EZMAIL:
USING INFORMATION VISUALIZATION TECHNIQUES TO HELP MANAGE EMAIL

by
Maryam Samiei
B.Sc., University of Tehran, 1992

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF APPLIED SCIENCE

in the School of Engineering Science

© Maryam Samiei 2004
SIMON FRASER UNIVERSITY
April 2004

All rights reserved. This work may not be reproduced in whole or in part, by photocopy or other means, without permission of the author.
APPROVAL

Name: Maryam Samiei
Degree: Master of Applied Science
Title of Thesis: EzMail: Using Information Visualization Techniques to Help Manage Email

Examining Committee:

Chair: Dr. Andrew Rawicz
Professor

Dr. John Dill
Senior Supervisor
Professor

Dr. Arthur Kirkpatrick
Supervisor
Assistant Professor

Dr. Stella Atkins
Examiner
Professor

Date Approved: April 6, 2004
Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author’s written permission.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Bennett Library
Simon Fraser University
Burnaby, BC, Canada
ABSTRACT

Electronic mail has become one of the major communication tools. Email is being used for a wide range of functions such as information management, task management, and document exchange, though it was originally designed simply as an asynchronous communication and messaging application. However, email clients have not progressed correspondingly to support an increasingly growing number of users in dealing with the complexity of email management. Therefore, several studies have focused on exploring the requirements of email users and proposed various solutions to improve the usability of email programs.

This thesis is an investigation of email usages, problems, and solutions. Based on this research, and making use of information visualization methods to help users in structuring and visualizing incoming and outgoing information flows, we have designed EzMail as an email visualization tool that runs in conjunction with an email client and creates a multi-view interface for email. Our primary focus is to group and visualize messages as components of their conversational threads to provide contextual information and conversational history. Our goal is to help email users perform better by getting more information with less effort using EzMail. We have developed visualizations for individual messages, properties of a message, and messages in a thread. We have incorporated the ability to annotate a message and the capability of replying simultaneously to several messages. To validate EzMail’s design ideas, we investigated how email users manage their messages, email folders, and conversational threads in a user survey. Following minor improvements resulting from the survey, a user study was designed and carried out to evaluate EzMail compared to Microsoft Outlook. In general, users performed significantly better using EzMail than using Outlook.
DEDICATION

to my parents

تقديم به بدر و مادرم
First and foremost, I am deeply grateful to Dr. John Dill who is a great supervisor. His invaluable instructions, support, and encouragement guided me through every step of my studies.

Then, I wish to express my gratitude to Dr. Arthur Kirkpatrick for all his valuable comments and advice. My sincere thanks to Dr. Stella Atkins for her guidance and suggestions as the external examiner at my thesis defense.

I would like to extend my thanks to all my friends, especially Mahta Boozari, for accompanying me throughout the period of my studies at SFU. I would also like to acknowledge all the members of GrUVi lab for being both good friends and colleagues.

My very special thanks to my dearest family for their endless love, support, and inspiration that made all this possible.

Finally, thanks to MohammadReza for always being there for me.
# Table of Contents

Approval .......................................................................................................................... ii
Abstract .................................................................................................................................. iii
Dedication ............................................................................................................................... iv
Acknowledgements ................................................................................................................. v
Table of Contents ................................................................................................................ vi
List of Tables ........................................................................................................................ vii
List of Figures ......................................................................................................................... ix

CHAPTER 1 Introduction ............................................................................................................. 1
  1.1 Our Approach .................................................................................................................. 2
  1.2 Outline of the Thesis ......................................................................................................... 4

CHAPTER 2 Background ............................................................................................................. 5
  2.1 Email System ...................................................................................................................... 5
  2.2 Email Clients ...................................................................................................................... 6
  2.3 Information Visualization ................................................................................................. 10
     2.3.1 Reference Model ........................................................................................................ 10
     2.3.2 Document Visualization ........................................................................................... 12
     2.3.3 Graph Visualization .................................................................................................. 17

CHAPTER 3 State of the Art ....................................................................................................... 19
  3.1 Email Research ................................................................................................................. 19
     3.1.1 Patterns of Email Use ............................................................................................... 19
     3.1.2 Email Problems ........................................................................................................ 21
     3.1.3 Proposed Solutions .................................................................................................... 23
  3.2 Approaches to Email Problems ......................................................................................... 25
     3.2.1 Visualization ............................................................................................................. 25
     3.2.2 Task Management .................................................................................................... 32
     3.2.3 Message Classification ............................................................................................... 34

CHAPTER 4 EzMail .................................................................................................................... 36
  4.1 Problem Statement ............................................................................................................. 36
  4.2 Analysis ............................................................................................................................... 37
     4.2.1 Threading....................................................................................................................... 37
     4.2.2 Information Visualization Methods ............................................................................... 39
     4.2.3 EzMail Development .................................................................................................. 39
  4.3 EzMail Design .................................................................................................................... 41
     4.3.1 Overview of EzMail .................................................................................................. 41
     4.3.2 EzMail Interface ........................................................................................................ 43
     4.3.3 EzMail Functions ....................................................................................................... 50
  4.4 Discussion .......................................................................................................................... 52
4.4.1 Information Visualization Methods and Systems ........................................52
4.4.2 Email Visualization Methods .................................................................54

CHAPTER 5 Email User Survey .........................................................................56
5.1 Design and Methodology ...........................................................................56
5.2 Results .........................................................................................................57
  5.2.1 General ...................................................................................................58
  5.2.2 Foldering ................................................................................................60
  5.2.3 Threading ...............................................................................................65
  5.2.4 Attachments ...........................................................................................67
5.3 Discussion .....................................................................................................69
5.4 Summary .......................................................................................................69

CHAPTER 6 EzMail User Study .........................................................................71
6.1 Design and Methodology...............................................................................71
  6.1.1 Scenario ..................................................................................................72
  6.1.2 Tasks ......................................................................................................74
6.2 Results ..........................................................................................................75
  6.2.1 Pre-Session Questionnaire ......................................................................75
  6.2.2 Statistical Analysis ..................................................................................76
  6.2.3 Post-Session Questionnaire .....................................................................81
6.3 Discussion ......................................................................................................84
6.4 Summary .......................................................................................................85

CHAPTER 7 Conclusion .....................................................................................87
7.1 Summary .......................................................................................................87
7.2 Research Contribution ..................................................................................88
7.3 Future Work ..................................................................................................89

Appendix A EzMail User’s Guide .....................................................................90
Appendix B Email User Survey ..........................................................................95
  Questionnaire ..................................................................................................95
  User Survey Results .........................................................................................99
Appendix C EzMail User Study .........................................................................101
  Pre-Session Questionnaire ............................................................................101
  User Study Tasks .............................................................................................103
  EzMail Post-Session Questionnaire ...............................................................106
  Outlook Post-Session Questionnaire ..............................................................107
List of References .............................................................................................109
LIST OF TABLES

Table 5-1 Email Programs Popularity ........................................................................................................ 59
Table 6-1 Task Completion Time .................................................................................................................. 77
Table 6-2 Task-Related Questions ................................................................................................................. 78
Table 6-3 Task Performance – Raw Data ....................................................................................................... 79
Table 6-4 Task Performance – Statistical Analysis ....................................................................................... 79
Table 6-5 Features for Tasks .......................................................................................................................... 81
Table 6-6 Ease of Task Performance ............................................................................................................ 82
LIST OF FIGURES

Figure 2-1 Email Client/Server Model........................................................................................................... 6
Figure 2-2 Microsoft Outlook............................................................................................................................. 7
Figure 2-3 Outlook Express............................................................................................................................... 8
Figure 2-4 Netscape Messenger......................................................................................................................... 8
Figure 2-5 Pine................................................................................................................................................... 9
Figure 2-6 Eudora............................................................................................................................................... 9
Figure 2-7 Webmail.......................................................................................................................................... 10
Figure 2-8 Information Visualization Reference Model...................................................................................... 11
Figure 2-9 Flip Zoom....................................................................................................................................... 13
Figure 2-10 Pad++............................................................................................................................................ 14
Figure 2-11 CZWeb......................................................................................................................................... 14
Figure 2-12 SeeSoft......................................................................................................................................... 15
Figure 2-13 ThemeRiver................................................................................................................................ 15
Figure 2-14 Galaxies....................................................................................................................................... 16
Figure 2-15 ThemeView................................................................................................................................... 16
Figure 2-16 Tree-Map..................................................................................................................................... 18
Figure 3-1 ReMail Project – Thread Structures............................................................................................... 26
Figure 3-2 ReMail Project – Message Overviews............................................................................................ 26
Figure 3-3 Threaded Email Client.................................................................................................................... 27
Figure 3-4 Mixed-Model Visualization............................................................................................................. 27
Figure 3-5 Thread Arcs.................................................................................................................................... 28
Figure 3-6 Email Radar.................................................................................................................................... 29
Figure 3-7 TimeStore....................................................................................................................................... 29
Figure 3-8 Netscan............................................................................................................................................ 30
Figure 3-9 Loom................................................................................................................................................. 31
Figure 3-10 CZTalk......................................................................................................................................... 31
Figure 3-11 Conversation Map......................................................................................................................... 32
Figure 3-12 TaskView..................................................................................................................................... 33
Figure 5-17 Saving Attachments to File Folders ................................................................. 68
Figure 5-18 Leaving Attachments with Messages .............................................................. 68
Figure 6-1 Task Completion Time .................................................................................. 77
Figure 6-2 Task Performance – Analysis Summary ......................................................... 80
Figure 6-3 Ease of Task Performance ............................................................................. 82
Figure 6-4 Usefulness of EzMail Features ..................................................................... 83
Figure 6-5 Ease of Use of EzMail Features ..................................................................... 83
Figure 6-6 Ratings of EzMail Visualizations ................................................................. 83
CHAPTER 1

