interface, none of the respondents had a fully direct manipulative email interface. There is no correlation found between perceived problems and computer type, interface type, or cross-platform usage (persons that use one computer type, but a mail tool on another platform). One conclusion is that the difference between the three simplest groups of interface types (text, text with menus and graphical) are not large enough to cause any differences in the number of perceived problems. Comparative studies with fully direct manipulative interfaces remain to be done.

The only mail tools in this study with at least three users were Mailstrom, Unix mail, MH and Mailtool V3. Some notable differences in the usage are shown in figure 4.12.

Mail tool	# users	# emails received per week	# emails saved	# emails received from mailing lists per week	# emails in inbox
Mailstrom	8	61 (39)	680 (1000)	25 (26)	200 (140)
Unix mail	10	60 (35)	1100 (1400)	34 (26)	210 (300)
Mailtool V3	4	41 (17)	300 (320)	14 (7)	24 (12)
MH	8	64 (30)	2400 (2600)	27 (13)	1000 (1400)
All above and Elm (1 user), Slate (1) , Emacs (1), Pine (2) and Eudora (2)	28	99 (180)	1500 (1900)	66 (180)	410 (820)

Figure 4.12: Differences between mail tools, mean values and standard deviation. The number of users do not sum up, as some users used several mail tools.

MH users have more messages saved and more messages in the inbox than others. An explanation can be that it is easy in MH to view a list with only a part of the messages in a folder, messages have a specific number, and it is possible to open messages without selecting them from a list. In most tools, all messages must be viewed in a list first (e.g. Unix mail to identify the current number of the message) or a message must be selected first in order to open it (all graphical tools). With MH it is also possible to use ordinary Unix-commands to handle the mail. These three features together either make MH more suitable for handling large quantities of stored email or MH-users less interested in deleting messages or both.

The users of Mailstrom, Unix mail, Mailtool V3 and MH have a tendency to receive less from mailing lists than the average user.

4.2.6 Multimedia email

Multimedia possibilities, e.g. to send and read messages with embedded pictures, audio or video, will affect both the interface and the usage of email tools. Editors for pictures, audio and video must be added to the mail tools, more disc space is required to store the messages. In order to find out whether this multimedia feature is important to the users we asked in question 20:

> U20. In what way would multimedia messages (audio, video and pictures) add something positive to your communication?

A total of 86% of the subjects responded that multimedia email would improve their way of communication. The common reasons for this were that it would be possible to send pictures (33%) and national characters (18%) (which has nothing to do with multimedia).

When asked what the three major problems with email were, 14% claimed that the limited text format was one. Messages with formatted text (different fonts, sizes and e.g. bold text) and national characters, has nothing to do with multimedia, but can be solved for Internet users with MIME that also solves the multimedia problems. The most interesting multimedia possibility for the respondents is the possibility to send pictures. Nothing is mentioned in question 20 about national characters, but many of the respondents knew, or hoped, that MIME would solve this problem as well.

4.3 Comparison between an industry site and a university site

Lantz (1995) made a study of an industrial site. That study is similar to the study described in 4.1 and 4.2. A comparison of Lantz (1995) and the survey in 4.1 and 4.2 is given in [Bälter and Lantz 1995] and is shortly described here.

While the access rate to the email system and the number of incoming and outgoing messages at the industrial site was similar to the university, the percentage of users at the university site that perceive problems are higher than at the industrial site, see figure 4.13.

	No problems (1)	Have problems (2)	Have had problems (3)	On the limit (4, 5)	Severe problems (6)
Industry	55%	9%	27%	9%	0%
University	37%	40%	6%	9%	6%

Figure 4.13: Respondents experience of how they can handle their mail

There can be several explanations:

1. There could be more external communication at the university site. National characters are often distorted when email messages are sent between different computer types or between different networks and attachments might become unreadable.
2. The use of modems could be more common at the university and that could add problems for the email users.
3. The diversity in computer types at the university site, see figure 4.15.

4. The diversity in mail tools at the university site, see figure 4.16.
5. The lower response rate at the industrial site, 29% compared to 88% at the university. When you send a questionnaire to people via email, the subjects that have problems or severe problems with email may be less inclined to answer than the subjects with no problems. At the university site, reminders could be sent by paper mail as well.

Explanation 1 and 2 were investigated in the interviews, see section 4.6.

4.3.1 Storage of messages

More messages are stored at the university, see figure 4.14. The mailtool MH is treated separated because it supports handling of large quantities of email. Even without those users, more email is stored at the university. An explanation might be that email has been used for a longer time at the university, and therefore the email users at the university have accumulated more messages. At the industry site, respondents with no problems had fewer messages stored in their inbox than respondents with problems (a t-test showed significant differences at the 97% confidence level).

	in the inbox	other folders
Industry	47	284
University	410	1500
University, mh users	1300	3000
University, except mh users	170	1100

Figure 4.14: Number of stored letters, subjectively estimated by the users (mean values).

4.3.2 Choice of mail tool

In this study, 7% of the industrial site users and 36% of the university users use more than one computer type, see figure 4.16. The number at the university site may be even larger since the question was ”Which computer do you mainly use”, but the respondents answered with several anyhow. The university policy on computers is to use the type that fits the job task best, at the industry site, they have a policy to do development for the Sun-platform.

	Sun	PC (DOS)	PC(Windows)	Macintosh
Industry	100%	5%	5%	5%
University	50%	-	11%	75%

Figure 4.15: Computer usage

More than one mail system was used by 34% at the industrial site and 39% at the university, see figure 4.16. The dominating complementing system at the industry was ”MEMO” (16 subjects, 26%). No system dominated as complementing system at the university. The number of users of more than one mailtool may be even greater at the university since the question was ”Which mailtool do you mainly use”, but people answered with several anyhow.

	Mailtool	mail	MEMO	Mailstrom	mh	other
Industry	74%	26%	26%	-	-	7%
University	14%	36%	-	29%	29%	50%

Figure 4.16: Mailtool usage

4.3.3 Address storage

One of the reasons why the subjects store mail is to store the email addresses. This is space consuming since the name and the address is around 50 bytes, but a typical short email is around 2 kilobytes, including headers. Many people use old messages to store addresses even though they have an electronic address book. At the industry site 49% of the subjects that use an electronic address book still use old letters to retrieve addresses. This indicates problems with the electronic address book. When the subjects at the university site were asked what the three major problems with email were, 21% claimed that address retrieval was one.

	Old letters	Memory	Electronic addressbook	Paper
Industry	50%	9%	60%	12%
University	79%	32%	25%	18%

Figure 4.17: Address retrieval

The problems with the electronic address book is related to the number of stored email messages in figure 4.14. More messages were stored at the university where fewer subjects used electronic address book. Of the subjects at the university that used an electronic address book, 71% still used old letters to retrieve addresses. People that use an electronic address book do not save less messages than people without electronic address book. At the industry site, people that used the electronic address book stored even more messages than people without an electronic address book! This is significant on the 99% confidence level, but it may be explained with a larger need for an address book when the number of messages is large, and that the address book in Mailtool V3 is unsatisfactory.

There are no significant differences between the group that uses an electronic address book and those that use other methods to store addresses and the number of received or sent messages.

4.4 Survey conclusions

Many of the email users perceive problems, even more than they admit. At the university these problems seems to be more frequent.

Work categories

The material was too small for stating significant differences in perceived problems between the different work categories but an interesting observation was the lack of perceived problems among the programmers and the (severe) problems among Researchers/Teachers.

Size of email flow

Exactly as the previous studies [Hiltz & Turoff 1985], [Mackay 1988] the conclusion of this study is that the size of the email flow has no influence on perceived problems. However, the number of messages influences the need for organisation, see below. The results from both the industrial site and the university indicate that it is important to help the user to “tidy up” the inbox so that the number of messages in the inbox can be kept low.

Interface

No significant differences in perceived problems between the interface categories text, text with menus and graphical interfaces have been found. These results are similar to a study of text-based and iconic representations in terms of time to complete tasks and number of errors. There is no observed difference between text and icons in [Benbasat & Todd 1993].

The support for direct manipulation when moving messages between folders, and between folders and the file system are even in the best tools in this survey only partial. It is possible to drag within the mail tool or between the mail tool and the file system, but never both, in the tools used in this study. Since organisation is an important source of problems, this lack of direct manipulation is unfortunate.

Technical problems and practises

National characters are often distorted and the email users perceive this as disturbing. The respondents are aware that MIME can solve this problem (for Internet users if everybody uses MIME consistently). Of the multimedia possibilities in MIME, sending pictures was the most interesting feature for our respondents.

Organisation of messages

Organisation of messages and files seems to be the single most important observed source of problems. Drag-and-drop both within the tool and between the tool and the file system could be an improvement. The search and sort facilities must be improved. Most respondents have more messages in their inbox than can be displayed on a computer screen, but still they believe that they have better order of the files on the computer than of papers in real life. There is a demand from the respondents for more support to delete messages.

Many messages are stored just for having the address stored. An advanced address book can give the user freedom to delete messages that are stored only to store the address of the sender. Those that perceive problems have a tendency to use paper to store addresses more than those that do not perceive problems.

Filtering incoming messages is an interesting possibility to help the user to organise and maybe decrease the number of incoming messages that must be handled manually. Support for keeping the number of messages in the inbox low might reduce the amount of problems for some users.

4.5 Interview method

The results of the questionnaire surveys gave some new questions that influenced the choice of subjects for the interviews. It was generally known at the university that the computing facilities group uses email a lot, and probably receives much more messages than the subjects in the questionnaire survey. In all these three groups (industry, university, computer facilities group) almost everyone have a background in computer science or a related technical subject. Non-technical people could have another view of email and its problems. Based on these considerations twelve interviews were made to four different groups of users:

1. Two technically educated people at a technical commercial company
2. Five people at a multi disciplinary research group at a university
3. Three people at the computing facilities group at a university
4. Two non-technical people at two different non-technical sites

The interviews were made according to the description in [Patton 1980] of an informal interview with open ended questions. As a preparation for the interviews, another “Final” questionnaire (a union of the University questionnaire and the Work situation questionnaire), see appendix A.4, was sent via email to the people in group 3 and 4.

Description of the interviewed email users

The two users interviewed at the industrial site participated in [Lantz 1995]. They work with computers daily, and have a Master’s Degree in some technical subject. Email is used mainly for communication within the company. Almost all people at that company use Unix computers and Mailtool V3 as it is company policy.

The five subjects that were interviewed at the university site were working at the same research laboratory that the questionnaire were sent to, see section 4.1. The interviewed subjects had participated in the survey and were selected to get a variety of users with both few and many incoming messages, few and many messages stored and users of different computers and mail tools.

The three subjects interviewed at the computing facilities group at the university site are responsible for the computers for the 140 people at the computer science department and approximately 5000 students at the university. They work both with hardware and software and rely heavily on email for internal communication and with students. Several types of computers are used, but no PCs. Many kinds of email programs are used since the computing facilities group often tests new programs.

One of the non-technical persons works at a medical research institute and assisted researchers when they need help with computations or their computer programs. Email had been used for a few months at that department. Both PCs and Macintoshes were used, Eudora was the only email program.

The other non-technical person works at a commercial company and had as main task to supervise the data from product tests, but this person was also a general help with computers and programs, although the person had no formal computer education. Mostly Macintoshes, but also a few PCs were used at the company. At the time of the interview, four different mail tools were used, but the company's goal was to reduce this to one mail tool within a few years.

4.6 Interview results

The citations from the interviews are translated from Swedish. In some citations added or removed parts are placed within “<>” to provide a context for the citations or to secure the anonymity of the interviewees. Within parenthesises after some citations there is a Q followed by a number and a text. The number after Q is the number of a related question of interest in the context and the text is the current person’s answer to that question.

Problems are more common at the university than at the industry site. Two of the suggested reasons from 4.3 could be investigated in the interviews:

1. There could be more external communication at the university site
2. The use of modems could be more common at the university and that could add problems for the email users

The first is true while the last part of the second is false. At the industry site email is used almost exclusively for internal messages, while email is frequently used at the university for communication outside the campus area. At the industry site people do not use modems, with few exceptions. At the university site people work both at home using a modem and at the university site, with a few exceptions. However, none of the interviewees complained about the modem or had any problems with the modem at all, not even the speed.

In the rest of this section we describe with citations results from the interviews using the same factors as in the account of the survey in section 4.2.

4.6.1 Size of email flow

In the interviews the subjects commented on their handling of their incoming messages. Four months had passed between the questionnaire survey and the interviews. Most subjects had more messages stored at the time of the interview. The rate of incoming mail was approximately the same. No numbers had decreased.

The subjects from the computing facilities mostly wanted support for processing incoming mail fast and to delete it:

> "Let the mail program read in advance and sort all messages with the same subject together. Not only on subject but also hierarchically within each topic so that you can see which message is an answer to which."

> "It takes a long time to read and answer all messages. I think that default answers would help, I mean "I don't give a shit for your...", or "We don't do PCs here."

> "I like to use one-character commands. It is extremely important for me that I can process a large amount of messages in a short time. Most messages I can delete at once. Of the 2000 letters I receive in a week, 1500 can be deleted directly without any action, but I still want to see them. Even if it's only notifications from some computer that needs attention."

The subjects from the computing facilities group share addresses (a message to "staff" results in several messages to all members of the group "staff"), this causes special problems with answering messages:

> "It is not the large amount of messages that is the problem, except for the chance that something may be lost in it. The worst problem is that there might be a message that no-one answers. We should get some kind of warning after a while when no-one answers a message <that we (staff) have received>."

Others perceived the same problems as the computing facilities group:

> "It would be nice if you got a reminder if you forget to do something about a message".

Cleaning the inbox is essential for some users:

> "If the inbox grows up towards 40 messages, then I start to feel sick about it."

There are also problems that you cannot solve with changes in the mail tool only ([Malone et. al. 1987] have drawn the same conclusion):

> "It's not the mail itself, it's the amount of information in general that went parodical."

In the interviews the subjects were asked if they found arriving mail interrupting for other tasks. No-one thought so.

> "No I think it's fun to receive mail" (Q6: 4 received messages a day, Q5: access continuously)

> "No I'm the one who makes the choice. I sort of only care when I want. I'm not bothered." (Q6: 70 messages a day, Q5: access continuously)

> "Well, a little sometimes. But then you can, like, ignore it."(Q6: 290 received messages a day, Q5: access continuously)

> "Yes, in a positive way. More micro-breaks that you need. So when I'm working I get a reason to think of something else. I don't feel interrupted. If I have something that I really think about, I don't care if something has arrived." (Q6: 10 received messages a day, Q5: access continuously)

This indicates both the importance of email and that email is perceived as less disturbing than a telephone. As a comparison, one might imagine a telephone call every fifth minute 24 hours a day.

One way of eliminating many notifications from the mail tool about incoming mail is to have a “hot list”. Only messages from the senders on the hot list, or people on the hot list, will cause the mail tool to notify the arrival of a new message. The interviewees were asked if they would find such a function useful.

> “I would definitely use the function to notify me if a message from this person arrives.” (Q6: 29 messages a day)

> “No I wouldn't use the watch function for messages from a certain person or subject. I read my mail when it arrives.” (Q6: 4 messages a day)

As many other functions, some find it useless and some find it vital. Here, many incoming messages might increase the interest in this function.