INTRODUCTION

Electronic mail or email is an asynchronous electronic messaging system that is basically used for the transmission of electronic messages among users. However, it is increasingly being used for task management, information management, document exchange, and time management as well as communication management [13][57]. The increasing number of email users, usages, and messages has been referred to as email overload [57], which makes users feel frustrated by the volume of their email and terrified at the risk of losing their important items.

Part of the problem arises from users’ attempts to organize a large number of messages into a filing system [57]. In practice, users find creating an efficient filing system and choosing appropriate folders difficult and time consuming [13] [57]. As a result, searching and retrieving information from email messages is also challenging. Therefore, users often leave their messages in the inbox both to keep their tasks and to-dos available and as a reminder that further actions are required [57].

Another problem with email is the limitation of this medium as a file transfer tool, though it is used extensively to exchange documents. The lack of version control on attached documents makes tracking of document revisions difficult. [13]

There are also other problems related to task management in email such as: managing deadlines and reminders, grouping messages of a thread and related files, and getting a task-oriented view. [5]
Email clients generally seem not to have progressed correspondingly to handle these new email functions. Therefore, email clients need to be redesigned to help users in classification and structuring of email messages into workable units to get a better overview of the contents and context of their mailbox.

Several approaches have been proposed by email researchers to create a better email client. Threading provides users with the context, history, and status of a thread and reduces the inbox clutter [57]. The benefits of a threaded email client are: better local context results from displaying a message along with all related messages, a greater global context to view more conversations by collecting messages of a thread into a single entity, and valuable global operations for groups of messages rather than single messages that reduce the user’s effort for handling messages [54].

Among other suggestions are semantic clustering of related messages and documents to help users view contextual information and reduce clutter, and marking particular inbox items as action items, e.g. to-dos, reminders, and so on. [57]

Another approach is email visualization that has been applied in two categories [21]. The first category is the visualization of activities performed in email such as tasks and threads. The other category is focused on the visualization of email data to reveal patterns in email communication such as usage patterns and social connections.

1.1 Our Approach

Our goal is to apply information visualization techniques to assist with information management and retrieval in email clients. Information visualization is the use of computer-supported, interactive, visual representations of abstract data in order to gain insight into the information [7]. The inherent relations among textual elements of a mail system suggest the use of document and graph visualization techniques as two classes of information visualization. In particular, we have focused on thread visualization to provide contextual information and conversational history for email messages.

We have designed a graphical tool, EzMail, to run in conjunction with an email client and create a multi-view interface for the messages. The purpose of this visualization is to develop a graphical hierarchy integrated with the traditional file organization to allow the presentation of relationships among messages as well as a high level overview of the linear stream of email
messages. Our primary focus is to group and visualize messages as part of their conversational threads to display content of messages in detail while both preserving contextual overview and visually representing relationships among messages. Threading collects messages of a conversation scattered throughout the mailbox and presents them as one entity. Our attempt is to help email users retrieve more information from a cluttered mailbox.

The main interface of EzMail consists of five interconnected views to display the data at several levels of detail: a folder view to display the list of user's personal folders, a list view to list messages in a selected folder grouping by thread, a message view to display message content, a greeked view to present an overview of the structure of messages in the list view, and a node view to represent a more detailed presentation of messages and threads.

Two additional views have been provided to help in managing messages. The thread view implements a threading feature by collecting and displaying messages of a thread scattered throughout the mailbox, regardless of folder. The sender view collects and displays messages to or from a correspondent from all folders.

Two additional functions support task and communication management in EzMail. First is the capability of adding annotation to a message, which may serve as a reminder or to-do. The second function is replying simultaneously to several messages allowing a user to gather two or more threads together. This function is helpful in converging conversations and drawing conclusions.

We conducted a user survey to investigate how email users manage their messages, email folders, and conversational threads. The results of this survey and comments of the participants were a validation to our design ideas for EzMail. We discovered that threading features of current email clients are not yet very well-developed and hence some of the users are not still really familiar with threading and its benefits.

A user study was designed and performed to evaluate the improvements of EzMail over Microsoft Outlook. We developed a number of tasks based on a real scenario to test the visualizations and functions of EzMail and compare them with similar features in Outlook. The dependent variables were the time to complete an assigned task, the number of task errors and the number of errors in task-related questions. Based on the results, the performance and satisfaction of EzMail users were significantly better than those of Outlook users. The results suggest that EzMail is helpful in finding messages in a thread, annotating a message, and replying simultaneously to several messages.
1.2 Outline of the Thesis

This thesis is organized as follows. Chapter 2 provides background information on email systems and standards, and describes typical email clients. It also introduces useful definitions, concepts, and methods in information visualization. Chapter 3 investigates related research on email patterns of use, problems, and suggested solutions, and reviews different approaches intended to improve email clients. Chapter 4 states the problem we have focused on and describes the design and implementation of EzMail. Chapter 5 contains the details of our user survey on email management behavior of email users. Chapter 6 discusses the development, performance, and the results of the evaluative EzMail user study. Finally, chapter 7 presents the conclusions and possible future work of this research.
CHAPTER 2

BACKGROUND

In this chapter we introduce some basic concepts in two areas directly relevant to our work. First, following a brief introduction to the basics of email operation, we overview the operation and user interfaces of several standard email clients. We then introduce various aspects of information visualization especially those applicable to this work, i.e. document and graph visualization.

2.1 Email System

Electronic mail or email is an asynchronous electronic messaging system that uses standard conventions for the exchange of messages among users over communication networks. Figure 2-1 shows an email client/server model. A message is composed by a sender using an email client and transmitted through a server and via the network to the mail server of the recipient end where it can be opened and read by the receiver of the message using another email client. SMTP (Simple Mail Transfer Protocol), which is a store-and-forward method, is used for the transmission of messages through a series of servers up to the final destination. An MUA (Mail User Agent), which is the client software, provides users with an interface to manage their email. An MTA (Mail Transport Agent), which is also referred to as the mail server program, receives messages from various sources (MUAs and other MTAs) and routes them to their proper destinations. An email client uses POP (Post Office Protocol) or IMAP (Internet Message Access
Protocol) to read email from servers. POP is used to download a user's inbox from a server to the local machine and hence provide an offline operation, while IMAP implements an online operation by allowing the MUA to manage a user's email on the server. \[8\][28][56]

![Email Client/Server Model](image)

Figure 2-1 Email Client/Server Model

An email message has typically two major parts: header and body. The standard for text-based email is based on RFC822 [44], which defines the format for the syntax and headers of email messages. The header contains a set of fields that provide the essential information needed to process a message such as message routing information and email addresses of the sender and recipients. The basic structure of an email address consists of two parts. The first part is the username by which the recipient is known at her/his site. The second part of the address begins with the at sign (@) followed by the destination computer address. Since the characters available for RFC822 messages are limited to US-ASCII, the MIME (Multipurpose Internet Mail Extension) protocol was developed to define a method for transferring more complex types of data (such as attached documents, graphic images, etc.) via email. [33] [58]

2.2 Email Clients

An email client (i.e. an MUA) is an application that provides visual interactive access to mail and allows the user to send and receive email messages through ISPs (Internet Service Providers). Messages are automatically stored in electronic mailboxes that are personal areas of disk memory on the ISP's server (for clients using IMAP) or on the client computer (for clients using POP) for each user. Email clients usually have a directory named inbox in which new messages are placed. Furthermore, users may create personal folders to categorize their email. The recipient can perform various actions on a message including reply, reply all, forward, save, or delete the message. [41]

There are basically two types of email clients. A client-based email, such as Microsoft Outlook and Netscape Messenger, uses a program installed on the computer to download email
messages into the user’s local mailbox. A web-based email, such as webmail and hotmail, is an email service that can be used on the web. The advantage of a web-based email program is that messages are kept on the server and accessible from any computer using a web browser.

Traditional email clients usually provide users with a two-paneled or three-paneled display for viewing messages. A folder list displays the list of personal folders while a message list presents messages of a selected folder. Another pane is used to preview the contents of a message. A number of functions such as foldering, filters, search, sort, and address book are also usually available to help users organize and manage their messages and contacts.

In this section, we give brief descriptions of several popular commercial email products.

**Microsoft Outlook** (Figure 2-2) is part of Microsoft Office and serves as a Personal Information Management (PIM) tool. Although often used as an email application, Outlook has features for managing contacts, a calendar, and to-do lists. The calendar is used for keeping the date-specific information such as appointments, meetings, holidays, and so on. The list of tasks with related information and documents can be organized in tasks view. The journal can be used as a personal diary to organize documents and communications related to a specific individual or project. Notes is a place to keep track of electronic notes. The address book has a large number of fields to record information about contacts. The one nagging problem with Outlook is that some of the features are not easily visible and getting at them can be a real chore. The security deficiencies of Outlook make it a popular target for most worm and virus writers. [9] [18]
Outlook Express (Figure 2-3) comes for free with Internet Explorer. The primary use of Outlook Express is to access email and newsgroups. The interface is straightforward and easy to use. Though Outlook Express lacks the advanced features of Microsoft Outlook, it supports many of the same email features that are included in Outlook, such as HTML mail, stationery, filtering tools, and support for multiple email accounts. [9][18]

![Figure 2-3 Outlook Express](image)

Netscape Messenger (Figure 2-4) is a free email client that is part of the Netscape Communicator software package. It is an advanced, versatile, and fully featured email program with support for rich HTML. Netscape offers easy to set up filters for automatic message filing. It is a security-conscious email client that helps deal with the spam problem in an easy and effective manner. Its few weak spots are custom filters and templates. [53]

![Figure 2-4 Netscape Messenger](image)

(Used by permission of Netscape Communications Corporation)
Pine (Program for Internet News and Email), (Figure 2-5), was originally developed for inexperienced users at the University of Washington. But it has evolved to support many advanced features and personal-preference options. It is a command line email client available mainly for Unix environments but also is supported in the Windows operating system. Pine is easy and quick to operate and has a good message editor. It lacks scripting and macros and does not support encrypted messages. [43][18]

![Figure 2-5 Pine](image)

(Trademark of the University of Washington, Used by permission)

Eudora (Figure 2-6) is a classic and powerful email client that is widely used by Microsoft Windows and Macintosh users. It handles multiple accounts and has versatile filters and an efficient search engine. It supports richly formatted HTML email as well as plain text features. The weak point is that it does not have built-in secure messaging support. [53]

![Figure 2-6 Eudora](image)

(Used by permission of Qualcomm Inc.)
Webmail (Figure 2-7) is a web-based user interface that allows users to manage their email using a web browser. The advantage is that the mailbox is accessible easily from any computer with a web browser. Another advantage is that the messages do not have to be downloaded. The disadvantages are the limited email storage space and the requirement to stay online to work with email. Most webmail services provide folders, filtering, address books, and spam detectors. Hotmail was the first webmail service. Today many web-based email services are available. [58]

Figure 2-7 Webmail

2.3 Information Visualization

Information visualization seems a promising approach for the improvement of an email client. The inherent relations among textual elements of a mail system suggest the use of techniques that are used to visualize text and graph. Therefore, document visualization and graph visualization methods are reviewed in this section.

Visualization can be described as the mapping of data to visual form that supports human interaction in a workspace. Information visualization is “the use of computer-supported, interactive, visual representations of abstract, nonphysically based data” to help users gain insight into the information space. [7]

2.3.1 Reference Model

The general reference model of Figure 2-8 represents the basic steps of visualizing information. [7]
2.3.1.1 Data

As shown in Figure 2-8, Data Transformations map Raw Data into Data Tables or any logical equivalent of Data Tables. It is important to transform the abstract data without a spatial component into a set of relations that has a structural format and thus is easier to visualize. To give an example, text might be considered as indexed strings that can be transformed into document vectors with the dimensionality of number of the words in x, y, z coordinates. [7]

2.3.1.2 Visual Structures

Visual Mappings transform Data Tables into Visual Structures using graphical elements. A good mapping must be expressive and effective. It is expressive if all and only the data in Data Tables are represented in Visual Structures. An effective mapping is fast to interpret and leads to fewer errors. Visual Structures are basically made from the following components: [7]

- Spatial Substrate: Efficient use of space is fundamental. 1D, 2D, and 3D visualizations position graphical objects on orthogonal axes. There are several techniques to increase the amount of data encoded by space: composition (orthogonal placement of axes, creating a 2D metric space), alignment (repetition of an axis at a different position in space), folding (continuation of an axis in an orthogonal dimension), recursion (repeated subdivision of space), and overloading (reuse of the same space for the same data table).
- Marks: are the visible graphical objects in space consisting of points, lines, areas, and volumes.
- Graphical Properties: Topological properties show the relations among objects such as connection, proximity, and enclosure. Retinal properties include spatial properties (such as position, size, and orientation) and object properties (such as gray scale, color, texture, and shape). Temporal encoding is also useful to show the changes in marks’ graphical properties.

2.3.1.3 View Transformations

Referring to Figure 2-8, the final stage is transforming the static presentations into Views in space-time. There are three common View Transformations: [7]
- Location Probes: use location to reveal more information from the data tables. Pop-up windows and brushing are examples of location probes that provide details-on-demand.

- Viewpoint Controls: use affine transformations to view more details by manipulating the viewpoint. The pan and zoom space is a powerful visualization for any kind of hierarchical data since nested details can be shown at a smaller scale, conveying the relations between objects very naturally. This method takes advantage of panning to move a viewing window around and zooming to choose between local detail and global overview [50]. Another example is overview + detail which is the use of two or more windows to display an overview of the visual structures along with detail views.

- Distortion: modifies the presentation to create focus + context views that is the concurrent presentation of local detail in focus and global context at reduced magnification. The distortion of contextual data is obtained by applying a transformation function to reduce the number of pixels devoted to the context space. The area of interest is moved into the central focus region to be viewed in more detail.

2.3.1.4 Interaction

"Human Interactions" may control the parameters of each stage of transformation from Raw Data to Views (Figure 2-8). These controls can be separate from the visualization or integrated into it. Some interaction techniques are dynamic queries, direct manipulation, overview + detail, magnifying lens, and zooming. [7]

2.3.2 Document Visualization

Document visualization is not aimed at retrieving information, but letting the user discover roughly what is available in a document collection [50]. The motivation of document visualization is to gain insight into the information on text documents without reading them, but by visualizing them individually or as part of a document collection in one, two or three dimensions. The increasing number of electronic documents organized into digital libraries or scattered around the web suggests that these aspects of information visualization are becoming increasingly important. The challenge is to present large amounts of data simultaneously on a limited screen. Data might be plain text documents, HTML files, or program codes. [50]

We may generally categorize document visualization methods into two main groups. One group including Zoom Browser [24], Pad++ [4], and CZWeb [16] are more concerned with grouping the documents, showing their relationships, presenting their content, and preserving the
context. The other group including ThemeRiver [22], Galaxies [59], and ThemeView [59] visualize the thematic variations of the documents.

**Zoom Browser** [24] [25] is a web browsing tool developed based on flip zooming technique. Flip zooming is a focus + context technique that lets the user ‘flip’ through a document like the pages in a book (Figure 2-9). A dataset is divided into equally sized pages which are presented as thumbnail sketches on the display. Clicking on a page brings it into focus and makes the other pages shrunk and rearranged keeping its left-to-right top-to-bottom order. Hierarchical flip zooming [6] is achieved by using flip zooming visualization recursively and is used for hierarchical data structures. With flip zooming, the text outside the focus is easy to read as no spatial distortion happens. Since the transformations are not computationally heavy, the technique is good for interactive applications. It is useful for searching or editing large documents and codes. It can also be an alternative to standard windowing interface. The problem with flip zooming is the unpredictable placement of pages when the rearrangement occurs.