In the survey the wish for arrival sorting was clear. In the interviews, the subjects were mostly positive, but had different views on how they wanted the filtering to work. Some wanted incoming messages to be stored directly into folders, some wanted this sorting only to be a suggestion for where to store the message after reading it.

> “Arrival sorting is OK, but it shouldn’t be moved to a folder until I’ve read it.”

The only interviewed person that already had arrival sorting possibilities said:

> “I would refuse to use a mail tool without arrival sorting. Arrival sorting is the most important function.” (Q6: 170 messages a day)

4.6.2 Organisation of messages

From the survey it is clear that organisation of messages is an important factor for perceived problems. In the interviews the subjects gave suggestions on how they would like be helped by the mail tool to organise and to delete. The exception was the two non-technical interviewees. They had very few messages stored and used only the inbox to store messages, although the rate of incoming messages was only slightly lower than at the university site. This can be explained by a shorter time of usage of email at that site, but also by that they use personal computers and are responsible for their disk space themselves.

One conclusion of the suggestions in the interviews is that it is important both to improve the current ways of organising and to provide totally new ways. Individual adaption may be an important factor.

> “I would like the mail tools to suggest a sorting and as I process each letter I would normally accept it, but still have a chance to just delete it or put it somewhere else.” (Q10: 110 messages in the inbox)

> “I moved everything that had to do with <subject> to a folder for it. But the support for that type of handling is not that good. You have to open that folder in particular and it is very difficult to search for a message if you do not know which folder it is in.” (Q10: 270 messages in the inbox)

> "I have a feeling that to get any order at all I would have to print out everything on paper and lay it out on a big desk <omitted> I do not have any good idea how I could do that kind of sorting on the computer." (Q10: 270 messages in the inbox)

The possibilities to sort and search can be improved:

> "It is important to be able to sort in different ways. Even when I have a folder for a topic I would like to sort them within that folder."

> "There should be a way to search that is independent of the folder structure."

A suggestion from one of the interviewees that had 9000 saved messages in 111 folders is an index function:

> "I would like to have index possibilities. You could make a list of words, connected with a list of messages that you can find those words in. That would not take that much space, but searching would be much faster."

Deleting messages

To delete messages is an easy way to decrease the problems with organisation, search and sort and those that do that in an organised way avoid many problems:

> "I clean regularly." (Q12: Never been a problem, Q11: 75 saved messages)

> "It takes long time to read each <again> before I delete them. It is better to sort them at once." (Q12: Never been a problem, Q11: 200 saved messages)

The support for deleting messages can be improved, but there are also problems that at first seem difficult to solve with a program:

> "There are many messages that I throw away, and then regret that I've deleted." (Q12: Have had problems, but have found solutions, Q11: 900 saved messages)

> "The problem with storing so much mail is that you don't know when a certain message transits from being important information to history." (Q12: Have problems, Q11: 500 saved messages)

> "The longest time is spent on scanning through all folders and deciding what should be deleted and what shouldn't." (Q12: Have problems, Q11: 5000 saved messages)

Other interviewees give simple suggestions to solve this problem with selecting messages that should be deleted:

> "I keep certain messages a week or a month, but if I never open them, time shows that I can delete them." (Q12: Never been a problem, Q11: 75 saved messages)

> "Sometimes I scan through my mail and say that I should delete all mail before, well let's say older than six months. It's a little bit awkward, I don't know. There's probably a better way to do it but as I have understood, you have to select these messages." (Q12: On the limit, Q11: 500 saved messages)

The interviewees where asked for their opinions on two possible solutions to the problems with deleting messages: A waste basket where all deleted messages stay until the user empties it or for a certain period of time, e.g. a week; and a function that deletes all messages in a folder when they grow older than a (folder) specific time, e.g. a year, similar to Pine’s “a folder a month”. Message-specific exceptions should be possible.

Most of the interviewees were positive, but stressed the importance of user control of when to delete.

4.6.3 Email address handling

During the interviews, there were both suggestions for improvements and complaints on the current address book functionality. The interviewees were asked whether they would like the address to be automatically extracted from incoming messages. Most thought that this was an excellent idea, but one person objected:

> “I don't want people to be automatically added to my address list, because I receive messages via mailing lists from people that I never have had any contact with.”

Two of the interviewees spontaneously came up with similar ideas when asked about the number of stored messages:

> “I save messages for the only purpose to have the addresses. I would like to have, to avoid saving a lot of uninteresting messages, a “save” button that extracts the address and then you can fll in a comment on who this person is, or let the program do it automatically. That is what I miss most.”
>
> “You should have a program that scans through all messages and collects the addresses so that you can shovel the rest of the crap.”

The demands on the interface to the address book are high. The address book must be better than the current alternative:

> “I keep a message from all people that I communicate with, even if I know their addresses. It is easier to press r (for reply) than to write the address.”

Even though there is an address book, there must be something wrong with it when the user prefers to make a new one:

> “I don’t use the address functions in the mail tools. <omitted> I have made a phone number list with the Finder <in the Apple menu with ordinary folders and fles>.” (Q4: Mailstrom user)

There are also suggestions for functionality that could be useful in an address book:

> “If I had an address list, I would like it to be sorted by frequency after the number of incoming and outgoing messages to that person. If you have it sorted alphabetically, you might get a message from someone that begins with an A, and you would see that person all the time, while you have to search for the persons that you communicate often with.”
>
> “I would like to be able to search for companies.”

One conclusion is that many things can be done to address books to improve their usability. For more details about improvements to address books, see 4.7.

4.6.4 Mail tool interface aspects

The choice of mail tool is not always made after thorough testing and consideration. At the industrial site Mailtool V3 and MEMO were parts of the standard user environment and this is the reason why all subjects at the industrial site used the same mail tool.

At the university the choice of mail tool is free, but the only subjects in the interviews that had actively chosen their mail tool by examining several different mail tools were the subjects at the computing facilities staff. Others used their tools mostly by habit. Often someone else had installed their mail tool and they had not investigated other alternatives, even though they had problems with their mail tool and knew about other mail tools that could solve some of their problems.

This may indicate that it is too complicated for an ordinary user to install a new mail tool and that the problems with transferring email messages from one tool to another are too severe. It is notable that almost all users in this study have a background in computer science or a related area.

In the interviews the subjects spontaneously asked for integration with other tools, more possibilities to do the same thing in different ways, better feedback and less limitations in the mail tool, such as forcing the user to use the mouse. A graphical direct manipulative interface would be an improvement:

> "Actually it would be more natural, I think if I had a tool to sort things in folders that were graphical. That you just dragged a message to a folder, then I would do it more than I do now." (Q4: text interface, Q10: 4900 messages in the inbox)

A graphical direct manipulative interface will not solve all problems, there must still be possibilities to use most functions from the keyboard:

> "With all due deference to drag-and-drop, it isn't particular fast. You cannot cope with drag-and-drop only, I think. You should naturally be able to do drag-and-drop as well, but there should be many ways to do the same things in a good program." (Q4: graphical interface)
>
> "A direct manipulative interface is important, but there must be a possibility to process the mail fast, without reaching for the mouse." (Q4: direct manipulative interface)
>
> "I admit that if I had a tool that allowed me to do a grouping of email, I would do it, but it must be simple, it shouldn't be more that a mouse click away." (Q4: text interface)

Many tools use a flag to show that a message is e.g. new, unread, marked, deleted, moved, prioritised. Some, e.g. Pine, provide a flag for "replied" messages. These flags should be user tailorable and it should be possible to set and clear these flags manually.

4.6.5 Who is adapting?

From the questionnaires and the interviews it is clear that there are many problems with the mail programs and that the users have many valuable things to say about the interfaces and the functionality.

One of the subjects defined important aspects of usability:

> "There are two things that decide whether a function is useful. It is how long it takes to do it and how many steps I have to do."

Naturally email should work like ordinary paper mail as much as possible:

> "I would like the email to work as the same way as ordinary paper mail. But now I have to think of formats here and formats there. Like, if I get a message in arabic, then I have to find someone who can read it. Why should the mail programs handle that?"

Portable computers demand very special knowledge, that few ordinary users have, for connecting directly to the Internet TCP/IP (modem connection is simpler, but much slower):

> "Everywhere you go where they have a connection to the net, you have to ask for an IP-address, bit mask, where the closest router is and what kind of name server they have got and so on. I mean, it's not like you have a socket in every phone booth, but we will get there eventually."

There are also problems that have nothing to do with the email program that forces the user to work in an awkward way:

> "One reason that I use mail on the Unix machine is that they automatically backup for you. If the Mac crashes, even if it hasn't happened yet, knock on wood, you would lose your email." (Q11: 9000 saved messages in 111 folders, Q4: Macintosh user)

The other comments from the interviews are here divided in adaptation, consistency, feedback, functionality and help.

Users are adapting to their tools

From the questionnaires and the interviews it is clear that the user must adapt to the mail program. When people are faced with a problem, they solve it, work around it, or adapt to it. This work-around or adaptation make people think that they do not have problems with their mail tool, but they have:

> "When I write to my American friends I write in Word and use the spelling checker and then I copy the message." (Q12: Never had problems)
>
> "It's good enough to find addresses in different folders." (Q12: Have had problems, but have found solutions)
>
> "I can only have one folder opened at the same time. I would like to have the inbox opened at the same time as I'm reading some other folder to check something out that is related to the message I'm reading."(Q12: Never had problems)

The programmers answered in the questionnaire that they did not have any problems at all. This may be caused by a resistance to admit that they have problems with a computer, but also their ability to find fast solutions to computer problems:

> "There are no problems with searching. I just read the entire mbox file into Emacs and use ordinary Emacs-commands to search for whatever I want." (Q12: Never had problems)

One person did not want to use email from home because of the problems, but did not think of this as a problem:

> "When I'm running mail from home, I use Unix mail and then the messages are stored in another way. This means that if I use mail from home I have to take care of that mail box manually to get it into my ordinary tool." (Q12: Never had problems)

Some interviewees simply avoid to do certain tasks to avoid problems, but still refuse to admit that this should be a problem:

> "I have avoided problems by not using certain functions that I really would like to have." (Q12: Never had problems)

Apparently the subjects have more problems than they admit.

Consistency

In Mailtool V3 "close" on a read window closes the window. But "close" on a write window closes the application. Once a write window is opened, the message must be saved! No matter if the message is empty or if the message should be deleted, you still have to save it! This and similar interface mistakes can be solved so easily, see e.g. [Hellgren & Olsson 1995].

A severe consistency problem is the structure of the mail files that makes it so difficult to switch mail programs. Almost every mail tool has its own format. Mail files and address lists cannot be transferred from one tool to another. Modern word processors can often read files from other word processors, even from other platforms. This should be possible for mail tools as well.:

> "My mail tool has an address book, but I only have one person in it. I was running an older version before and the new one isn't compatible with the last one."

> "I wish that the <file and folder> structure I have made should be possible to use, not only for this program, but for other mail tools as well so that you can use the best parts of each program. And then preferably for other programs as well."

> "There is definitely an inflamed question if you should change to something else. You must have the freedom to work as you want. Not forcing people to work with Macintoshes or Unix-machines."

> "I have to discover that the message has an attachment and that I can do with Mailstrom, but once I have opened the message with Mailstrom, then Eudora cannot open it. So I have to send the message again and then very quickly turn off Mailstrom so that Mailstrom does not catch the message again. So I switch

> to Eudora and read it. But it works."

Programmers that have connections will of course always find solutions:

> "No there is no problem with switching mail programs. You just call <the mail guru> and he has some kind of magic converter between the formats, or you can make your own filter to convert <the old files to the new format>." (Q12: Have certain problems, but have not solved them yet)

This can be a reason why the developers of email programs do not think of consistency as a problem. Another reason is that the developers might think that they would loose customers if it was too easy to switch mail tool.

Consistency in how programs handle e.g. attachments is also a demand:

> "The attachments must work in all tools, not be tool dependent." (Q4: direct manipulative interface)

Feedback

Exactly as the consistency problems, feedback problems are very easy to find and to solve, if you involve the users in the development process, but the interviewed users had feedback problems:

> "I don't really know if the contents of these two search fields are connected with "or" or "and". Preferably it should be possible to choose." (Q12: Have had problems, but have found solutions)
>
> "This black dot means that it is unread I think, it will disappear if I open this message, no it didn't. Strange, I wonder what it is?" (Q12: Have problems)
>
> "Here for example it says 'no new mail has been received' and then you might think that you have not received any mail, but that is not the case. <omitted> You see? I had mail even though it said that I didn't, so there are many people that wonder what that little sign is for." (Q12: Never had problems)

Functionality

The interviewees spontaneously complained about the functionality in their mail tools. Here are some citations and a short description on what should be done to the mail tool to solve these particular problems:

> "There is problems if the Mac goes down while Mailstrom is running. There is a process left in the Unix machine that prevents you from running Mailstrom again. You have to connect to the Unix system and kill a process there and then start all over again." (Q12: Have problems)

When the same user tries to run Mailstrom again, the process could be automatically terminated.

> "I have made a Word-document where I save addresses, and then I use cut and paste. It's a little complicated and maybe it will be impossible to manage after a while, but it will do for now. " (Q12: Have problems)

An address managing program that could communicate with the mail tool, or an address book functionality in the mail program that could communicate (e.g. via files) with other applications would solve this problem.

> "I don't have a signature file because I don' t want to have the same text at the end always." (Q12: Have had problems, but have found solutions)

A tool that provides possibilities for several signature files, and a default file that could be selected by the user could be better.

> "The mail tool asks me for a password each time I try to open a new folder. I can't do anything about it and it's annoying. You have to manage anyhow." (Q12: Have had problems, but have found solutions)

If the user so prefers, the mail tool could store the password during the session.

> "The problem is that the girls at the switch board run Macintoshes. It's hopeless because a lot of 'U's and control characters appear instead of åäö." (Q12: Have problems)

> "I sometimes write aao or aa, ae, oe instead of åäö. That is weak. "

Consistent use of MIME solves this problem for Internet users, but until all mail tools support MIME, some functionality that convert the 'U's and control characters back to åäö would be useful. At least between the character formats used on Sun, PC and Macintosh. If the user so prefers, the mail tool could automatically convert incoming messages with control or other strange characters.

> "When you reply you often want to include the name of the person that wrote the message in the first place, now I have to do that manually ." (Q12: Have had problems, but have found solutions)

The name can easily be extracted from the message and easily inserted, see "Address book" on page 76.

> "It forgets what is written in the search field between searches, that is very frustrating.

> "When I search I have to know the folder to search in. That's another reason to keep all messages in the same folder." (Q12: Have problems)

It is also a reason to let the search function work over all folders. The main reason to perform a search is to find items that the user does not know where they are. Why should the users have to know in which folder the searched messages are to be able to perform the search?

The program is prioritised higher than the user

In some cases the user is not allowed to disturb the mail tool:

> "It was disturbing <when the mail tool fetched a message every 5:th minute>, because the computer slows down when it fetches mail".

> "If I mark a number of messages that should be deleted and delete them, then I have to sit and wait until it's done. I cannot do anything in the meantime."