![Figure 2-9 Flip Zoom](Holmquist, 1997, Used by permission of ACM)

**Pad++** [4] [34] surface is an infinitely zoomable space, with objects residing at specific \(x, y, zoom\) coordinates within that space (Figure 2-10). An important feature of the zoomable coordinate system is that objects have an absolute physical location. Therefore, in Pad++ all the data is placed directly on the surface and the user navigates through the data by panning and zooming. With Pad++, graphical objects can be created or loaded from external sources and manipulated in a 2D zooming interface. It supports images, text files, and hypertext. It can be used to view hypertext files while representing their parent-child relationships. Using an unfamiliar metaphor increases the learnability of Pad++.
CZWeb [16] [62] is a web visualization tool that communicates with the web browser and creates a dynamic fisheye view of the visited pages using the Continuous Zoom (CZ) algorithm. Continuous zoom [3] is a fish-eye based method that supports multiple focus points and smooth transition between views. By maintaining the relative location of nodes, the algorithm reduces the user's sense of disorientation. CZWeb facilitates web navigation and management of the retrieved information by generating a flexible map of the web sites as they are visited (Figure 2-11). As the user navigates through the web space, CZWeb automatically changes the size and position of web page icons and icon clusters to make room for new pages. The user can return to any page by clicking on its icon, and can change the configuration of icons and clusters. Interface cluttering and links crossing each other and nodes are problematic in CZWeb.
SeeSoft [15] is a software visualization tool that deals with statistics related to the source code. The files are displayed as columns and lines of code as thin rows (Figure 2-12). Color-coding is used to show statistics associated with each line. The user can directly interact with the display and open the code reading windows that display the actual code. Currently, SeeSoft can show up to 50,000 lines of code simultaneously. SeeSoft takes advantage from folding and recursion.

![Figure 2-12 SeeSoft](Eick, 1992, Used by permission of the Author)

ThemeRiver [22] is designed to visualize the thematic variations in a collection of documents over time (Figure 2-13). Themes or topics are represented as colored ‘currents’ flowing within a ‘river’ that narrow or widen to indicate decreases or increases in the strength of a topic in associated documents over time. The familiar river metaphor is believed to help users find the patterns quickly and easily, particularly for macro trends.

![Figure 2-13 ThemeRiver](Havre et al., 2002, Used by permission of Pacific Northwest National Laboratory)
**Galaxies** visualization [59] displays document relationships in a 2D scatterplot (Figure 2-14). The users may identify a reference document as a representative of the sort of content they are looking for. Other documents are compared to this reference for similarity. Projection of this high dimensional representation onto two dimensions shows a Galaxy of clusters, which the user can explore to gain insight into the information space.

![Galaxies visualization](image)

**Figure 2-14 Galaxies**
(Wise et al., 1995, Used by permission of Pacific Northwest National Laboratory)

**ThemeView** [59] (formerly known as ThemeScapes [40]) is generated from the projection of a multidimensional cluster into a three-dimensional landscape (Figure 2-15). The documents are represented by small points and those with similar content are placed close together and form an elevation. Moving a focus circle around reveals more detail of an area of interest. Its landscape metaphor calls upon human abilities for pattern recognition and spatial reasoning.

![ThemeView visualization](image)

**Figure 2-15 ThemeView**
(Wise et al., 1995, Used by permission of Pacific Northwest National Laboratory)
2.3.3 Graph Visualization

A second important class of problems addressed by information visualization methods is that of graph visualization. Graph visualization is the fundamental visual presentation of structured data whose underlying feature is connectedness. Typically, the elements of an information space are treated as nodes of the graph and the links between them as its edges. Graphs can be used to visualize two classes of networks: trees which have an underlying graph with no loop, and general networks in which any node can be connected to any other node. [50]

Tree visual structures use connection or enclosure to visualize hierarchical data. Connection creates a node-link diagram in which nodes and links represent cases and their relationship respectively. X and Y axes are used to show the tree depth and nodes separation. Unlike connection, enclosure fills the space. [50]

Network visual structures consist of nodes, links, and information associated with them. They can be used to display communication networks such as telephone networks, email social networks, and Internet, which have an underlying feature of connectedness. Node size may indicate the aggregated volume of incoming and outgoing traffic and links represent volume of transmission, capacity, and overload. [50]

A layout algorithm specifies the position of nodes and the shape of the edges that connect them. Trees have received the most attention. Their conventional presentation is easy to layout and interpret since it uses one spatial axis to separate levels and does not have any loop. However, trees occupy a great deal of space. The size of the graph is a key issue in graph visualization. Large graphs cause presentation, viewability, and usability problems. Clustering is an approach to reduce the number of visible objects by grouping data based on some selected attributes. Another concern is predictability which means that different runs of the algorithm for similar graphs should result in similar visual presentations. Time complexity or system response time is another important issue. Many layout algorithms including Treemap and Hyperbolic Browser have been proposed. [23]

Tree-Map [27] is a 2D space filling approach that partitions the display space into a collection of rectangular bounding boxes representing the tree structure (Figure 2-16). Each node is mapped to a rectangle whose area is proportional to some attribute such as node size. The algorithm first assigns a rectangular area to the root. The number of outgoing edges from the root node determines the number of partitions of the region. The algorithm then recurs down the left tree using the 90-degree rotated rectangle and splits on the y-axis direction. Therefore nodes are
partitioned vertically at even and horizontally at odd levels. For visual clarity, different colors must be used within each region. Interactive control allows users to specify the presentation of both structural and content information. Tree-Maps provide an overall view of the entire hierarchy, making the navigation of large hierarchies much easier.

![Tree-Map](image)

**Figure 2-16 Tree-Map**
(Shneiderman, 1991, Used by permission of the Author)

**Hyperbolic Browser** [31] [26] is a focus + context technique for visualizing and manipulating large hierarchies. The approach is to lay out the hierarchy on a hyperbolic plane and map this plane onto a circular display region. It initially displays a tree with its root at the center, but the display can be smoothly transformed to bring other nodes into focus.
CHAPTER 3

STATE OF THE ART

Electronic mail has become one of the major communication tools for an increasing number of users. Actually, in some cases, it has transformed into more of a *habitat* than an application [13]. Although email was originally designed as an asynchronous communication and messaging application, it is now used for a wide range of functions such as task management, information management, personal archiving, sending reminders, and document exchange, to name a few variations. This has been referred to as *email overload* [57], which makes users feel frustrated by the volume of their email and terrified at the risk of losing their important items [46]. That email clients have not progressed correspondingly to handle email overload has prompted several studies [13] [54] [57], which have focused on email clients and their ability to manage information. The studies explored the deficiencies of current email clients in fulfilling the needs and expectations of the users for management of their messages, tasks, and documents and proposed various approaches to improve the usage of email programs. These studies will be discussed in this chapter.

3.1 Email Research

3.1.1 Patterns of Email Use

Denning (1982) [10] was probably the first to bring the problems of email into attention. He believed that though electronic mail provided faster methods for document dissemination, it
lacked features to prevent unwanted reception. He proposed that “there must be special paths by which urgent, certified, and personal messages can arrive and all other paths must be filtered”.

Mackay (1988) [34] was one of the first researchers who observed the diversity in patterns of email use. She identified the contribution of mail to the management of information, time, and task. Conducting a series of interviews, she found that users generally fall into three groups:

- Those who view mail as an information management tool are archivers who are interested in classifying messages after reading.
- The prioritizers use mail as a time management tool and want to identify important messages before reading.
- The third group use email to manage their tasks, though email has no inherent task management property. They are requesters who assign tasks to performers.

A few years later, Whittaker and Sidner (1996) [57] presented a quantitative analysis of the mailboxes of 20 users. They discussed three main email functions: task management, personal archiving, and asynchronous communication. They found that the simple one-touch model, which assumes the immediate action of user on incoming messages to reply, delete, or file the message, does not work. This is because in practice, many messages cannot be acted on upon arrival and will be kept in the inbox to act as reminders to the user to handle them later. Three strategies were identified among users for handling email overload:

- No-filers who make no use of folders and keep their email in the inbox.
- Frequent filers who try in vain to minimize the size of their inbox by frequent use of folders.
- Spring cleaners who clean up their inbox periodically while extensively use folders.

Venolia et al. (2001) [54] developed a conceptual model of the user’s activities with email. Based on a field study and previous research, they outlined five email activities:

- Flow: keeping up with the flow of incoming messages.
- Triage: catching up with the accumulated email using serial or prioritized strategy.
- Task management: using email messages as a to-do list.
- Archive: keeping messages for future use based on their contents and attachments.
- Retrieve: retrieving older messages from archive.

Ducheneaut and Bellotti (2001) [13] believe that electronic mail serves as a personal information management (PIM) tool following an observation that users tend to embed their daily work into email as their favorite workspace.
Fisher and Moody (2001) [17] performed an automated collection of email data to study the structural and behavioral information of the mailboxes of 76 users. They found that the organizational behavior of users has not changed much since the study of Whittaker and Sidner in 1996 [57]. Based on the collected data, they observed that threads are quite prevalent in users’ mailboxes (one third of messages were threaded).

Kerr (2003) [29] in his survey on 42,000 messages found that 62% of messages belong to threads, which is much higher than that of [17] because he took into account the missing root messages (either deleted or not saved when sent).

3.1.2 Email Problems

We classify and summarize email problems in this section.

3.1.2.1 Email Overload

It seems that email overload is the main problem that exacerbates other email challenges. There are three factors that cause users to feel overwhelmed:

- Email is being used for a variety of functions it was not designed for. Information management activities such as: personal archiving, document exchange, storing personal names and addresses, and task management activities like: work task delegation, task tracking, sending reminders, to-dos, asking for assistance, scheduling appointments, and handling technical support queries are among these functions. [13] [57]
- The volume of sent and received messages is increasing because of the increasing number of email users and various uses of email. This includes spam email.
- The work that comes with the incoming messages, including the actions on the messages within the email program and other related activities outside the email program, adds to the workload of the user.

3.1.2.2 Organization of Messages

Part of the email problem arises from users’ attempts to organize a large number of messages into a filing system [57]. Email clients usually provide users with a directory structure to let them group and file related messages. Users create folders based on sender, organization, project, or personal interests [13]. A successful organization of messages needs a high degree of effort by the user. In practice, creating an efficient filing system and choosing appropriate folders is likely to be difficult and time consuming for users [57]. As a result, searching and retrieving
information in email messages is also challenging. Therefore, email users tend to use their inbox as a place for incomplete tasks, unfiled information, and ongoing conversations, which results in a cluttered inbox [57]. Users leave messages in the inbox both to keep them available and as a reminder that further actions are required [57]. Among the difficulties that users have with filing, Whittaker and Sidner [57] and Ducheneaut and Bellotti [13] mention:

- Filing information makes it less visible.
- Remembering the definitions of large number of folders for future archiving or retrieval is difficult.
- Too many or too few messages in a folder make it inefficient.
- An alphabetical ordering of messages is not always the best ordering.

Automatic filing based on filters aims at helping users with the filing activity, though it is not very helpful. Since automatic filing is performed on receipt of messages and moves the messages out of sight, email users hesitate to apply rules and filters unless for simple and safe filing such as filing away spam or moving messages from a mailing list to a specific folder.

Among the major reasons for the information retrieval difficulty are the lack of explicit semantic clustering of or linkage between relevant pieces of information and the limits of conventional search techniques using keywords [52]. Ducheneaut and Bellotti [13] discovered that the sort feature is more popular for message retrieval than search. They say the reason is that searching for a criterion such as subject, sender, or date is more difficult, much slower, and less efficient.

### 3.1.2.3 Document Exchange

Users exchange documents by email extensively, though they complain about the limitations of this medium as a file transfer tool. Users often have a limited quota on their mail servers which is easily exceeded by the volume of the attachments [54]. Additionally, lack of version control on the attached documents makes tracking of document revisions difficult. [13]

### 3.1.2.4 Task Management

Venolia et al. [54] observed that the most popular method among users to keep track of their tasks or to-dos is to leave them in the inbox. In another user study, Bellotti et al. [5] identified extended-response tasks that need more time and effort to be responded. These tasks happen partly because of the users' volume of email and to-dos and partly because of the length of the intervals between messages of a conversation caused by the delay of the other participants.
of the conversation to reply. They mention other problems with task management in email such as: managing deadlines and reminders, grouping messages of a thread and related files, and getting a task-oriented view.

3.1.3 Proposed Solutions

Based on the research in sections 3.1.1 and 3.1.2, it seems clear that email needs to be redesigned to better support its various functions and users. It is believed that a single strategy cannot serve all the users due to the diversity of email usage [45]. Some of the researchers have suggested solutions to improve the email functionality and interface. Here we classify and analyze these suggestions and their justifications.

3.1.3.1 Communication Management

Most of the researchers agree that threading is a tool of significant benefit to the email users in managing their communications.

- Collecting messages of a thread, or threading, provides users with the context, history, and status of a thread. [57]
- Viewing by thread simplifies tracking of a thread and reduces the inbox clutter. [57]
- Venolia et al. [54] mention three benefits for a threaded email client: better local context results from displaying a message along with all related messages, a greater global context to view more conversations by collecting messages of a thread into a single entity, and valuable global operations for group of messages rather than single messages to reduce the user's effort for handling messages.
- Threading helps the user review and understand large conversations faster and easier. [38]
- Whittaker and Sidner [57] offer the usage of a common thread ID to identify a conversation.
- Kerr [29] outlines seven key qualities for an effective thread visualization: chronology (arrival sequence of messages), relationships (in-reply-to relations), stability (stable positions for messages), compactness (small size), attribute highlighting, scalability, and interpretation (a sense of the conversation type).

3.1.3.2 Information Management

- Information management should support both the static storage of messages and the dynamic flow of incoming messages. [52]
- To improve the information management ability of email, an improved system of classifying messages is necessary. [34]
• Semantic clustering of related messages and documents may help users view the contextual information and reduce the clutter as well. [57]
• A more flexible organization of folders and a cache of recently accessed items may better support the use of email as a PIM (Personal Information Management) tool. [13]
• Allowing messages to exist in several folders simultaneously may reduce the fear of losing important items for the users. [54]

3.1.3.3 Task Management

• Management of to-dos and reminders should be supported within an email client [13]. The user should be able to mark particular inbox items as action items, e.g. to-dos, reminders, and so on. They must be highly visible and re-appear as action items in time as the deadline approaches [57].
• Viewing by thread might better help users manage their tasks. [54]
• The transfer of tasks’ information from email client to other media where the tasks are going to be performed must be easy (e.g. to meeting scheduler). [20]
• Additional support such as better visualizations is needed for management of new tasks. [20]

3.1.3.4 Document Management

Since email is a central medium for exchanging documents, email clients should provide their users with some document management features.

• It is quite helpful if the email client provides some form of version control on the documents being exchanged via email. This way, the email user can track the document revisions more easily and make sure to access the latest version of the document. [13]
• To help users handle the burden of attachments, Venolia et al. [54] suggest that the user must be able to file them separately from their mail but keep the links between the message and attachments.
• The ability to make reference, e.g. by hyperlinks, to sub-components of documents, such as a particular paragraph in a document, is useful. [13] [60]

3.1.3.5 Time Management

• A prioritized list of pending messages helps users with the triage activity [54]. Identifying important messages and sequencing them based on the status of the user would support time management in email [34].
To support flow activity, new message notification in email clients should provide enough information for the user to decide on reading the new email immediately or not. It must be based on the importance of the message and provides the user with quick common commands like open, reply, or delete. [54]

3.2 Approaches to Email Problems

For the improvement of email clients, a number of visualizations and user interfaces have been proposed. Different approaches have been chosen to help address the problems of email. We classified them in three categories: visualization, management of tasks, and classification of messages.

3.2.1 Visualization

3.2.1.1 Email Visualization

Email visualization has been applied in two categories [21]. Some of the researchers have tried to visualize the activities performed in email such as tasks and threads. The other category is focused on the visualization of email data to reveal the patterns in email communication such as usage patterns and social connections. The subject of this thesis is more related to the representations of the first category.

A research group at IBM (Rohall et al.) has looked at different aspects of electronic mail and designed several email prototypes in a project on Reinventing Email (ReMail) [47] to develop new visualizations for a better email client. They have designed an email program that uses tree structures combined with timelines (Figure 3-1) to visualize the relationships among messages in threads and among people involved in a conversation [46]. It also displays reduced-resolution color-coded overviews of messages with a column for each day (Figure 3-2) to visualize the structure of documents and help users pick out different types of email. The annotation feature of the prototype serves as a reminder and as a means of communication in inboxes that are shared among users [47]. They have also incorporated a synchronous communication or chat facility to their prototypes [38]. An email summarizer has been developed to process messages of a thread and pass it to a commercial summarizer to provide summaries of email messages to the users [45]. In one of their user studies, they asked the subjects to use a test email client for about two weeks. In general, the subjects found the interface and especially the threading and message
overviews useful [47]. The research of this group is ongoing through a combination of studies and experiments to develop a new set of features that serve the business use of email [38].

![Figure 3-1 ReMail Project – Thread Structures](image1)

*Figure 3-1 ReMail Project – Thread Structures*  
(Rohall et al., 2001, Used by permission of IEEE)

![Figure 3-2 ReMail Project – Message Overviews](image2)

*Figure 3-2 ReMail Project – Message Overviews*  
(Rohall et al., 2001, Used by permission of IEEE)

A threaded email prototype that supports the *triage* activity was developed by Venolia et al. [54]. An application window (Figure 3-3, left pane) displays a thread list with a tree thumbnail for each thread, and a thread browser (Figure 3-3, right pane) lists the messages of a selected thread. The results from testing the prototype validate the benefits of a threaded email client.
In an improved prototype, Venolia and Neustaedter [55] present a mixed-model conversation visualization that supports two models of a conversation: a *sequential model* that creates a chronological list of messages clustered by conversation, and a *tree model* that specifies the sequence of replies in a thread (Figure 3-4). A part of the visualization shows the conversation messages along with their contents in a chronological order. A schematic overview of the conversation tree is also shown, which keeps the context in view. In a user study, the designers of the system verified the effectiveness of this visualization in understanding the two models of a conversation.
Thread Arcs [29] is an email thread visualization technique that encodes the chronology of a thread by placing message nodes in a linear layout and shows the context of the thread by connecting the nodes with relationship arcs (Figure 3-5). This layout gives a compact and stable visualization of a thread capable of highlighting a number of message attributes such as messages from a sender or unread messages. The technique has been incorporated in an email client prototype and tested on users' threads. It has been compared to Tree Diagrams (used in Netscan [49]) and Tree Tables (used by Rohall et al. [46]) and proved to be more stable, compact, and supportive of chronology visualization. The compact presentation looks quite appealing. However, it is not combined with any timeline.