> "It is possible to adjust it to automatically expunge <but I don't do that> mainly because it takes time."

The users' actions should have the highest priority. If it takes time to copy files (that is necessary when messages are deleted and the mail tool is using the mbox format, see

File and folder structure in section 2.1), then the messages could be removed from the interface at once, and from the file system in the background.

Bad help

Some subjects wrote in their free-text comments to the questionnaires that they did not know how to perform certain operations in their mail tool, even for operations that they had searched for. Others asked for functionality that they already had, but were not aware of. There are manuals and help systems, but there must be something wrong with them. The subjects are not encouraged to explore their tools. Remember that many of these comments are from people at the computer facility staff, experienced programmers and computer scientists.

> “It wasn’t hard to do, but I asked <name> before I tried myself.” (university degree in mathematics with three years of computer usage)
>
> “There is a function for template documents, but I haven't managed to make it work.” (computer scientist)
>
> “It should be possible to refine a search, but I don't know how to do it. I've seen it done.” (computer facility staff)
>
> “I have read the man<ual> page several times, but never managed to make the select command work, except for the most simple cases.” (computer scientist)
>
> “I have a feeling that I do not use all possibilities. That’s another advantage of a graphical interface, it becomes easy to find more functions.” (computer science teacher)

4.7 Interview conclusions

Much can be done with the mail tools to support the user. Some of these suggestions have been described more thoroughly earlier in this chapter.

Size of email flow

When the flow of incoming messages increases, the need for support for processing them fast increases. Direct manipulation with a mouse is not enough, there must also be support for one-button keyboard commands. Filtering can be used so separate out e.g. messages from distribution lists, as in Eudora. Default answers could help users with a large number of incoming messages about the same topic. The results from the surveys that indicate that it is important to keep the number of messages in the inbox low are given some explanation in the interviews; too many messages in the inbox can make the user uncomfortable.

Interface

Some functions in mail tools could have a different priority level than they have today. The highest priority should be the user. When the user deletes a message, the message could disappear from the screen at once and the control should be given back to the user to continue with other tasks. The actual removing of the message could be done in the background so that the user seldom have to wait. If the user so prefer, the messages could be marked for deletion instead.

The user must have control over the flags for read/unread messages and similar. The help system should be developed together with users that do not know how the tool works.

Format standards

It should be possible to read mail folders in several (all) formats to simplify use from different computers and switching mail tools. The mail tool should preferably be running on all platforms.

MIME should be used to provide compatibility between sender and receiver. Consideration must also be taken to users without MIME and provide possibilities to at least convert incoming messages with distorted national characters back to the original characters.

Organisation of messages

Organisation can be supported both by improving the current methods, e.g with direct manipulation, and by providing totally new ways such as an index for all words that have occurred in the email messages. Drag-and-drop could be provided both within the tool and between the tool and the file system. It should be possible to search in all folders with one command. Filtering is a useful possibility, but the control on how and when filtering should be done must be with the user. The tool should provide flags or similar means to show which messages that are read, new, answered and so forth. The user must be in full control of these flags. There should be support for following dialogue threads.

An advanced address book can give the user freedom to delete messages that are only saved to have the address. A waste basket that stores deleted messages for e.g. a week can also help. Today, messages can be undeleted only during the same session as they were deleted.

Address book

Some functions for an useful address book are:

- Automatic extraction of addresses. It should be user tailorable whether it should extract addresses and names from: all new messages, only when the user says so, or only from certain kinds of messages.
- It should be possible to add more information than just the name and address.
- It should be possible to save the address book in some format that can be read by other programs.
- It should be possible to read a file with name, addresses and other information, so that an address book can be created by other programs.
- There should be a function that scans an existing structure of saved messages and extracts addresses, names and information from all saved messages. The “take” functionality in Pine is good, but not enough. When a user has used email for a while and has hundreds of messages stored, and then switches to Pine, it would be

very tedious to open each and every one of these messages manually just to extract the addresses.

- The address book should provide grouping possibilities (alias names).
- There should be search possibilities to search for both for individuals and (sub-) domains in the address book.
- There should be advanced support for deleting addresses, e. g. delete addresses older (not used for) two years, delete all addresses from a domain.

It should be possible to view several messages and several folders at the same time, preferably each in a separate window, as in e.g. Eudora and Mailstrom.

Adaptation to the mail tools

The users adapt themselves to the mail tools. Many users are not aware of this adaptation and answer that they do not have problems even though they can describe many problems that they experience.

Users learn the inconsistencies of the programs and e.g. avoid using the “close” button in some windows because it will close the application, instead of just that window. The users are also aware of inconsistencies between mail tools, such as the impossibility to read messages stored with Mailstrom from Eudora. Mail tools also use different formats to send attachments and national characters, which can give problems for the receiver, who must have a tool that understands the format in order to read the message. Mail tools could be improved with possibilities to read other mail file formats.

The users have many suggestions for improvements in their mail tools that would be valuable for the developers to know. Some of the most obvious mistakes occur when the mail program is prioritised higher than the user; so that the user has to wait while the program does work, e.g. sending email, that could have been done when time permits, in the background. Another is when not even computer scientists can understand the manual or the on-line help.

5 CoMail – a collaborative environment for email

CoMail is an interface to electronic mail and is under development at IPlab since 1993. The development of CoMail is based on an iterative process of prototypes and user studies. This chapter describes the development of CoMail. The influence from related work is described in chapter 3 "Related research and development" (in particular the design principles described in 3.1 "User centred design"). The iterative process of developing CoMail is not finished. Many of the suggested improvements in the survey and interviews are not yet implemented. The version of CoMail described here was (mostly) implemented before the studies described in chapter 4 were made.

This chapter opens in 5.1 with a description of CoDesk, the environment for which CoMail first was developed. A description of the development of CoMail and CoMail's most interesting features follows in 5.2. CoMail naturally contains many functions, but when these are similar to other mail programs they are not mentioned or only mentioned briefly. A summary of the results from studies of CoMail users follows in 5.3. The chapter is concluded in 5.4 with a description of suggested improvements of CoMail that follows from the user studies described in chapter 4.

5.1 CoDesk

The MultiG project was conducted from 1990 to 1993 [Pehrson & Sundblad 1994]. MultiG was financed by Swedish industry and the Swedish government (Telia, Ellemtel, Ericsson, NUTEK) and was a multi-disciplinary research programme with IPlab, SICS and Teleinformatics as academic partners. One of the goals was to develop multimedia applications that are possible to use in a network with high bandwidth. One of these applications was the Collaborative Desktop, CoDesk [Tollmar and Sundblad 1995].

CoDesk is a user tailorable environment for computer supported cooperative work. In CoDesk, a room metaphor is used to “furnish” meeting situations. CoDesk is partly a Macintosh-Finder-similar interface to a Sun/Unix-environment (see figure 5.1), partly a set of tools that are used for cooperation and communication integrated into CoDesk.

A main purpose is to extend the ordinary desktop environment with objects and concepts, in the form of icons and operations, for cooperation between users at separate work stations, that can be located in different parts of the world. A goal is to make cooperation as natural and ubiquitous as use of files, programs and folders in the common desktop interface.

Email can be used for intense communication, with several messages exchanged with the same person(s) within a few minutes, but in such cases there is often a need for more direct means of communication. CoDesk provides an environment for direct (synchronous) communication, as well as the indirect CoMail. The synchronous communication tools in CoDesk spans from simple talk, a window where two people can write, to video conferencing and collaborative drawing on a “whiteboard”.

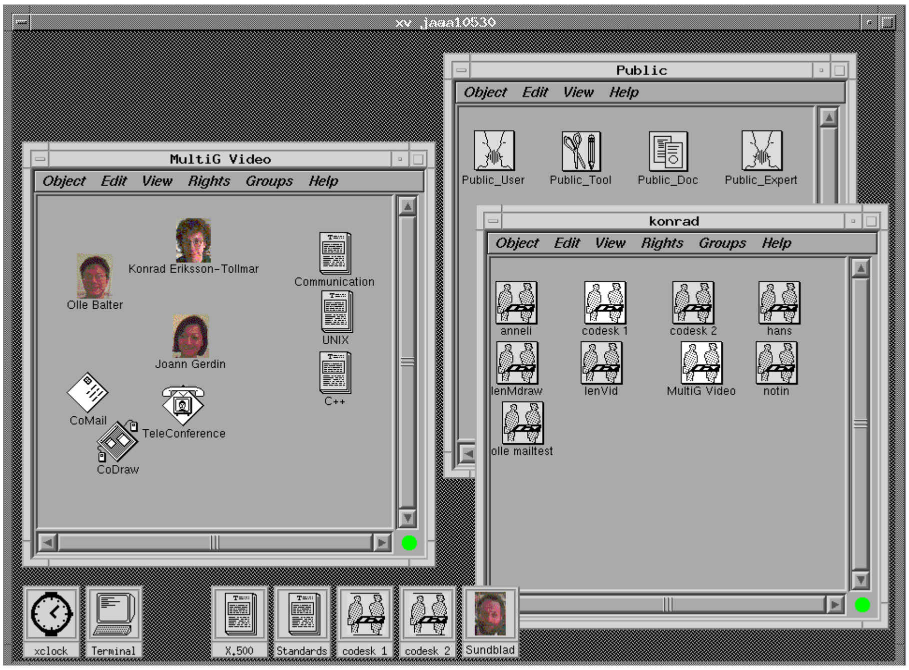


Figure 5.1: CoDesk

Another tool in CoDesk is CoBoard, see [Hägglund forthcoming] and figure 5.3. CoBoard is a USENET-news similar tool that uses a bulletin board metaphor where messages that are posted can be read and written by all people, not only by the author of the message. In that case the first person who reads a question can answer directly. The next person will then see both the question and the answer and the flow of information is reduced. CoBoard have functionality for awareness. If two users are reading the same bulletin board at the same time, they will be aware of each other.

Figure 5.3: A CoBoard with four read and two unread messages

5.2 Description of CoMail

The purpose of CoMail was initially to provide CoDesk with an interface to electronic mail. An email program in CoDesk must understand what to do when different types of icons are dropped onto the mailtool icon and vice versa. Informal interviews with secretaries and the computing facilities staff at NADA were made before any programming was done.

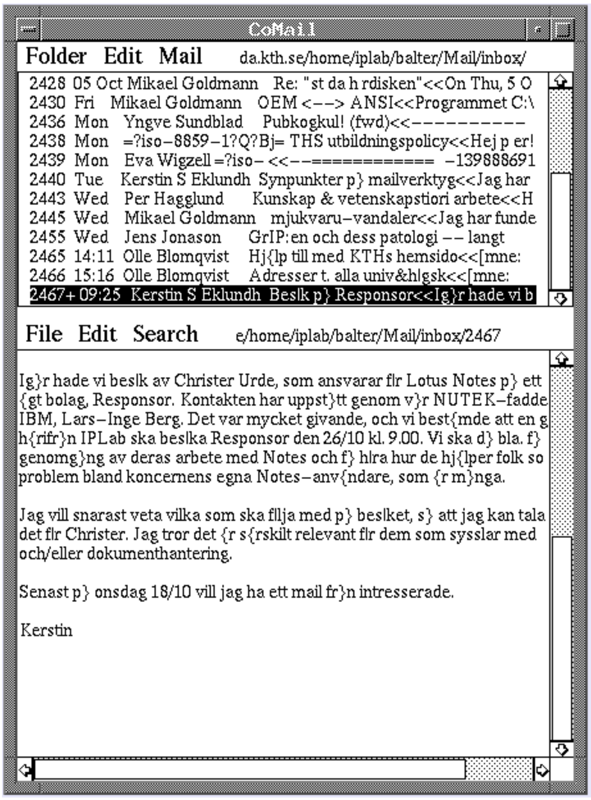
Surprisingly, the secretaries were very pleased with email, even when they used such old programs as Unix mail. The computing facilities staff had very many opinions on what should be improved even in modern programs such as Mailstrom. Most of the interviewed secretaries used email a few times per week while the computing facilities staff used email continuously every day. This may be the reason for the differences in opinions. When email is used infrequently, a simple interface with the most basic commands (read, write and send) seems to be enough. When email is used a lot, the demands on the interface and functionality grow.

After the discussions with the computing facilities staff and in the IPLab group we decided that a graphical interface to the Unix mail handler MH would be a good start. Basing CoMail on an existing mail handler had several advantages:

1. No time had to be spent on developing code for low level parts such as fetching mail from the mail spooler, make a list with headers of mail and so on. There are of course more general ways of doing this, such as using the C library C-Client for the connection between the interface and the server, but MH was a faster solution.
2. The interface for text terminals was already finished.
3. It would be easy for people that already used MH to test CoMail for a while and then go back to MH (if they were not pleased, or felt that the interface was stopping them from doing things that they could do with MH). CoMail and MH could be used simultaneously without any conversions between mail formats.

A graphic interface was a natural choice for CoMail as CoDesk has a graphical interface. Menus were used to minimise the number of commands a user must remember, see e.g. [Anderson 1990].

There are other graphic interfaces to MH, such as XMH and nowadays EXMH, but to be integrated with CoDesk the mail program must be able to understand drag-and-drop i.e. what to do when a document or a person or both are dropped on the mail tool icon: (the document should be used as an email message, the email address of the person should be added to a new email message or both). Since CoDesk was developed in C++ with the Interviews system [Linton, Calder & Vlissides 1988], this was the first choice for CoMail as well and after three months a useable version was running, see figure 5.4.


CoMail
Folder Edit Mail
da.kth.se/home/iplab/balter/Mail/inbox/
File Edit Search
e/home/iplab/balter/Mail/inbox/2467

```
2428 05 Oct Mikael Goldmann   Re: "st da h rdisken"<<On Thu, 5 O
2430 Fri   Mikael Goldmann   OEM <--> ANSI<<Programmet C:\
2436 Mon   Yngve Sundblad    Pubkogkul! (fwd)<<----------
2438 Mon   =?iso-8859-1?Q?Bj= THS utbildningspolicy<<Hej p er!
2439 Mon   Eva Wigzell =?iso- <<--============  -139888691
2440 Tue   Kerstin S Eklundh  Synpunkter p} mailverktyg<<Jag har
2443 Wed   Per Hagglund      Kunskap & vetenskapstiori arbete<<H
2445 Wed   Mikael Goldmann   mjukvaru-vandaler<<Jag har funde
2455 Wed   Jens Jonason      GrIP:en och dess patologi -- langt
2465 14:11 Olle Blomqvist    Hj{lp till med KTHs hemsido<<[mne:
2466 15:16 Olle Blomqvist    Adresser t. alla univ&hlgsk<<[mne:
2467+ 09:25 Kerstin S Eklundh Beslk p} Responsor<<Ig}r hade vi b
```

Ig}r hade vi beslk av Christer Urde, som ansvarar flr Lotus Notes p} ett
{gt bolag, Responsor. Kontakten har uppst}tt genom v}r NUTEK-fadde
IBM, Lars-Inge Berg. Det var mycket givande, och vi best{mde att en g
h{rifr}n IPLab ska beslka Responsor den 26/10 kl. 9.00. Vi ska d} bla. f}
genomg}ng av deras arbete med Notes och f} hlra hur de hj{lper folk so
problem bland koncernens egna Notes-anv{ndare, som {r m}nga.

Jag vill snarast veta vilka som ska fllja med p} beslket, s} att jag kan tala
det flr Christer. Jag tror det {r s{rskilt relevant flr dem som sysslar med
och/eller dokumenthantering.

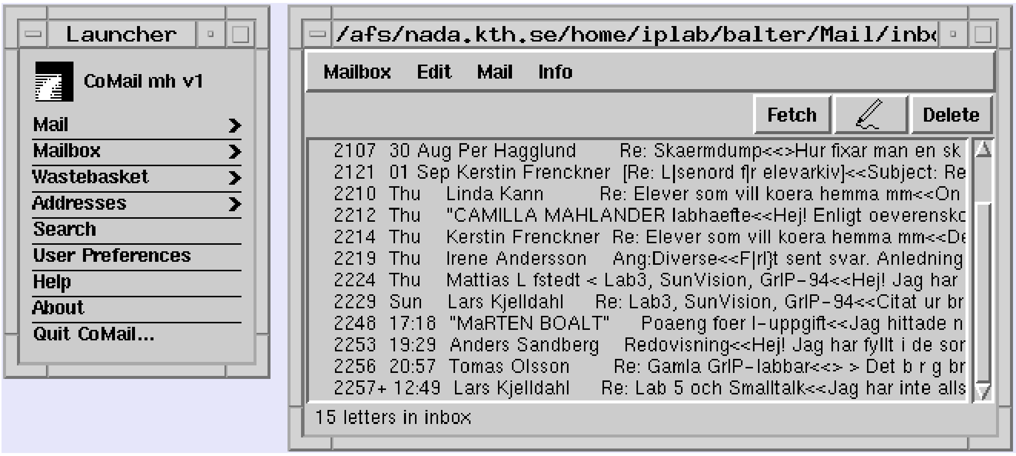
Senast p} onsdag 18/10 vill jag ha ett mail fr}n intresserade.

Kerstin

Figure 5.4: CoMail, C++-version.

The C++-version worked fine but had some disadvantages: The window was divided into two parts, one for the list of messages and one where a selected message could be read or a new one could be written. This means that only one message can be read at the same time. The C++-version could only have one mail folder opened. Messages could be moved between folders, but only via a menu choice, not by drag-and-drop. User interviews clearly stated that a drag-and-drop possibility would be an advantage.

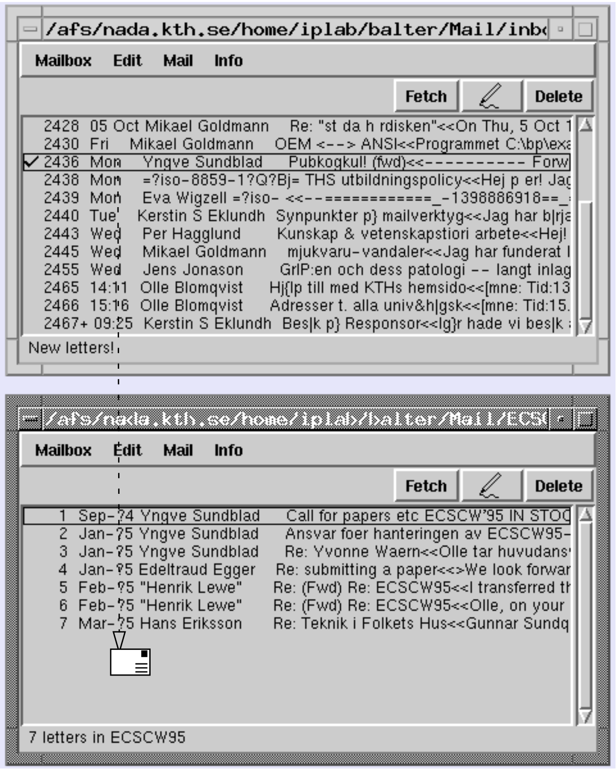
Most of these problems could be more easily solved in the Smalltalk\VisualWorks environment than in C++ with Interviews. A new version was made in Smalltalk that solved these problems, except for drag-and-drop. This first Smalltalk version, see figure 5.5, had a richer functionality than the C++-version and took only two weeks to develop.



```
  2107  30 Aug Per Hagglund      Re: Skaermdump<<>Hur fixar man en sk
  2121  01 Sep Kerstin Frenckner  [Re: L|senord f|r elevarkiv]<<Subject: Re
  2210  Thu    Linda Kann         Re: Elever som vill koera hemma mm<<On
  2212  Thu    "CAMILLA MAHLANDER labhaefte<<Hej! Enligt oeverensko
  2214  Thu    Kerstin Frenckner  Re: Elever som vill koera hemma mm<<De
  2219  Thu    Irene Andersson    Ang:Diverse<<F|rl}t sent svar. Anledning
  2224  Thu    Mattias L fstedt < Lab3, SunVision, GrIP-94<<Hej! Jag har
  2229  Sun    Lars Kjelldahl     Re: Lab3, SunVision, GrIP-94<<Citat ur br
  2248  17:18  "MaRTEN BOALT"     Poaeng foer I-uppgift<<Jag hittade n
  2253  19:29  Anders Sandberg    Redovisning<<Hej! Jag har fyllt i de sor
  2256  20:57  Tomas Olsson       Re: Gamla GrIP-labbar<<> > Det b r g br
  2257+ 12:49  Lars Kjelldahl     Re: Lab 5 och Smalltalk<<Jag har inte alls
15 letters in inbox
```

Figure 5.5: CoMail, Smalltalk version.

There was no support in VisualWorks to simply add drag-and-drop functionality. This problem was rather difficult to solve, it took two weeks of programming before we had a fully working solution. Now messages can be dragged between mail folders in order to move a message from one folder to another, or into the mail editor to add a message to another, see figure 5.6.


/afs/nada.kth.se/home/iplab/balter/Mail/inb
Mailbox
Edit
Mail
Info
Fetch
Delete
/afs/nada.kth.se/home/iplab/balter/Mail/ECS
Mailbox
Edit
Mail
Info
Fetch
Delete

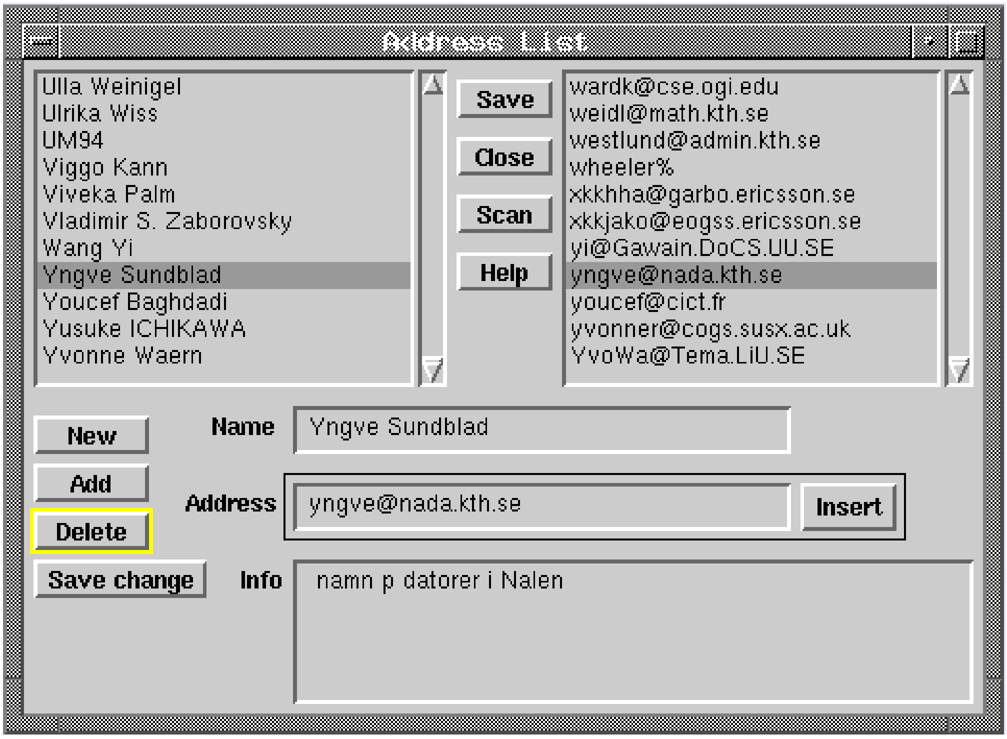
```
  2428  05 Oct Mikael Goldmann   Re: "st da h rdisken"<<On Thu, 5 Oct 1
  2430  Fri   Mikael Goldmann   OEM <--> ANSI<<Programmet C:\bp\exa
✓ 2436  Mon    Yngve Sundblad    Pubkogkul! (fwd)<<------------ Forw
  2438  Mon    =?iso-8859-1?Q?Bj= THS utbildningspolicy<<Hej p er! Jag
  2439  Mon    Eva Wigzell =?iso- <<--============_-1398886918==_
  2440  Tue   Kerstin S Eklundh  Synpunkter p} mailverktyg<<Jag har b|rja
  2443  Wed    Per Hagglund       Kunskap & vetenskapstiori arbete<<Hej!
  2445  Wed    Mikael Goldmann    mjukvaru-vandaler<<Jag har funderat I
  2455  Wed    Jens Jonason       GrIP:en och dess patologi -- langt inlag
  2465  14:11  Olle Blomqvist     Hj{lp till med KTHs hemsido<<[mne: Tid:13
  2466  15:16  Olle Blomqvist     Adresser t. alla univ&h|gsk<<[mne: Tid:15.
  2467+ 09:25  Kerstin S Eklundh  Bes|k p} Responsor<<Ig}r hade vi bes|k
New letters!
```

```
     1  Sep-?4 Yngve Sundblad     Call for papers etc ECSCW'95 IN STOC
     2  Jan-?5 Yngve Sundblad     Ansvar foer hanteringen av ECSCW95-
     3  Jan-?5 Yngve Sundblad     Re: Yvonne Waern<<Olle tar huvudans
     4  Jan-?5 Edeltraud Egger    Re: submitting a paper<<>We look forwar
     5  Feb-?5 "Henrik Lewe"      Re: (Fwd) Re: ECSCW95<<I transferred th
     6  Feb-?5 "Henrik Lewe"      Re: (Fwd) Re: ECSCW95<<Olle, on your
     7  Mar-?5 Hans Eriksson      Re: Teknik i Folkets Hus<<Gunnar Sundq
7 letters in ECSCW95
```

Figure 5.6: Dragging an email message from the inbox to the ECSCW folder.

Address book

From the survey and the interviews it is clear that an advanced address book can help the user to delete the messages that are stored for the purpose of keeping sender address only, and thereby diminish the problem with messages to organise. In CoMail the email address, the name and the subject line are automatically extracted from each opened message. The subject line is extracted to provide the user with a context if the name and the address are not enough to remember the person, see figure 5.7.



Ulla Weinigel
Ulrika Wiss
UM94
Viggo Kann
Viveka Palm
Vladimir S. Zaborovsky
Wang Yi
Yngve Sundblad
Youcef Baghdadi
Yusuke ICHIKAWA
Yvonne Waern

wardk@cse.ogi.edu
weidl@math.kth.se
westlund@admin.kth.se
wheeler%
xkkhha@garbo.ericsson.se
xkkjako@eogss.ericsson.se
yi@Gawain.DoCS.UU.SE
yngve@nada.kth.se
youcef@cict.fr
yvonner@cogs.susx.ac.uk
YvoWa@Tema.LiU.SE

Name: Yngve Sundblad
Address: yngve@nada.kth.se
Info: namn p datorer i Nalen

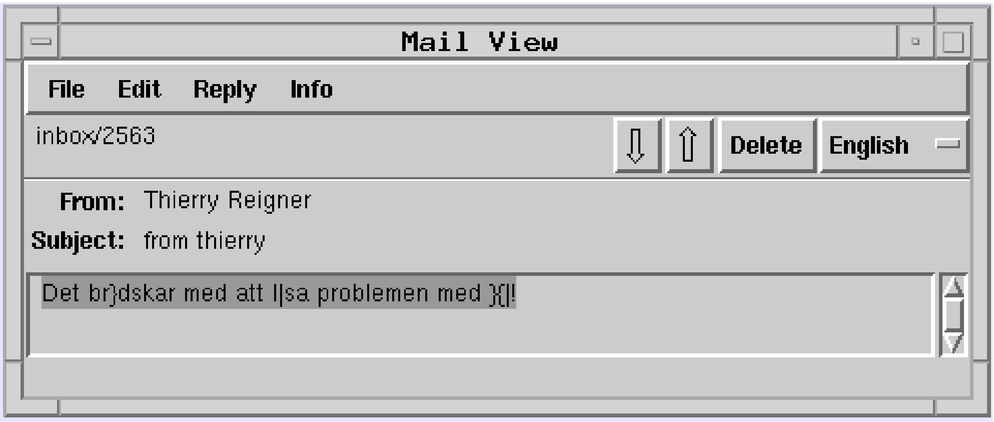
Figure 5.7: CoMail, address book.

The information in the info-field in figure 5.7 can be altered by the user (as well as the address and the name). The insert button can be used to insert a selected address in the current email message under construction. There is no need to scroll through the list of names or addresses. By entering characters on the keyboard, the list will automatically scroll to the first item in the list with those initial characters. We had discussions on whether the address should also be draggable into the email message and agreed that it should, but never implemented it for two reasons:

- Copy and Paste work fine, and would be the natural choice for many users.
- Draggable text was not used in other applications on the Unix platform.

Mail editor

The CoMail message editor was originally used both to read and to write messages, but to provide the user with more feedback a mail view window was added, with a slightly different interface, see figure 5.8.



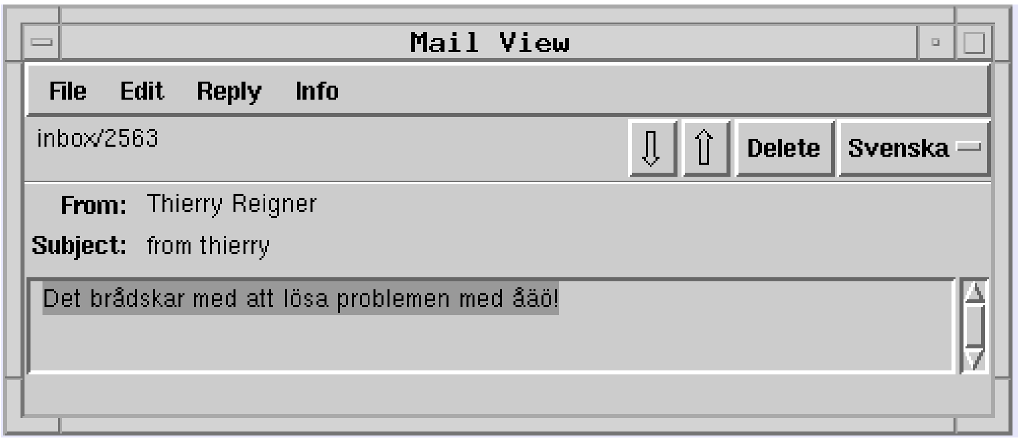
From: Thierry Reigner
Subject: from thierry

Det br}dskar med att l|sa problemen med }{|!