![Thread Arcs](image)

Figure 3-5 Thread Arcs
(Kerr, 2003, Used by permission of IEEE)

Email Radar [37] is a highly visual interface to act as a peripheral application to email clients and help users manage their messages. Its purpose is to support message visual markings, a reminder system, and operations on conversations as well as maximizing contextual information and reducing barriers to message management. The main component of the interface is the folder slope that represents a user-defined folder and is divided into time segments with icons showing different kinds of messages (Figure 3-6). The contents of multiple folders may be viewed at the same time with Radar. There is also a sticky region at the top of each slope to display reminders or important messages out of folders. We think that relying solely on a visual interface without any text might make the interface difficult to use and remember.
TimeStore [61], a time-based interface for email and task management, uses the time of arrival to organize messages as dots on a two-dimensional graph by time and sender (Figure 3-7). Mailbox views are used to create dynamic mail folders instead of the traditional static ones, e.g. inbox is a view to list new and unread messages. TimeStore allows the user to create tasks from within the message window, which provides related context for the task.
3.2.1.2 Newsgroups Thread Visualization

There are also several visualizations and user interfaces developed for Usenet newsgroup conversations. Usenet is a global system for hosting, storing, and disseminating asynchronous threaded conversations. Newsgroups are repositories of Usenet messages for discussions on particular topics. The difference between Usenet thread visualization and email thread visualization is that in the former, the social structure of discussion is more important to be visualized [29].

**Netscan** is a general platform for Usenet messages used by Smith and Fiore to integrate multiple visualizations and illustrate social and temporal patterns embedded in thread discussions of Usenet newsgroups [49]. Their interface (Figure 3-8) consists of a thread tree (A) to visualize the structural and temporal history of a thread, a piano roll (B) to display a list of participants, the number of their posts, and the day they posted, an interpersonal connection (C) to show the reply connections among users, and a message display (D). The thread tree seems to occupy a lot of space and may not provide similar presentations in different runs.

![Figure 3-8 Netscan](image)

(Smith et al., 2001, Used by permission of ACM)

**Loom** [12] is a visualization tool for threaded conversations of Usenet groups. Each message is represented as a dot on a 2D interface with time as x-axis and sender as y-axis (Figure 3-9). This visualization shows the pattern of participation of each member in the discussion. Lines are used to connect messages of a conversation to each other and reveal the interaction style.
CZTalk [30] is a graphical tool designed to alleviate some of the usability problems in existing online discussion and conferencing software interfaces. It is based on CZWeb and in a similar manner, provides a new interface to display and organize online discussion messages in a network structure instead of the traditional file structure. In CZTalk, each page node represents a message or a posting, a cluster node is a group of messages of the same topic or subtopic, and directional links between nodes show the flow of conversation (Figure 3-10). With a network display of messages, CZTalk conveys structural, relational, and textual information of the discussion. Some other features of CZTalk include: allowing convergence of messages, color-coding of participants, automatic and user defined visual organization of messages, and providing detail-in-context displays. Cluttering and lack of sort feature are some of the limitations of CZTalk.
**Conversation Map** [48] uses text analysis algorithms to create a content-based Usenet newsgroup browser that helps users get an overview of the social and semantic archives of newsgroup discussions. The interface has four major parts (Figure 3-11). The social network reveals the main participants of the discussion by showing the pattern of messages and replies (sent and received messages) among people. The discussion themes are found by an analysis of lexical cohesion. The messages of a thread are shown as nodes of a spider web. The semantic network displays the central terms used in the discussion by parsing and analyzing contents of messages. One problem with this visualization is that it does not provide a temporal order of messages.

![Figure 3-11 Conversation Map](Sack, 2000, Used by permission of ACM)

### 3.2.2 Task Management

Some of the researchers have focused on improving task management feature of email as one of its main uses.

**TimeStore-TaskView** [19] [20] focuses on pending tasks by presenting alternative representations of messages and tasks. This interface has been designed based on TimeStore and presents tasks as small icons on a two-dimensional grid with time as x-axis and other task information such as subject and sender as y-axis (Figure 3-12). Only those messages with references to the future that cannot be acted upon immediately (containing pending tasks) are represented.
Taskmaster [5] is a task management environment (add-on to Outlook) that groups task-related messages including single messages and messages of a thread along with their attachments and links. The designers call these task items thrasks and present them in the main list view of their interface (Figure 3-13, top pane). The middle pane displays all the information (messages, links, and files) related to the thrask and the bottom view shows the contents of the selected item. Thrasks are flexible and can be renamed, moved, split, or combined. Users can also include other types of information such as web pages, presentations, and documents in their task list. In addition, meta-information like reminders, deadlines, and flags can be added to the thrasks. In a field study, thrasks were found to be reliable and successful in collecting and organizing messages.

Figure 3-12 TaskView
(Gwizdka, 2002, Used by permission of the Author)

Figure 3-13 Taskmaster
(Bellotti et al., 2003, Used by permission of ACM)
3.2.3 Message Classification

A group of researchers has tried to reduce the burden of email management for the user by automatic classification of messages.

Sudarsky and Hjelsvold [51] use information visualization tools to provide an automatic classification of messages based on the hierarchical nature of the domain names in email addresses (Figure 3-14). A temporal view is used to display messages on a two-dimensional grid with time as the x-axis and sender as the y-axis. A querying interface supports a full-text search. A potential difficulty with this approach is that classification based on domain names may not be very familiar for users. Additionally, since we often receive most of our messages from our workplace, the volume of messages in a single category increases dramatically and hence reduces the usefulness of this categorization.

Figure 3-14 Prototype by Sudarsky and Hjelsvold
(2002, Used by permission of IEEE)

Nelson Email Organizer (neo) [39] is an add-on product to Microsoft Outlook that classifies email by date, correspondent, bulk mail, attachment, and status and shows them in different views. There is no inbox in neo. Instead, messages automatically appear in more than one category at the same time, e.g. categories by date and correspondents. Active Mail is a folder for keeping new mail until it is acted upon (Figure 3-15). Its difference from inbox is that removing messages from Active Mail does not delete them from the mail system since they exist in other folders. A benefit of this system is that when the user picks a correspondent, all the sent and received messages for that person are displayed and hence give a sense of the ongoing conversation with that person.
Bifrost Inbox Organizer [2] categorizes messages with predefined rules into several categories of interest identified based on an observation of users. These categories include messages that are related to an item in calendar for today, messages from VIP senders (defined by the user), messages whose sole recipient is the user, and messages from listservs (Figure 3-16). A user study has shown that the tool is helpful for prioritizing incoming messages.

Figure 3-15 Nelson Email Organizer
(Used by permission of Caelo Software Inc.)

Figure 3-16 Bifrost Inbox Organizer
(Balter et al., 2000, Used by permission of ACM)
CHAPTER 4

EzMail

In this chapter, we first state the problem that we are aiming at and the solutions we believe could help. Then we will give a thorough analysis and discussion of EzMail design and development.

4.1 Problem Statement

The fact that email is overloaded is inevitable. Email clients, originally designed as communication tools, have not kept up with the growth of email uses and have left users with difficulties in managing their email. Among the basic solutions developed within email clients to help users cope with email overload are folders, filters, search, and sort. Far from perfection, these features cannot fulfill the users' requirements and even raise their own problems. Therefore, email clients need to be redesigned to support the various range of users and functions.

The basic idea of email is still communicating with people, though it is used for a variety of functions. Conversations are among the major components of an email client and form a major part of a user's mail. Therefore, conversation management is one of the main focuses of email research. Threading has been proposed as an effective solution to help users manage their email conversations and tasks. Presentation of a message as a component of its conversation thread provides the relevant context and history for the message. In chapter 3, we reviewed different approaches that have been taken to accomplish this feature in an email client. Some email clients
support a textual representation of threads while more visual presentations of threads are of interest to the researchers.