Figure 5.8: CoMail, mail view window.

There are three buttons that enable fast processing of messages wanted by some users with many incoming messages. The arrows in figure 5.8 are used to read the next or the previous message from the current mail folder into the mail view window. The delete button deletes the current message and reads the next message. These buttons have keyboard alternatives as well.

The button labelled “English” can be changed to e.g. “Svenska” and is used for converting incoming Swedish messages with {}\ to åäö, see figure 5.9. This is a first step towards support for the user to convert incoming messages with distorted national characters.

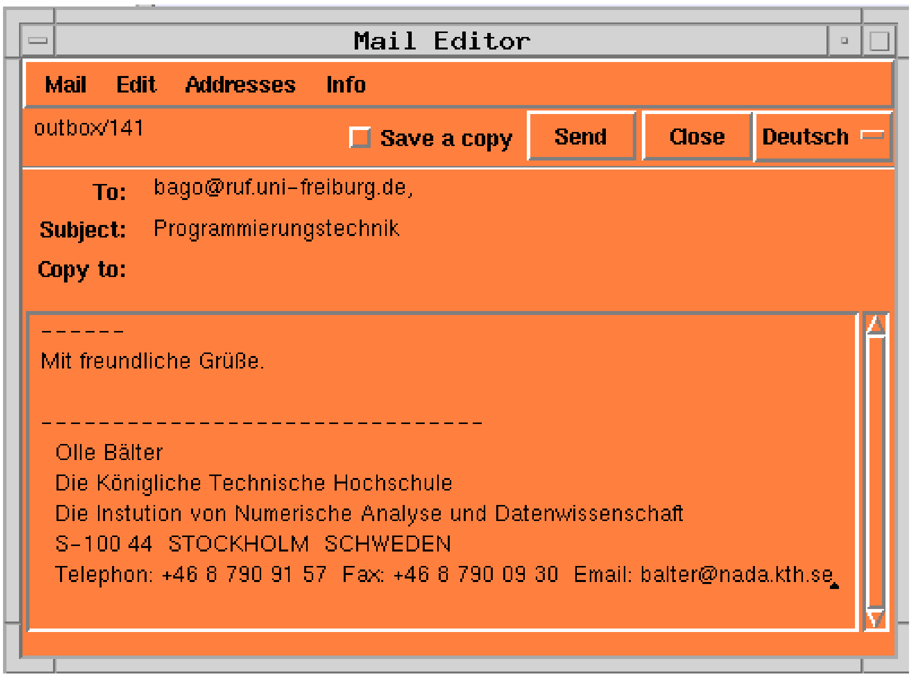


From: Thierry Reigner
Subject: from thierry

Det brådskar med att lösa problemen med åäö!

Figure 5.9: CoMail, mail view window of figure 5.8 after conversion.

In outgoing messages the state of the language button influences the footer. When the button is labelled “English”, an English footer will be inserted with the menu command “Footer”. When the button is labelled “Deutsch” a footer in German is inserted, see figure 5.10.



outbox/141

To: bago@ruf.uni-freiburg.de,

Subject: Programmierungstechnik

Copy to:

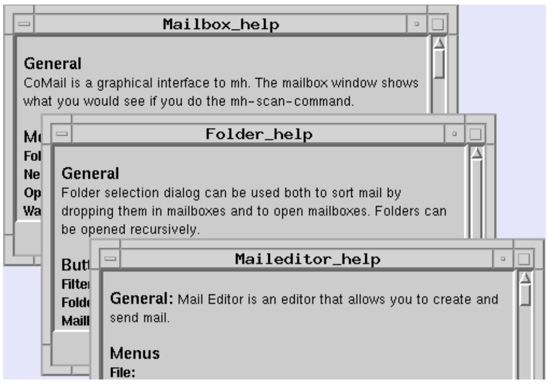
```
------
Mit freundliche Grüße.

------------------------------
 Olle Bälter
 Die Königliche Technische Hochschule
 Die Instution von Numerische Analyse und Datenwissenschaft
 S-100 44  STOCKHOLM  SCHWEDEN
 Telephon: +46 8 790 91 57  Fax: +46 8 790 09 30  Email: balter@nada.kth.se
```

Figure 5.10: CoMail, mail edit window with a German footer.

The converting made by the language button can be extended to convert incoming messages with other problems such as åäö from PCs and Macintoshes. Consistent use of MIME will solve these problems for Internet users, but as long as there are non-MIME mail tools used by email senders, there will be non-MIME messages that will have strange characters.

Help system

The help system in CoMail was built-in from the start, see figure 5.11. At first it only provided the user with a simple explanation of each menu command, but as soon as any user asked a question about CoMail, an explanation of that was added to the help files. The help system is partly context sensitive, there is a separate help for the folder, the mail editor, the address book and the folder list (of figure 5.13).



General
CoMail is a graphical interface to mh. The mailbox window shows what you would see if you do the mh-scan-command.

General
Folder selection dialog can be used both to sort mail by dropping them in mailboxes and to open mailboxes. Folders can be opened recursively.

General: Mail Editor is an editor that allows you to create and send mail.

Menus
File:

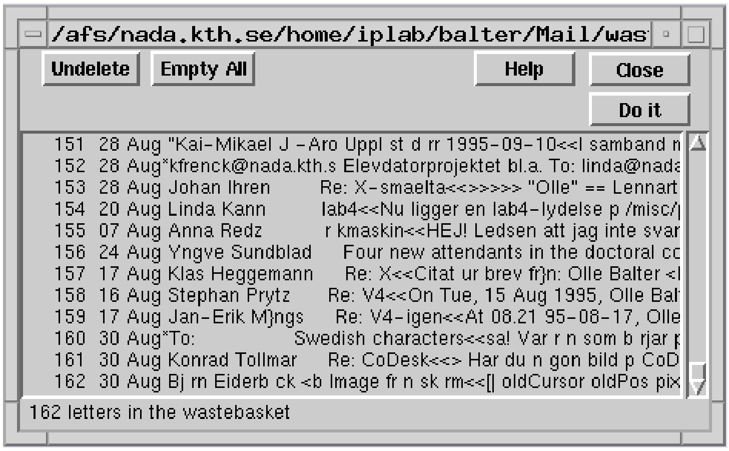
Figure 5.11: CoMail, help system.

Waste basket

The survey and the interviews clearly state the importance of supporting the users to delete messages and thereby lessen the amount of messages to search and sort. One way to help the user to delete messages, is to provide a simple way to undelete deleted messages. This is possible in some mail tools, but only within the same session. Once the program is turned off, the messages are lost.

In CoMail, all deleted messages are moved to a waste basket, see figure 5.12. To actually delete a message the waste basket must be emptied, and even then it is possible via Unix-commands to undelete them before midnight. The idea is that deleted messages should be kept in the waste basket long enough to be stored on a backup tape and thereby it will be possible to undelete any message, even after several years.

The user is ensured that any message can be deleted and still there will be a copy of it somewhere. As time goes by it will be more and more difficult to undelete the message, but it will always be possible to undelete. This may help the user to delete messages and reduce the amount of messages to search in, instead of saving messages “just in case”.



```
151  28 Aug "Kai-Mikael J -Aro Uppl st d rr 1995-09-10<<l samband m
152  28 Aug*kfrenck@nada.kth.s Elevdatorprojektet bl.a. To: linda@nada
153  28 Aug Johan Ihren       Re: X-smaelta<<>>>>> "Olle" == Lennart
154  20 Aug Linda Kann        lab4<<Nu ligger en lab4-lydelse p /misc/
155  07 Aug Anna Redz          r kmaskin<<HEJ! Ledsen att jag inte svar
156  24 Aug Yngve Sundblad     Four new attendants in the doctoral cc
157  17 Aug Klas Heggemann     Re: X<<Citat ur brev fr}n: Olle Balter <l
158  16 Aug Stephan Prytz     Re: V4<<On Tue, 15 Aug 1995, Olle Bal
159  17 Aug Jan-Erik M}ngs     Re: V4-igen<<At 08.21 95-08-17, Olle
160  30 Aug*To:               Swedish characters<<sa! Var r n som b rjar p
161  30 Aug Konrad Tollmar    Re: CoDesk<<> Har du n gon bild p CoD
162  30 Aug Bj rn Eiderb ck <b Image fr n sk rm<<[| oldCursor oldPos pix
162 letters in the wastebasket
```

Figure 5.12: CoMail, waste basket.

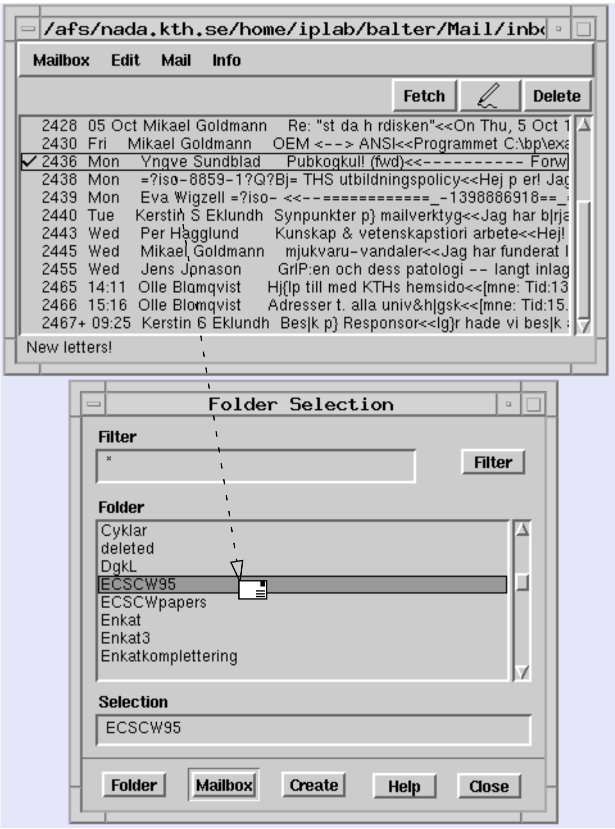
It will be possible to define a folder specific time that defines how long a message should be stored in e.g. the waste basket. Messages stored for a longer period than e.g. a week in the waste basket will then be automatically deleted from the waste basket.

5.3 User evaluation

The user studies of CoMail have been made in three different ways: daily usage by colleagues, specific tasks performed by guests, and a user study made by two computer science students as a part of their Master's thesis [Hellgren & Olsson 1995].

Three colleagues have been using CoMail as their only mail tool for a period of between one month and six months. Those that were already using Smalltalk on a Sun work station could run CoMail without any problems, but to run CoMail from a Macintosh via a terminal program was too slow (this person gave up for this reason after a month). They found errors and their complaints and suggestions are the reasons for, among other things, the following improvements to CoMail:

- It is now possible to drag a message to another folder without opening that folder. The first version of drag-and-drop demanded that both the source and the target folder should be open in order to drag a message from one to the other. With the improved version it is possible to drag messages to the list of folders, see figure 5.13.



```
  2428  05 Oct Mikael Goldmann    Re: "st da h rdisken"<<On Thu, 5 Oct 1
  2430  Fri    Mikael Goldmann    OEM <--> ANSI<<Programmet C:\bp\exa
✓ 2436  Mon    Yngve Sundblad     Pubkogkul! (fwd)<<---------- Forw
  2438  Mon    =?iso-8859-1?Q?Bj= THS utbildningspolicy<<Hej p er! Jag
  2439  Mon    Eva Wigzell =?iso- <<--============_-1398886918==_
  2440  Tue    Kerstin S Eklundh  Synpunkter p} mailverktyg<<Jag har b|rja
  2443  Wed    Per Hagglund       Kunskap & vetenskapstiori arbete<<Hej!
  2445  Wed    Mikael Goldmann    mjukvaru-vandaler<<Jag har funderat l
  2455  Wed    Jens Jonason       GrIP:en och dess patologi -- langt inlag
  2465  14:11  Olle Blomqvist     Hj{lp till med KTHs hemsido<<[mne: Tid:13
  2466  15:16  Olle Blomqvist     Adresser t. alla univ&h|gsk<<[mne: Tid:15.
  2467+ 09:25  Kerstin S Eklundh  Bes|k p} Responsor<<lg}r hade vi bes|k
New letters!
```

Filter: *

Selection: ECSCW95

Figure 5.13: CoMail, dragging a message to the folder list.

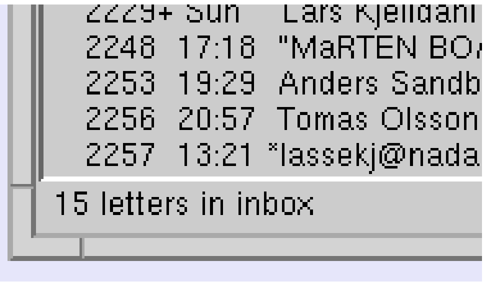
- One of my colleagues had more than 10.000 messages saved. Mostly in order to have the addresses stored somewhere. The address book functionality in CoMail was appreciated, but even if this person opened an old message every 5:th second, it would take almost 14 hours to scan through the old saved messages to collect the addresses. On his request a “scan” function was added to the address book that automatically extracts the name, address and subject from all messages in all folders. The scanning still takes a while, but no-one has to be present.
- The list of messages can be viewed in a proportional font (non-proportional font was the default option in VisualWorks).

The guests that have tried CoMail have been guest researches and interested colleagues outside IPLab. The studies have both been formal and informal. In the formal studies the guest should perform a certain task while thinking aloud and/or being observed. In the informal studies the guests have either seen a demonstration of CoMail or tried it for ten minutes to half an hour without any specific task to perform.

The guests’ complaints and suggestions are the reasons for, among other things, the following improvements to CoMail:

- A button for fast processing of messages. There is now a button in the Mail View window that deletes the current letter and opens the next message in the same folder. This is to help users that receive many messages that mostly should be read and then deleted.
- Improved command names of menu alternatives. In the beginning, all menu commands had the same names as the corresponding MH-commands. This was the natural choice for all MH-users, but commands such as “inc” and “scan” have no meaning for non-MH-users.
- The reply command was divided into “reply” and “reply to all” in order to make it clear when the reply would be sent to all recipients of a message.
- When replying, a part of the text in the message can be selected and then only that part will be automatically included in the reply.
- The included part will be selected so a single key-stroke will erase it, in order to help the user *not* to include by mistake.

The computer science students, Hellgren and Olsson, have used four different methods to evaluate CoMail and CoBoard (see below): *Expert inspection*, *user testing*, *metaphor analysis* and *menu analysis*. They have in particular studied functionality, usability and the structure of the programs. Some of their suggestions were implemented in new versions of CoMail already during their evaluation, such as a feedback line at the bottom of each window, see figure 5.14.



```
2229+ Sun   Lars Kjelldahl
2248  17:18 "MaRTEN BO
2253  19:29 Anders Sandb
2256  20:57 Tomas Olsson
2257  13:21 *lassekj@nada
15 letters in inbox
```

Figure 5.14: CoMail, feedback line

Hellgren and Olsson were especially pleased with the multi threading possibilities (to have several messages and folders opened at the same time), undo possibilities and the help system.

Most of their suggestions for improvements concern the naming and the placement of the menu commands, but there are also other, small but important, improvements suggested: the mail view window should be wider, menu commands that cannot be used should be dimmed, the list of messages should clearly state which messages are unread, it should be faster to start CoMail, more functions should be user tailorable and they also suggest improvements to the help system.

5.4 Suggested improvements to CoMail based on the user studies

The user studies described in chapter 4 have given valuable input to improvements of CoMail. Those that still remain to be implemented are described here.

Interface

It still remains to implement flags that states whether a message has been read, answered and user tailorable flags. The user should have control over all these flags.

Organisation of messages

It still remains to implement an interface to the file system, to enable drag-and-drop between CoMail and the file system; provide new ways to organise messages; an advanced search function, at the moment only cryptic pick commands can be used in one folder at a time; filtering; and support for following dialogue threads.

Waste basket

It will be possible to define a folder specific time that defines how long a message should be stored in e.g. the waste basket. Messages stored for a longer period than e.g. a week in the waste basket will then be automatically deleted from the waste basket.

Size of email flow

It remains to provide default answers that could help users with a large number of incoming messages about the same topic.

Technical problems and practises

There is a need for more support to convert incoming messages with distorted national characters back to the original characters.

Address book

Of the functions for an useful address book mentioned in section 4.7, “Address book” the following are not yet implemented:

- It should be user tailorable whether it should extract addresses and names from: all new messages, only when the user says so, or only from certain kinds of messages.
- It should be possible to save the address book as text, so that it can be read by other programs.
- It should be possible to read a text file with name, addresses and other information, so that an address book can be created by other programs.
- The address book should provide possibilities to define one name for several receivers (alias names).
- There should be advanced possibilities to search in the address bok, e.g for domains.
- There should be advanced support for deleting addresses, e. g. delete addresses older (not used for) two years, delete all addresses from a domain.

Format standards

The only problem of the format standards that is solved in CoMail is that it is possible to read both the MH-format and the Unix mail format. MIME is used in the same extraction as it is in MH. CoMail is running only on the Unix platform.

6 Conclusions

A conclusion of this study is that the email users often perceive problems with email, even more than they admit. There are of course several reasons for this, including lack of desired functionality, bad interfaces, and lack of common technical standards. However, most of the identified problems are solvable, or at least possible to diminish, often with only minor changes in the mail tools.

This chapter starts in 6.1 with some conclusions on what could be done to the mail tools in order to help the user to manage mail. In 6.2 the interface aspects that affects usability are discussed and in 6.3 the technical practices that affect usability are discussed. The chapter is concluded with some advice on the design of mail tools in 6.4.

6.1 User aspects of managing email

The usage of email is a dynamic and *evolutionary* process. The problems that users perceive come, change, and disappear. When the first basic skills of writing, sending and reading messages are mastered, there are new challenges for the user, e.g. how to handle the increasing number of incoming messages, and how to store addresses. The users get problems and then solve them in some way, or learn to live with them, develop their usage further, and find new problems.

The usage and learning of an email tool differ from e.g. the usage and learning of a word processor. Since both incoming and outgoing messages often are stored, the number of stored messages will increase and eventually something must be done to handle the stored messages. A strategy for handling email that works in the beginning may not work later when the messages are too many. The process that changes the situation and makes a strategy useless may take several years.

A word processor has some basic functionality necessary for short documents and some that are useful only for long documents or books. The fact that the user has written and read a lot of documents will not change anything in the use of the word processor or make it more difficult to use due to the many written documents. Many written documents will of course affect the use of the file system.

There is some support for this evolutionary hypothesis in this study. The users at the university that were interviewed, were interviewed between three and five months after they had answered the questionnaire. None of the interviewees had fewer messages stored (despite some attempts to “tidy up” before the interview), but many had more messages stored. The rate of incoming messages at the time of the interview was about the same. The two non-technical users that were interviewed did not perceive any problems with the storage of messages; but they had used the mail systems at their sites a shorter time than the subjects at the university.

A similar difference exists between the university and the industry described in section 4.3; more messages are stored at the university where email has been used longer.

Occasional questionnaires and interviews are therefore not enough to follow this evolutionary process. The users must be studied for a long consecutive time in a longitudal study. In |Mackay 1988] email users are divided in *prioritizers* and *archivers*. A prioritizer concentrates on the problem of managing incoming messages. An archiver concentrates on how to archive messages for subsequent use. An interesting study would be to study how users develops into prioritizers or archivers and what factors influence this development. There might be several stages that email users go through.

Size of email flow

There is a correlation between the number of perceived problems and the ratio between the number of incoming messages a day and the number of messages in the inbox. Those that perceive problems have a lower ratio compared to the users that do not perceive problems. This implies that problems are caused by too much work (not enough time to process the incoming messages) and/or that it is important to limit the number of messages in the inbox.

The problem with too much work cannot be solved by a better mail tool, but a better mail tool could provide the user with possibilities to process email faster and thereby provide the user with more time.

The number of messages in the inbox could be decreased with better support for processing messages fast. Possibilities for one-button keyboard commands is one way of supporting the user. Filtering can be used so separate out messages with low priority e.g. from distribution lists. Default answers could help users with a large number of incoming messages about the same topic.

A large number of stored messages can cause problems with overview. Too many messages in the inbox can make the user uncomfortable. These problems can be solved or diminished by better support for *organising*, *searching* and *deleting* messages.

Organisation of messages

Organisation is a main factor in coping with and causing email problems. In this study 100% of those that perceived problems had problems with searching and/or storing of messages. Of those that answered that they did not perceive problems, an additional 33% had problems with searching and/or storing of messages. Even though it is possible to organise messages in folders, it might be too difficult to move the messages. In all the mail tools used in this study, messages could be stored in different folders, but none of the tools fully supported drag-and-drop both within the mail tool and between the mail tool and the file system.

Organisation can be supported by improving the current methods, e.g with drag-and-drop both within the tool as between the tool and the file system, search that works in all folders and user controlled filtering.

There might be better ways than the traditional folders to organise messages. All messages could be stored in a database and the mail tool could provide different views towards this database. One view could be the sender of the messages, one could be the subject, one could be a list of all messages that contain a user defined keyword.

More support to follow dialogues can also be useful. Links between messages in a dialogue can provide another structure of messages, similar to e.g. the COM system [Palme 1995 b] and the Mona system [Cockburn & Thimbleby 1993].

Arrival sorting, when incoming email messages are automatically sorted into different folders or prioritised within a folder, may be one way to help the user organise. The varying demands from the users that want to have arrival sort indicates that *how* this should be done must be very tailorable by the user, e.g. if the messages should be moved directly to the folder or if all should appear in the inbox first. The automatic sorting could in that case be a default option that the user can accept or override.

Deleting messages is one way of diminishing the problems with organisation and overview. Some users do not want to delete messages because they might delete the wrong message, or might regret the decision to delete later. If the messages were moved to a waste basket and kept there instead, the user would be assured that the messages are possible to undelete in the future, not only during the current session. In systems that automatically backup files (e.g. Unix systems), this would give enough time to store the messages on backup tape and thereby allow the user to undelete a message even after several years. The messages could then be removed from the waste basket after a user specific time, e.g. a week.

Searching

Email users need more support to search for messages, not only in the currently opened folder, but in all folders. When the user knows which folder a message is stored in, the need to search for it is limited to those folders that are too large to overview. Eudora’s way of solving this problem with a possibility to skip large folders, while still searching in the background is excellent.

Flags that mark that messages are e.g. not read or answered could be added to CoMail. This would give the user additional possibilities to sort messages. The user must have full control of these flags, i.e. possibilities to raise and set these flags manually.

When there are too many messages in a folder to get an overview, simply changing the presentation order may be enough. For a thorough study of the problems with searching and sorting, see [Reignier 1995 and forthcoming].

Deleting helped by the address book

Many messages are stored only in order to save the address. These messages will clutter the mail folders. An advanced address book with possibilities to automatically extract the name and the address from all messages would help the user to delete more messages, see “Organisation of messages” in section 4.7 for details.

The CoMail address book could be extended with search possibilities for domain names and sorting after frequency (number of messages to and from an address). The alias possibilities in CoMail today are separated from the address book, but could be used as a grouping facility in the address book.

6.2 Interface aspects that affect email usability

Maybe surprisingly, there is no difference in the amount of perceived problems between users of the different interface types used in this study (text, text with menus, and graphical). The direct manipulative mail tools used in this study support drag-and-drop for messages, but only within the mail tool or between the mailtool and the file system, not both. Further studies with users that have such tools would be interesting.

Thus, not even on the Macintosh, the email programs used in this study fully support drag-and-drop. On command based Unix computers and old PCs, this lack of direct manipulation can be excused with the argument that drag-and-drop is not a part of the metaphor. On the Macintosh and PCs with Windows this lack of direct manipulation is even a violation of the interface metaphor. This lack may contribute to the problems with organisation of messages. A user friendly mail tool could fully support drag-and-drop, as well as other ways to do the same thing without the mouse.

A fully direct manipulative interface is not enough. Email users will eventually use a different computer, e.g while travelling, where a graphical interface is not possible to run. Therefore, there must be at least a limited text interface to the mail files, so that the user can read and send mail from a text terminal.

The possibility to view several messages and several folders at the same time, preferable each in a separate window, is important. This simplifies overview and gives the user possibilities to read and write messages simultaneously. This is at present only implemented in tools with a graphic interface, but could also be available for text interfaces. This overview can be supported with flags or similar to show which messages that are read, new, answered and so forth.

The highest priority in a program should always be the user. When the user deletes a message, the message could disappear from the screen at once (if the user so prefers) and the control should be given back to the user to let the user continue with other tasks. The actual removing of the message could be done in the background, as the Eudora background search. This possibility to give the user the highest priority could be used in many other situations: when messages are moved to other folders, when addresses are extracted from messages (this is implemented in CoMail), and when messages are sent. The user should seldom have to wait for the program.

Flaws in the interface are annoying, regardless of application. The mail tools in this study often have inconsistencies such as same name for different commands or different names for the same command. This causes the user to make unnecessary mistakes and unwillingness to explore the possibilities in the mail tool. Formal user studies, such as the study of CoBoard and CoMail [Hellgren & Olsson 1995] where they e.g. studied the names of different and similar commands, the placement in menus, and the effect of various commands; can point out such flaws for the developers, so that the users having to face them is avoided.

Adaptation to the mail tool

In this study, the users show a great capacity of working around the problem when they do not find any solution within the mail tool. Some users find other programs to use to search in the messages when the search functionality in the mail tool is inadequate. Other users simply give up and avoid problems by not using certain functions or even avoid using email in certain situations. For example they use only one folder, even though they want to have several folders, if it is difficult to get an overview of folders or to move messages between the folders. They avoid to read email from home because the mail formats of the mail tool at work and at home are incompatible.

Users learn the inconsistencies in their mail tools, and between mail tools, and adapt their management of email to the inconsistencies; e.g. users write aa, ae and oe or a a o instead of å, ä and ö when they know that the receiver will receive the national characters distorted.

6.3 Technical practices that affect email usability

The lack of common technical standards affect usability in ways visible in most mail tools. Often there are simple technical solutions that are not used. The main technical limits discussed in this study are associated with *character codes*, *mail file format*, *prerequisite of drag-and-drop* and *disconnected usage*.

Coded character sets

National characters such as åäö are often distorted when email messages are sent between different computer types or between different networks. One obvious solution to this is to use only one computer type, one mail tool and only use email within a local network. This is a policy in some companies. A comparison of email usage at an industrial site and a university [Bälter & Lantz 1995] indicates that the percentage of users at the university site that perceive problems is higher than at the industrial site. There are several explanations:

1. More external communication at the university site
2. The diversity in computer types at the university site
3. The diversity in mail tools at the university site

All these factors, in combination with the lack of MIME usage, causes national characters to be distorted and some attachments to email to be unreadable.

The solution with a company policy may seem attractive, but it may be a short term solution. Besides that many people will be frustrated when they are forced to work with computers and programs that they have not chosen themselves, they will still get problems with email; when the email communication outside the company grows (e.g. via Internet), the same problems will appear. This problem can be solved by using an international standard, which on the Internet is MIME.

Mail file formats

Most mail tools use their own mail file format and cannot read mail files with other file formats. E.g. Eudora cannot read mail files created with Mailstrom. This will work for some time, but no mail tool will last forever. Users will change mail tools and maybe have thousands of messages stored that must be transferred to the new mail tool. Users may want to use several mail tools simultaneously, to choose the best interface or functionality depending on the computer type available or the current task. A user friendly mail tool could therefore be able to read several mail formats and preferably run on all platforms.

The possibility to read several mail formats is also a solution for graphical mail tools that need an interface that can be used on a text terminal. There are already several text based mail tools on most computers.

The problems with specialised file formats are pertinent not only for the messages but also for the address books that also should be possible to transfer between different mail programs. Otherwise, users that know that they will change mail tool someday still have to save all messages in order to have the addresses stored even after they have changed mail tool.

In CoMail it is possible to read the Unix mail mbox format as well as the MH format. The problem with moving messages between mbox files has not yet been solved satisfactory, and the address book should of course be possible to save in a text format.

Prerequisite of drag-and-drop

Drag-and-drop is an interface aspect, but drag-and-drop is also a technical problem. Drag-and-drop is difficult to implement seamlessly between applications and even more difficult between platforms. However, at least on the same platform, the efforts needed by the developers to do this will pay off for the user, since it helps in organisation of email messages, which is a main factor in perceived problems.

Disconnected usage

When email is used via a modem, the phone cannot be used at the same time. When the call to the mail server is long distance, it might be frustrating and expensive to wait for the commands to be performed on the mail server. The IMAP4 protocol will support both disconnected usage, and possibilities to read messages (old and new) from several terminals. With IMAP4, the user may call the mail server to fetch the new mail, disconnect, read the mail, delete and organise while using the phone, connect again and just transfer the necessary commands to the mail server that handles the commands as a batch job.

6.4 Some advice on design of email tools

Here are some conclusions on design drawn from this study:

Users have many valuable suggestions to improvements in their mail tools. Obvious mistakes, which make the mail program prioritised higher than the user, or when not even computer scientists can understand the manual or the on-line help, can easily be identified when users are involved in the development process.

Informal user studies are valuable for finding suggestions for functions in an application. Formal user studies are valuable for finding flaws and inconsistencies in menus and commands, see e.g. [Hellgren & Olsson 1995].

Real usage by non-developers (of the tested system) points at different types of problems. This is in line with [Twidale et. al. 1993] who believe that it is more productive to heed the particular demands from a user performing a real task rather than follow general suggestions from people observing a demonstration of the system.

Our conclusion is that real usage give the single most important information on how to form a usable application. Due to the evolutionary usage of email, problems will occur in different stages for the users. By following users in different stages and handling users' problems a tool can be developed into a more usable tool.

7 References

Akin O. & Rao R.: *Efficient Computer-User Interface in Electronic Mail Systems*. International Journal of Man-Machine Studies, 1985, 22, pp 589-611.

Anderson J.: *Cognitive Psychology and its Implications.* W.H. Freeman and Company, New York 1990, pp 150-156.