Our goal is to design a visualization tool to run in conjunction with a user's email client and create a multi-view interface for the messages. We want to display the contents of messages while both preserving the context overview and presenting the relationships among messages. Our main concern is to group and visualize messages as components of their conversation thread to provide contextual information and conversational history. Like many researchers, we believe threading can improve the email clients and help users handle email overload. We have considered information visualization as an effective approach to represent threads in an email client. It can help email users in structuring the incoming and outgoing information flow to workable units. We have made use of information visualization methods to visualize email messages, focusing on threads of a mailbox. We have designed visual cues to represent different attributes of a message such as attachments and annotations. We also developed new functions to let the user perform certain operations on messages and conversations. We have named our prototype EzMail to emphasize our goal that it be easy to use. Our purpose is to help email users perform better by getting more information with less effort using EzMail.

4.2 Analysis

4.2.1 Threading

The idea of conversation threads is borrowed from discussion databases. Lewis and Knowles [32] define threads as “conversations among two or more people carried out by the exchange of messages”. Jacob Palme [42] describes a thread as “a set of messages which are replies to each other”. Rohall et al. [46] define a conversation thread as “the series of replies to a message and the replies to those replies”. Bernard Kerr [29] defines the first message in a thread as the root, the message that receives replies as the parent and those replies as children of that parent.

Threading collects messages of a conversation scattered throughout the mailbox and presents them as one entity. Viewing by thread provides the history and context of each message by allowing the user to select one message and access all messages from that conversation thread [57]. Therefore, threading helps the user better understand a conversation and track its status.
Most threading systems identify threads based on structural information within the individual messages. Certain headers in email messages are used to recognize threads. *Message-ID* is a globally unique code for a message. This field is generated either by the MUA (Mail User Agent) or the first MTA (Mail Transport Agent) the message passes through [29]. *In-Reply-To* contains the message-ID of the message to which this message is a reply, and the *References* field contains a list of message-IDs of previous messages to the current message. Another way to recognize threads is by Subject line. After stripping “Re:” and “Fwd:“ from Subject field, all the messages of a thread will have identical subjects. [42]

Another way to recognize a thread is from the sequence of previous messages often quoted in the reply message. Lewis and Knowles [32] propose that threading of electronic messages be treated as a language processing task. In order to recognize threads, they apply standard text matching methods to the Subject line, and the quoted and unquoted parts of the message body.

However, the above approaches to identify messages of a thread are error-prone. There are some reasons for these errors:

- Email clients are inconsistent in the usage and format of the headers that lead to inaccuracy in finding threads based on headers [32]. There is also no standard convention for the clients to include the context or history of previous messages in a reply [57].
- Threads are frequently incomplete since the user may delete some messages of a thread or not save sent messages. Therefore, different participants of a thread may have different collections of the thread messages. [46]
- To avoid retyping the recipient address, users frequently reply to a message instead of starting a new one. We study this phenomenon in our Email User Survey (Chapter 5). If the user does not change the subject, the reply would be misinterpreted as part of a thread. Even in a better case when the user uses a new subject line, the In-Reply-To field may still refer to the parent message. This is usually invisible to the users but if a method were to use this field to find messages of a thread, errors might occur.

We identified threads as collections of messages with the same topic. The *topic* of a thread is the subject line of each message after stripping all the “Re: “ and “Fwd: “ tags from its beginning and end. By comparing these topics, we find messages with identical topics that we define as belonging to the same thread. To increase the reliability of this method, we then compare the senders and recipients of those messages to make sure that the sender of a *child*
message is among the recipients of the *parent* message. If this is not the case, we conclude that
the messages do not belong to one thread despite sharing the same topic.

### 4.2.2 Information Visualization Methods

Electronic mail is a complex collection of high volumes of information with many
different relations among its items. The textual nature of messages and the relations among them
(textual elements of a mail system) suggests the use of Document and Graph Visualization
techniques to visualize content and structure of messages. The idea is to view email messages in
the context of their thread, present the contents of messages, and show their connections using a
diagram.

Email messages have several features that can be visualized. Some are numeric data such
as message size, number of attachments, or date. Some are nominal including sender, email
content, subject, or thread. However, most attributes can be considered as ordinal data as well
such as date, sender, or subject. Among these data, we have focused on visualizing conversational
threads, messages in a folder, attachments, and size of the message.

We have designed a multiple-view interface for our prototype to be able to present all the
data at several levels of detail. Views provide different ways to look at information by putting it
in different arrangements and formats. Baldonado et al. [1] define a multiple view system as the
presentation of a single conceptual entity in two or more distinct views. The user may discover
hidden relationships in data and perform better with a multi-view interface. The views might
display different sets of data or one data set at different levels of abstraction. The controls of
different views are tied together so that actions in one view affect other views as well. A common
interaction technique is linking or brushing in which highlighting an item in one view highlights
the same item in the other views.

The visual structures that can be used for email interfaces include lines, areas,
connection, proximity, enclosure, position, size, grey scale, color, and time. Viewpoint Controls
such as detail-in-context help users interact with the interface.

### 4.2.3 EzMail Development

The interface between application programs and the operating system, often called API
(Application Programming Interface), provides access to the services of the operating system for
the application. APIs for email are particularly useful to add messaging functionality to other
applications such as word processors or mail-enabled applications.
MAPI (Messaging API), originally designed by Microsoft, is a complete architecture to serve as the basis for information exchange applications. It is a standard library of messaging functions that allows access to the mail infrastructure from Windows applications [41]. It supports full-featured client applications and service providers. However, for simpler client applications, MAPI Programming Interfaces (Figure 4-1) have been designed to allow messaging clients based on MAPI to interact with service providers.

![Figure 4-1 MAPI Architecture](image)

There are several programming interfaces for MAPI as shown in Figure 4-2. Simple MAPI and CMC (Common Messaging Calls) provide a limited set of messaging functions to support a basic level of messaging functionality. CDO (Collaboration Data Objects) is an additional scripting interface to the MAPI model to simplify the creation of applications with messaging functionality by avoiding the low-level details. CDO libraries expose programmable messaging objects such as folders and messages, which are extensions to programmable objects of visual basic and can be used to implement messaging functionality in an application. [36]

![Figure 4-2 MAPI Programming Interfaces](image)
We have used Microsoft Visual Basic as the programming language to develop our prototype. Visual basic is an object-oriented language and development environment for building complete and functional Windows applications. We made extensive use of MAPI and its programming interfaces for the development of EzMail. We employed CDO libraries to let EzMail communicate with the email client. The weak point with CDO is that it only supports Microsoft Outlook. This was not ideal but CDO was convenient to start the development of our prototype to prove the concept.

Since EzMail communicates with an email client to access messages that are already downloaded, client to host bandwidth is not an issue in EzMail. For this implementation, running on a PC with a 1.7 GHz Pentium 4 under Windows 2000, it takes about 10 seconds to read, sort into threads, and display about 400-500 messages from a folder in Outlook to EzMail. This would of course vary with the number of messages, number of threads, and the degree of scattering of the messages of threads.

**4.3 EzMail Design**

**4.3.1 Overview of EzMail**

EzMail is an email visualization tool that runs in conjunction with a user’s default email client and creates a multi-view interface for email messages. We chose to design a companion tool rather than create a complete email client from scratch because it was a more convenient way to develop and test our design concepts in a shorter time. Our primary focus is to provide thread visualization and thus the structure, context, and history of the conversations.

The main window of EzMail (Figure 4-3) consists of five interconnected views to display the data at several levels of detail, from a broad overview to a fine structure, and provide a kind of focus + context overview. The *folder view* lists the personal email folders created by the email user. The *list view* presents messages of a folder sorted by date and grouped by thread. Contents of a selected message are shown in the *message view*.

The *greeked view* displays a much-reduced image of the contents of messages to give an overview of their structure in a smaller space. Each message is encoded as a line whose length represents length of the message.
The *node view* represents thread information in a graphical hierarchy that reveals the linkage among messages. Each *frame node* shows a singleton message or the root message of a thread containing a series of replies represented by *box nodes*.

Graphical views such as greeked and node views allow adding visual information cues to represent different attributes of a message such as attachments and annotations.

Two additional views are shown in separate windows. The *thread view* collects and displays messages of a thread from all folders. The *sender view* collects and displays messages to or from a correspondent regardless of folder.

In addition to these visualizations, EzMail provides two functions to support task and communication management. One is the capability of adding annotation to a message, which may serve as a reminder or to-do. The other is the possibility of a simultaneous reply to two or more messages that let the user gather two or more threads together.

---

**Figure 4-3 EzMail Interface**
4.3.2 EzMail Interface

4.3.2.1 Main View

The main window (Figure 4-3) consists of five resizable views that are interconnected by linking, i.e., if a message is selected in the list, greeked, or node view, the same message is highlighted in the other two views. At EzMail start-up, the list of folders and new (unread) messages, if there are any, are retrieved and displayed. The status bar at the bottom of the view shows the number of messages in the selected folder. To provide more information, the additional views (thread, sender, and annotation) are available through menu bar and pop-up menu.