Anderson R. & Gillogly J.: *Rand Intelligent Terminal Agent (RITA): Design Philosophy.* Rand Corporation Technical Memorandum R-1809-ARPA, February, 1976.

Bair J.: *The Impact of Office Automatisation*, The Office of the Future, Uhlig, Farber, Bair, North-Holland 1979.

Bannon L.: *CSCW: An Initial Exploration.* Scandinavian Journal of Information Systems, August 1993, Vol.5, pp 3-24

Benbasat I. & Todd P.: *An experimental investigation of interface design alternatives: icon vs text and direct manipulation vs menus*. International Journal of Man-Machine Studies, 1993, 38, pp 369-402.

Borenstein N. & Freed N.: *MIME-Mechanisms for Specifying and Describing the Format of Internet Message Bodies.* Network Working Group, Request for Comments: 1341. June 1992.

Borenstein N.: *Computational Mail as Network Infrastructure for Computer-Supported Cooperative Work*, CSCW92 proceedings, pp 67-74.

Borenstein N. & Thyberg C.: *Cooperative Work in the Andrew Message System*, Proceedings of CSCW 88, pp 306-323.

Borenstein N. & Thyberg C.: *Power, ease of use and cooperative work in a practical multimedia message system*. International Journal of Man-Machine Studies, 1991, 34, pp 229-259.

Bälter O. & Lantz A. : *What causes problems with email?* Proceedings of Information Technology Interfaces 1995, pp 179-186.

Cockburn A. & Thimbleby H.: *Reducing User Effort In Collaboration Support*. Proceedings of 1993 International Workshop on Intelligent User Interfaces, pp 215-218

Communications of the ACM: 95:4, pp 9-10, April 95.

Dix A., Finlay J. , Abowd G. & Beale R.: *Human-Computer Interaction*. Prentice-Hall 1993. ISBN 0-13-458 266-7.

Eriksson P.: *Datorbaserad information i småföretag: meddelande och informationssystem i amerikanska datanät*. (in Swedish) Teldoc, Report No. 15, 1985.

Furht B.: *Distributed Multimedia Systems and Applications*. Tutorial at International Conference on Computer Communication, 1995, Seoul, Korea.

Fältström P. 1994: *Internet Mail, Test report from Interoperability 94,* http://www.snus.se/SNUS/

Fältström P. 1995: *Internet Mail, Test report from Interoperability 95,* http://www.snus.se/SNUS/

Goldberg Y., Safran M. & Shapiro E.: *Active Mail–A Framework for Implementing Groupware,* CSCW 92 Proceedings, pp 75-83.

Gould J.: *How to Design Usable Systems*. Handbook of Human-Computer Interaction, edited by Helander, Elsevier Science Publishers B. V. (North-Holland), 1988, pp 757-789.

Heckel P.: *The Elements of Friendly Software Design*, Sybex Inc., Alameda CA, USA, ISBN 0-7821-1538-1. 1993 (reprint, first version 1982).

Hellgren T. & Olsson M.: *Some Methods for Evaluating user Interfaces and an Evaluation of CoMail and CoBoard*. (in Swedish) TRITA-NA-E9530. NADA 1995, Royal Institute of Technology, Stockholm, Sweden

Hiltz S.R. & Turoff M.: *Structuring computer mediated communication systems to avoid information overload*. In Communications of the ACM. vol. 28, No. 7, (pp 680-689).

Hjalmarsson A., Oestreicher L. & Waern Y.: *Human Factors in Electronic Mail System Design*. Behaviour & Information Technology, 1989, vol. 8, no 6, pp 461-474.

Houde S.: *Iterative Design of an Interface for Easy 3-D Direct Manipulation*. Proceedings of CHI’92. pp 135-141.

Hägglund P.: *CoBoard*, forthcoming IPLab report.

Ives B.: *Graphical user interfaces for business information systems*. MIS Quarterly, Special Issue, December 1982, pp 15-42.

Janlert L-E.: *Människa-Dator interaktion,* (in Swedish) pp 80-81, 1991, Umeå University, Sweden.

Java WWW: http://java.sun.com/whitePaper/java-whitepaper-1.html

Jeffries R. & Rosenberg J.: *Comparing a Form-Based and a Language-Based User Interface for Instructing a Mail Program*. Proceedings of CHI+GI 1987, pp 261-266.

Jones S., Bock G. & Brassard A.: *Using Electronic Mail: Themes Across Three User Interface Paradigms*.SIGCHI Bulletin Jan. 1990, Vol. 21, No 3 pp 45-48.

Järnefors O.: *70 ways (or so) of (mis)representing Swedish letters*. ftp://ftp.admin.kth.se/pub/misc/ojarnef/internet/aao-errors-memo.txta

Killander F.: http://www.dsv.su.se/~fk/if_Doc/IntFilter.html, 1995.

Lai K.-Y. & Malone T: *Object Lens: A “Spreadsheet” for Cooperative Work,* Proceedings of CSCW 88, pp 115-124.

Lantz A.: *"Tunga" användare av datorpost*. (in Swedish) IPLab report 90. NADA 1994, Royal Institute of Technology, Stockholm, Sweden

Linton M., Calder P. & Vlissides J.: *InterViews: A C++ Graphical Interface Toolkit*. Technical Report CSL-TR-88-358, Stanford University, CA, USA, July 1988.

Mackay W. 1988: *More Than Just a Communication System: Diversity in the use of Electronic Mail*. Proceedings of ACMs CSCW'88: Computer Supported Cooperative Work, pp 215-218.

Mackay W., Malone T., Crowston K., Rao R., Rosenblitt D. & Card S.: *How do Experienced Information Lens Users Use Rules?*, Proceedings of the ACM conference on Human Factors in Computing Systems, April 1989.

Mackay W. 1990: *Users and Customizable Software: A Co-Adaptive Phenomenon*. Doctoral dissertation, Sloan School of Management MIT.

Maes P.: *Learning Interface Agents*. ACM-SIGCHI International Workshop on Intelligent User Interfaces, January 1993.

Malone T., Lai K.-Y. & Fry C.: *Experiments with Oval: A Radically Tailorable Tool for Cooperative work.* Proceedings of CSCW´92, pp 289-297.

Malone T., Grant K., Turbak F., Brobst S. & Cohen M.: *Intelligent Information sharing systems.* Communications of the ACM, Vol. 30, No 5, May 1987, pp 390-402.

man Sun 4.1: Manual pages from Sun version 4.1.

Mander R., Salomon G. & Wong Y. Y.: *A 'Pile' Metaphor for Supporting Casual Organization of Information.* Proceedings of CHI'92. pp 627-634.

Network Wizards: http//www.nw.com.zone/WWW/report.html. 1995

Nielsen J.: *Usability Engineering*. Academic Press, Boston, MA, USA, 1993.

Nordic 1992: *Nordic Cultural Requirements on Information Technology, Summary report*, Libris 1992, ISBN 9979-90004-3-1.

Norman D.: *The psychology of everyday things*, pp 99, Basic Books, New York, 1988.

Paavonen W.: *Arbete på distans*. (in Swedish) R1992:6. Department of Business administration, University of Stockholm 1992.

Palme J. 1995 a: *KOM-95 user interface screens.* Department of Computer and System Sciences. Stockholm University., 1995.

Palme J. 1995 b: *Electronic mail*, ISBN 0-89006-802-X. Artech House, Inc. Norwood, MA, USA, 1995.

Palme J. WWW1: gopher://mars.dsv.su.se:70/00/dsv-reports/research-reports/messaging-research/History-of-com.rtf, 1995.

Palme J. WWW2: *Email and computer conferencing functions*, gopher://mars.dsv.su.se/11/dsv-reports/research-reports/messaging-research/e-mail-checklist, 1995.

Patton M.Q.: Qualitative Evaluation methods. Beverly Hills, California Sage cop, 1980.

Pehrson B. & Sundblad Y.: *Multi-G—Slutrapport.* (in Swedish) IPLab report 68, Multi-G no 9, TRITA-NA-P9404. NADA, Royal Institute of Technology, Stockholm, Sweden

Petroski H.: *Form Follows Failure, in The evolution of useful things,* Alfred A Knopf, New York 1992.

Pliskin N.: *Interacting with electronic mail can be a dream or a nightmare: a users point of view.* Interacting with computers vol. 1 no 3 (1989), pp 259-272.

Poltrock S. & Grudin J.: *Computer-Supported Cooperative Work and Groupware*, Tutorial at CHI´94, pp 355-356.

Reignier T.: *Sorting and Searching Functionalities for Mail Tools*. Forthcoming IPlab report. NADA, Royal Institute of Technology, Stockholm, Sweden

Reignier T.: *Review of sorting and searching functionalities of several mail systems*. IPlab report 96. NADA, Royal Institute of Technology, Stockholm, Sweden, 1995.

RFC 822: *Standard for the Format of ARPA Internet Text Messages*. Revised by David H. Crocker. August 1982 University of Delaware, Newark, DE, USA.

RFC 1800: URL:http//ds.internic.net/rfc /rfc1800.txt

Rose M & Romain J: *How to process 200 messages a day and still get some real work done*. http://www.cs.indiana.edu/docproject/mail/mh.html.

Severinson Eklundh K. 1986: *Dialogue processes in computer-mediated communication: A study of letters in the COM system*. Linköping Studies in Arts and Science 6, University of Linköping.

Severinson Eklundh K. 1994: *Electronic mail as a medium for dialogue*. In L. van Waes, E. Woudska & P. van den Huven (Eds.), *Functional Communication Quality.* Ulrecht Studies in Language and Communication. Rodopi Publishers, Amsterdam/ Atlanta.

Severinson Eklundh K. 1995: *A dialogue perspective on electronic mail: implications for interface design*. IPLab report 75. NADA, KTH, Stockholm, Sweden. To appear in: van Oostendorp (ed.) *Cognitive aspects of electronic text processing*. Ablex Publishing Company, Norwood, N.J., USA 1995.

Sheth B. & Maes P.: *Evolving Agents for Personalized Information Filtering*. Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications 1993.

Shneiderman B.: *Direct Manipulation A step beyond Programming Languages.* IEEE Computer 16(8), August 1983.

Shneiderman B.: *Designing the User Interface. Strategies for Effective Human-Computer Interaction.* Addison-Wesley 1987.

Sproull L. & Kiesler S.: *Connections – New ways to work in the networked organisation.* MIT Press 1991, Cambridge MA, USA.

SUNET Sunetten (in Swedish), no 2, pp 1, April 1995.

Söderhielm M: *Change in Cyberspace: The Integration of Online Services with the Internet*. pp 5,7,9,25,28,31,42,52. IT 94/34. Department of Teleinformatics, Royal Institute of Technology, Stockholm, Sweden.

Tollmar K. & Sundblad Y.: *The Collaborative Desktop –Experience from building and designing an environment for CSCW.* COMIC Report KTH-4-16, May 1994.

Tollmar K. & Sundblad Y.: *The Design and Building of the Graphic User Interface for the Collaborative Desktop.* Computers and Graphics. Vol. 19, No 2, pp 179-188, 1995.

Twidale M., Rodden T. & Sommerville I.: *Developing a tool to support collaborative dialogues and graphical representation of ideas*, Proceedings of NATO ARW, April 1993, Segovia, Spain. In Verdejo and Cerri (Eds) Collaborative dialogue technologies in distance learning. Berlin: Springer-Verlaag. pp 219-235.

Unicode 1991: The UNICODE Standard; "*Worldwide Character Encoding*", Version 1.0, Volume 1, The UNICODE Consortium, Addison-Wesley, 1991, 0-201-56788-1.

Unicode 1992: The UNICODE Standard; "*Worldwide Character Encoding*", Version 1.0, Volume 2, The UNICODE Consortium, Addison-Wesley, 1992, 0-201-60845-6.

Unicode 1993: The UNICODE Standard; "*UNICODE Technical Report #4*", Version 1.1, The UNICODE Consortium, Prepublication Edition, 1993.

Westergren A.: *Elektroniska meddelande och konferenssystem.* (in Swedish) UNINF 163.89, ISSN 0348-0542. Institute of Information Processing, University of Umeå, Sweden.

Xephon: *Who is using the Internet?* http://www.hiway.co.uk/xephon/internet.html (May 1995).

A Appendices

A.1 Industrial questionnaire

(Translated from Swedish)

22 QUESTIONS ABOUT YOUR ELECTRONIC MAIL

This questionnaire is a part of ...

In the research project “Communication patterns in computer supported cooperative work”, at IPLab, Royal Institute of Technology in Stockholm, we are interested in how electronic mail is used.This questionnaire is the first of two parts. In the next part, interested people will be interviewed during one hour.

This questionnaire has 24 questions. Each question has different alternatives and sometimes a line where you can write comments to your answer. We estimate the time to answer the questions to 30 minutes.

All answers will be treated confidently. When results are published, data will be presented by group and not by individual.

The best way to answer is to use the “reply” command, the answer will then automatically be sent to alz@nada.kth.se.

If you wish to return the answer in paper format you send it to

Ann Lantz

IPLab

NADA, KTH

100 44 Stockholm.

If you have any questions, please contact Ann Lantz: at 790 62 81 or via electronic mail to alz@nada.kth.se.

----

1. What is your profession (title)?

2. Describe your main work task.

3. I mainly use the following computer:

_ PC (DOS)
_ PC(MS Windows)
_ Macintosh
_ Sun work station
_ Other:

Comment:

4. Which electronic mail system do you use?

_ mail

_ Mailstrom

_ Eudora

_ Pine

_ Mail Tool V3

_ Other:

Comment:

5. How did you learn to use that electronic mail system?

_ From a teacher in a class

_ From a study/work colleague

_ By written material e.g. manual

_ By on line documentation/help

_ Other:

6. For how long have you been using electronic mail?

7. How often do you enter the email system?

_ Continuously. I read the mail when it arrives.

_ Several times a day.

_ Once a day.

_ Several times a week.

_ Once a week

_ More seldom.

_ Other:

Comment:

8. What do you use the electronic mail system for?

Try to order the different alternatives by putting a 1 for the most common alternative, 2 for the second most common and so on.

_ To receive information on what to do in my work.

_ Communicate with others.

_ Order services from others

_ Find relevant information for my work

_ Exchange documents (text files) with others.

_ Other:

Comment:

9. Try to estimate how many email messages that you receive and send the days you use the electronic mail system.

I receive ___ messages.

I send ___ messages.

Comment:

10. Try to estimate how many messages you receive that are private and from distribution lists respectively

Mark with an X on the line below. If you e.g. think that you receive half of your messages from distribution lists, you put your X in the centre of the line.

Mostly from distribution lists Equal amount Mostly private messages

I------------I------------I------------I-------------I

11. Try to estimate how many messages you send that are private and from distribution lists respectively

Mark with an X on the line below.

Mostly to distribution lists Equal amount Mostly private messages

I------------I------------I------------I-------------I

12. How do you handle email addresses of people that you communicate with (several alternatives are possible)?

_ Use old messages.

_ Use an electronic address book.

_ Use a paper address book.

_ Other:

Comment:

13. What do you do with the email you receive?

_ Save everything.

_ Save certain mail:

_ Delete everything.

_ Other:

Comment:

14. Try to estimate how many messages you have in your electronic mail box.

_ It is empty.

_ 1-10 messages.

_ 11-50 messages.

_ 51-100 messages.

_ 101-250 messages.

_ 251-500 messages.

_ More than 500 messages.

Comment:

15. Try to estimate how many messages you have in different folders or files on your computer.

_ No saved messages.

_ 1-10 messages.

_ 11-50 messages.

_ 51-100 messages.

_ 101-250 messages.

_ 251-500 messages.

_ More than 500 messages.

Comment:

16. How can you handle your electronic mail today?

1) To handle my email has never been a problem.

2) I have some problems with my email, but I have not solved them yet.

3) I have had problems with my email, but I have found strategies to handle it.

4) I am close to not being able to handle my email.

5) I am close to not being able to handle my email, despite having tried different strategies.

6) I cannot handle my email.

Comment:

If you answered question 16 with alternative 1 “To handle my email has never been a problem”, please continue with question 19.

17. Comment with your own words the problems and difficulties that you have had with your electronic mail based on your answer to question 16.

Try to describe your situation based on

a, private messages,

b, messages in mailing lists (directed to several persons under the same address)

c, postings in conference systems (e.g. Usenet News).

a, private messages,

b, messages in mailing lists (directed to several persons under the same address)

c, postings in conference systems (e.g. Usenet News)

18. In which of the following situations have problems occurred?

_ When I shall receive information on what to do in my work.

_ When I communicate with others.

_ When I should order services from others

_ When I want to find relevant information for my work

_ When I should exchange documents (text fles) with others.

_ Other:

19. If there was a possibility to automatically sort arriving messages (in different folders, e.g. one for a mailing list, one for messages from a certain person), would you use that?:

_ Yes.

_ No.

_ Uncertain.

Comment:

20. In which order do you read your mail (several alternatives possible)?

_ Presentation order

_ Private messages first

_ Distribution lists first

_ Select based upon the sender

_ Select based upon the subject

_ Select after size of the message

Comment:

21. Do you have a need to read and write several messages at the same time?

_ Yes.

_ No.

Comment:

22. Do you think that the possibilities to perform selections are adequate in your mail tool?

_ Yes.

_ No.

Comment:

23. Do you think that a direct manipulative interface would be an advantage when you handle email (to drag messages to folders, attach documents to email messages)?

_ Yes.

_ No.

_ Uncertain.

Comment:

24. In what way would multimedia messages (audio, video and pictures) add something positive to your communication?

Are you interested in, and do you have the time, to be interviewed during one hour regarding electronic mail (at your work site)

_ Yes, I can be interviewed.

Phone:

Email address:

_ No, I cannot be interviewed.

Thank you for answering this questionnaire!

A.2 University questionnaire

(Translated from Swedish)

22 QUESTIONS ABOUT YOUR ELECTRONIC MAIL

These questionnaire is a part of the development of CoMail, a mail program developing at IPlab. There is also a research project “Communication patterns in computer supported cooperative work”, where we study how electronic mail is used. This questionnaire is the first of two parts. In the next part, interested people will be interviewed during one hour.

This questionnaire has 22 questions. Each question has different alternatives and sometimes a line where you can write comments to your answer. We estimate the time to answer the questions to 25 minutes.

All answers will be treated confidently. When results are published, data will be presented by group and not by individual.

The best way to answer is to use the “reply” command, the answer will then automatically be sent to balter@nada.kth.se. If you wish to return the answer in paper format you can put it in Olle Bälter’s mail box at IPlab.

If you have any questions, please contact Olle Bälter or Thierry Reigner on 790 91 57 or by email to balter@nada.kth.se or thierry@nada.kth.se

----

1. What is your profession?

_Researcher

_Teacher

_Research student

_Secretary

_Other:

2. Describe your main work task.

3. I mainly use the following computer:

_ PC (DOS)

_ PC(MS Windows)

_ Macintosh

_ Sun work station

_ Other:

Comment:

4. Which electronic mail system do you use?

_ mail
_ Mailstrom
_ Eudora
_ Pine
_ Mail Tool V3
_ mh
_ Other:

Comment:

5. How often do you enter the email system?

_ Continuously. I read the mail when it arrives.
_ Several times a day.
_ Once a day.
_ Several times a week.
_ Once a week
_ More seldom.

Comment:

6. How many email messages have you received and sent respectively in the last 7 days?

I have received ___ messages.

I have sent ___ messages.

Comment:

7. How large amount (in percent) of your email is private (that is addressed to you and not to some distribution list)?

8. How do you handle email addresses of people that you communicate with (several alternatives are possible)?

_ Use old messages.
_ Use an electronic address book.
_ Use a paper address book.
_ Remember the addresses.
_ Use another program that can search for addresses.
_ Other:

Comment:

9. How do you save the email you receive (several alternatives possible)?

_ All in the same folder.
_ A folder for each month.
_ Sort in folders after subject
_ Sort in folders after sender.

_ Delete everything.

_ Delete all messages after they have been take care of.

_ other:

Comment:

10. How many messages do you have in your incoming mail box (mbox, inbox or similar)?

I have _____ messages in my inbox.

Comment:

11. How many email messages have you saved in different folders or files?

I have ______ messages saved.

Comment:

12. How can you handle your electronic mail today?

1) To handle my email has never been a problem.

2) I have some problems with my email, but I have not solved them yet.

3) I have had problems with my email, but I have found strategies to handle it.

4) I am close to not being able to handle my email.

5) I am close to not being able to handle my email, despite having tried different strategies.

6) I cannot handle my email.

Comment:

If you answered question 12 with alternative 1 “To handle my email has never been a problem”, please continue with question 15.

13. Comment with your own words the problems and difficulties that you have had with your electronic mail based on your answer to question 12.

Try to describe your situation based on

a, private messages,

b, messages in mailing lists (directed to several persons under the same address)

c, postings in conference systems (e.g. Usenet News).

a, private messages,

b, messages in mailing lists (directed to several persons under the same address)

c, postings in conference systems (e.g. Usenet News)

14. In which of the following situations have problems occurred?

_ When I shall receive information on what to do in my work.
_ When I communicate with others.
_ When I should order services from others
_ When I want to find relevant information for my work
_ When I should exchange documents (text files) with others.
_ When I should save messages
_ When I try to find saved messages
_ When I try to delete messages
_ Other:

15. If there was a possibility to automatically sort arriving messages (in different folders, e.g. one for a mailing list, one for messages from a certain person), would you use that?:

_ Yes.
_ No.
_ Uncertain.

Comment:

16. In which order do you read your mail (several alternatives possible)?

_ Presentation order
_ Private messages first
_ Distribution lists first
_ Select based upon the sender
_ Select based upon the subject
_ Select after size of the message

Comment:

17. How do you find old messages that you want to read again (several alternatives possible)?

If you use several alternatives, please comment when the alternative is of immediate interest for you.

_ I search in presentation order.
_ I make a selection according to sender.
_ I make a selection according to subject.
_ I make a selection according to date.
_ Other:

Comment:

18. Do you think that the possibilities to perform selections are adequate in your mail tool?

_ Yes.
_ No.

_ Uncertain.

Comment:

19. Do you think that a direct manipulative interface would be an advantage when you handle email (to drag messages to folders, attach documents to email messages)?

_ Yes.

_ No.

_ Uncertain.

Comment:

20. In what way would multimedia messages (audio, video and pictures) add something positive to your communication?

21. What are the three major problems that you have with email?

22. If you were forced to change mail system, which three features should definitely not be changed?

Are you interested in, and do you have the time, to be interviewed during one hour regarding electronic mail (at your work site)

_ Yes, I can be interviewed.

Phone:

Email address:

_ No, I cannot be interviewed.

Are you interested in testing CoMail during a month?

_ Yes.

_ No.

Comment:

Are you interested in discussing problems with electronic mail in a group?

_ Yes.

_ No.

Comment:

Name:

Thank you for answering this questionnaire!

A.3 Work situation questionnaire

(Translated from Swedish)

Here is six complementary questions to the questionnaire that were sent to IPlab last autumn. Please answer as soon as possible, you know how frustrating it is to have me reminding you until you have answered. Answer with integers. If you wish to write a comment, please do so.

1. Do you feel that you have time to do what you should in your work?

Use a scale from 1 to 5, where 1 is no, you do not have time at all and 5 is yes, you have time to do everything.

>>

Comment (If you have no time, when do you have no time):

2. Are you often stressed and in the last minute?

Use a scale from 1 to 5, where 1 is always stressed and 5 is never stressed.

>>

Comment (When are you stressed, when are you not):

3. What order do you have on your papers? (ordinary paper mail, reports e.t.c)

Use a scale from 1 to 5, where 1 very bad order (many unidentified piles) and 5 very good order (all piles are possible to identify, everything is sorted in files).

>>

Comment (What is in good order, what is not):

4. What order do you have on your files on the computer? (documents, programs)

Use a scale from 1 to 5, where 1 very bad order and 5 very good order.

>>

Comment (What is in good order, what is not):

5. What order do you have in your diary (paper or electronic)?

Use a scale from 1 to 5, where 1 very bad order and 5 very good order (you make notes of meetings and what should be done, and use them).

>>

Comment (What is in good order, what is not):

6. Who decide what you should do in your work?

Use a scale from 1 to 5, where 1 is someone else decides everything and 5 is you decide everything.

>>

Comment (What can you decide, what can you not decide):

Thank you for answering this questionnaire!

A.4 Final questionnaire

The interviews were made to 2 people that answered the industrial questionnaire, 5 people that answered the university questionnaire and the work situation questionnaire and 5 people that was not a part of those questionnaires. These people got a separate version of the questionnaire to shorten the time of the interview.

(Translated from Swedish)

28 QUESTIONS ABOUT YOUR ELECTRONIC MAIL

This questionnaire is a part of the development of CoMail, an experimental mail program developing at the Interaction and Presentation Laboratory (IPlab) at KTH since 1993. There is also a research project “Communication patterns in computer supported cooperative work”, where we study how electronic mail (email) is used and what kinds of problems there is with email.

As a preparation for the coming interview we would like you to answer this questionnaire. When we say “message”, “mail box” and similar, we always refer to electronic mail. The interview will take approximately 45 minutes.

This questionnaire has 28 questions. Each question has different alternatives and sometimes a line where you can write comments to your answer. We estimate the time to answer the questions to 15 minutes.

All answers will be treated confidently. When results are published, data will be presented by group and not by individual.

The best way to answer is to use the “reply” command, the answer will then automatically be sent to balter@nada.kth.se. If you wish to return the answer in paper format you can put it in Olle Bälter’s mail box at NADA, or send it to Olle Bälter, NADA KTH, 100 44 STOCKHOLM.

If you have any questions, please contact Olle Bälter on 790 91 57 or 82 77 86 or by email to balter@nada.kth.se

----

1. What is your profession?

2. Describe your main work task.

3. When I use email I mainly use the following computer:

    _ PC (DOS)
    _ PC(MS Windows)
    _ Macintosh
    _ Sun work station
    _ Other:

Comment:

4. Which electronic mail system do you use?

_ mail
_ Mailstrom
_ Eudora
_ Pine
_ Mail Tool V3
_ mh
_ Pegasus
_ Other:

Comment:

5. How often do you enter the email system?

_ Continuously. I read the mail when it arrives.
_ Several times a day.
_ Once a day.
_ Several times a week.
_ Once a week
_ More seldom.

Comment:

6. How many email messages have you received and sent respectively in the last 7 days?

I have received ___ messages.

I have sent ___ messages.

Comment:

7. How large amount (in percent) of your email is private (that is addressed to you and not to some distribution list)?

8. How do you handle email addresses of people that you communicate with (several alternatives are possible)?

_ Use old messages.
_ Use an electronic address book.
_ Use a paper address book.
_ Remember the addresses.
_ Use another program that can search for addresses.
_ Other:

Comment:

9. How do you save the email you receive (several alternatives possible)?

_ All in the same folder.
_ A folder for each month.
_ Sort in folders after subject
_ Sort in folders after sender.

_ Delete everything.

_ Delete all messages after they have been take care of.

_ other:

Comment:

10. How many messages do you have in your incoming mail box (mbox, inbox or similar)?

I have _____ messages in my inbox.

Comment:

11. How many email messages have you saved in different folders or files?

I have _____ messages saved.

Comment:

12. How can you handle your electronic mail today? (The answers represent more problems with higher numbers)

1) To handle my email has never been a problem.

2) I have some problems with my email, but I have not solved them yet.

3) I have had problems with my email, but I have found strategies to handle it.

4) I am close to not being able to handle my email.

5) I am close to not being able to handle my email, despite having tried different strategies.

6) I cannot handle my email.

13. What have caused/causes problems for you when you use email?

14. If there was a possibility to automatically sort arriving messages (in different folders, e.g. one for a mailing list, one for messages from a certain person), would you use that?:

_ Yes.

_ No.

_ Uncertain.

Comment:

15. In which order do you read your mail (several alternatives possible)?

_ Presentation order

_ Private messages first

_ Distribution lists first

_ Select based upon the sender

_ Select based upon the subject

_ Select after size of the message

Comment:

16. How do you find old messages that you want to read again (several alternatives

possible)?

If you use several alternatives, please comment when the alternative is of immediate interest for you.

_ I search in presentation order.

_ I make a selection according to sender.

_ I make a selection according to subject.

_ I make a selection according to date.

_ Other:

Comment:

17. Do you think that the possibilities to perform selections (all messages from a certain person, after a certain date) are adequate in your mail tool?

_ Yes.

_ No.

_ Uncertain.

Comment:

18. Do you think that a direct manipulative interface would be an advantage when you handle email (to drag messages to folders, attach documents to email messages)?

_ Yes.

_ No.

_ Uncertain.

Comment:

19. In what way would multimedia messages (audio, video and pictures) change something in your communication?

20. What are the three major problems that you have with email?

21. If you were forced to change mail system, which three features should definitely not be changed?

22. Do you use email via a modem?

_ Yes.

_ No.

Finally some questions about your work situation.

23. Do you feel that you have time to do what you should in your work?

Use a scale from 1 to 5, where 1 is no, you do not have time at all and 5 is yes, you have time to do everything.

>>

Comment (If you have no time, when do you have no time):

24. Are you often stressed and in the last minute?

Use a scale from 1 to 5, where 1 is always stressed and 5 is never stressed.

>>

Comment (When are you stressed, when are you not):

25. What order do you have on your papers? (ordinary paper mail, reports e.t.c)

Use a scale from 1 to 5, where 1 very bad order (many unidentified piles) and 5 very good order (all piles are possible to identify, everything is sorted in files).

>>

Comment (What is in good order, what is not):

26. What order do you have on your files on the computer? (documents, programs)

Use a scale from 1 to 5, where 1 very bad order and 5 very good order.

>>

Comment (What is in good order, what is not):

27. What order do you have in your diary (paper or electronic)?

Use a scale from 1 to 5, where 1 very bad order and 5 very good order (you make notes of meetings and what should be done, and use them).

>>

Comment (What is in good order, what is not):

28. Who decide what you should do in your work?

Use a scale from 1 to 5, where 1 is someone else decides everything and 5 is you decide everything.

>>

Comment (What can you decide, what can you not decide):

Name:

Thank you for answering this questionnaire!