4.3.2.2 List View

The list view (Figure 4-4) shows a list of all messages in a folder. Each row in the list represents a message whose attributes (subject, sender, and date) are displayed in three columns. The main idea is to view messages as part of their threads. Therefore, the messages are clustered by thread topic while sorted by date in descending order. However, a linear list of messages (without thread grouping) may be achieved by sorting the list by sender or date in ascending or descending order. To revive the thread grouping, the user may click the ‘Subject’ column header.

```
<table>
<thead>
<tr>
<th>Subject</th>
<th>From</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reminder: Ope</td>
<td>Richard (Hao)</td>
<td>10/29/2003 10:10</td>
</tr>
<tr>
<td>Re: Reminder</td>
<td>Richard (Hao)</td>
<td>10/29/2003 5:05</td>
</tr>
<tr>
<td>Re: Reminder</td>
<td>David Favegh</td>
<td>10/29/2003 4:54</td>
</tr>
<tr>
<td>GruVi open ho</td>
<td>Hilary Zhang</td>
<td>10/28/2003 12:03</td>
</tr>
<tr>
<td>Open House</td>
<td>Richard (Hao)</td>
<td>10/28/2003 12:30</td>
</tr>
<tr>
<td>Reminder: GruVi</td>
<td>Richard (Hao)</td>
<td>10/24/2003 10:45</td>
</tr>
<tr>
<td>Open house</td>
<td>Soleil Lapierre</td>
<td>10/22/2003 7:19</td>
</tr>
<tr>
<td>GruVi Lab</td>
<td>Richard (Hao)</td>
<td>10/7/2003 12:57</td>
</tr>
<tr>
<td>RE: GruVi Lab</td>
<td>Bob Smith</td>
<td>10/24/2003 11:46</td>
</tr>
<tr>
<td>RE: OpenHouse</td>
<td>Richard (Hao)</td>
<td>10/2/2003 1:08</td>
</tr>
<tr>
<td>RE: OpenHouse</td>
<td>Ken Chidlow</td>
<td>10/6/2003 4:40</td>
</tr>
<tr>
<td>RE: OpenHouse</td>
<td>Haris Widjaya</td>
<td>10/6/2003 10:04</td>
</tr>
<tr>
<td>RE: OpenHouse</td>
<td>Torsten</td>
<td>10/5/2003 1:47</td>
</tr>
<tr>
<td>RE: OpenHouse</td>
<td>Soleil Lapierre</td>
<td>10/5/2003 1:27</td>
</tr>
<tr>
<td>Open House</td>
<td>Soleil Lapierre</td>
<td>10/1/2003 12:43</td>
</tr>
</tbody>
</table>
```

Figure 4-4 List View

The messages in a thread are displayed in a tree structure. The root message is the root of the tree and the replies are displayed as children of the root thus indented comparing to other
messages. If the root message of a thread is not found (either deleted or filed in another folder),
the oldest message in the thread, which is a reply, is counted as the first message of the thread and
displayed as the root of the tree. The replies in each thread are sorted by date in descending order.
This way, the user can view the first and last (most recent) messages of the thread at a glance that
gives a quick overview of the conversation status.

There are visual icons at the beginning of each row representing the status of a singleton
message as ‘read’ or ‘unread’. The icons for messages of a thread indicate whether the message is
a ‘child’ or ‘root’ in the thread. Unread messages are shown with bold font.

It is possible to select more than one message using shift or ctrl keys. The selected
messages are highlighted and their contents are shown in horizontally tiled text boxes in the
message view (Figure 4-5). This feature is useful in replying to more than one message.

![List View: 'Office'

<table>
<thead>
<tr>
<th>Subject</th>
<th>From</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wireless Innovation</td>
<td>Mehrdad Sail</td>
<td>12/9/2003 2:45</td>
</tr>
<tr>
<td>Graduate awards...</td>
<td>Raj Pabla</td>
<td>12/5/2003 9:04</td>
</tr>
<tr>
<td>Re: Graduate a...</td>
<td>Raj Pabla</td>
<td>1/8/2004 11:23</td>
</tr>
<tr>
<td>Re: Graduate a...</td>
<td>Mary Nelson</td>
<td>1/7/2004 10:35</td>
</tr>
<tr>
<td>Re: Fw: Graduate...</td>
<td>John Dill</td>
<td>12/12/2003 12:</td>
</tr>
<tr>
<td>Re: Graduate a...</td>
<td>Raj Pabla</td>
<td>12/10/2003 10:</td>
</tr>
<tr>
<td>Re: Graduate a...</td>
<td>Raj Pabla</td>
<td>12/8/2003 11:5</td>
</tr>
</tbody>
</table>

![Message View]

Subject: Re: Graduate awards and scholarships
From: Raj Pabla
Hello all,

As this might be your problem too I send this message to all of you.
You can definitely apply for both. Being accepted for one does not affect the other one.

Best Wishes,

Subject: Re: Graduate awards and scholarships
From: Mary Nelson
Hi Raj,

I have applied for fellowship this semester. I wonder if I still can apply for the grad scholarships as well.

Thanks for your help
Mary Nelson

Figure 4-5 Selecting Two Messages
4.3.2.3 Message View

The message view (Figure 4-6) consists of two parts. The top pane displays message header information including subject of the message, sender, list of recipients, and list of attachments if there are any. The bottom pane shows contents of the message. If two or more messages are selected in the list view, their contents are displayed in horizontally tiled text boxes in the message view (Figure 4-5).

![Message View]

Figure 4-6 Message View

4.3.2.4 Greeked View

The idea of the greeked view (Figure 4-7) comes from greeked text, which means “displaying text as abstract dots and lines in order to give an overview of layout without actually being legible” [11]. Using graphics symbols to approximate the text is faster than writing the characters correctly that may require scaling or other transformations. Greeking is particularly useful when displaying a reduced image or a preview of a document where the text would be too small to be legible on the display anyway and giving an overall appearance of the document is sufficient. [56]

Our inspiration for the idea of the greeked view was SeeSoft [15] in which lines of code are displayed as thin rows (Figure 2-12). The greeked view displays a much reduced image of the messages that appear in the list view. They are shown as a list of rows where each row is a line that represents a message. The length of the line is proportional to the size of the message excluding the attachments. The replies in a thread are indented as in the list view. This
presentation gives an overview of the structure of messages in a smaller view and allows comparison of messages based on their size.

The greeked view always follows the same order of messages as in the list view. Therefore, if the list view is sorted by date or sender (thus losing grouping by thread), the greeked view is updated by listing the messages in the same order and without any thread grouping. Selecting a message line highlights it (in blue) in this view and, via linking, in the list and node views as well. A mouse-over a line shows the related message sender, subject, and date. Unread messages are shown using thicker (bolded) lines.

In the greeked view, each attachment is represented by a small vertical blue line at the lower right corner of the message line, making it easy to determine the number of attachments for that message. Similarly, annotations are represented by small vertical lines at the upper right corner of the message line. White, yellow, or red shows the priority of the annotation (normal, high, or very high, respectively).

![Greeked View](image)

**Figure 4-7 Greeked View**

4.3.2.5 Node View

The node view (Figure 4-8) is a more detailed graphical representation of messages in the list view. We have used clustering to group data based on thread topic and enclosure as a visual structure to show messages of a thread. Any singleton message or root of a thread is represented by a frame node whose title is the subject of the message. The replies in a thread are shown as box nodes inside the root frame and in the same order as the list view. This visualization gives a clear presentation of messages in a thread while visualizing singleton messages as well. The consistent presentation of all messages allows the use of visual cues to represent attributes of a
message. We have used a simple layout to show the structure of messages by placing the frame nodes following each other in a chronological order (descending), and clustering the messages of a thread inside their frame in a temporal order from left to right; this reduces space needed for nodes and interface clutter, giving a clear view of messages and threads. This thread visualization provides a less cluttered interface as well as the opportunity to create single actions for group of messages. Another advantage of our visualization is that it is stable and results in similar visual presentations in different runs.

The node view always follows the same order of messages as in the list view. Therefore, if the list view is sorted by date or sender (thus losing grouping by thread), the node view is updated by representing all messages with frame nodes and without any thread grouping. Selecting a node highlights the message in this view and all other views. A mouse-over a frame node displays the sender and date of the message while a mouse-over a box node displays the sender, subject, and date of the message. Unread messages are indicated by bold text for a frame node and by an un-pressed box node for a reply. Even receiving new replies in a thread does not really change the presentation since replies are enclosed in the root frame node.

![Figure 4-8 Node View](image-url)
The width of a frame node is proportional either to the character length of the subject line or to the width of the box nodes inside, whichever is bigger, and limited to the width of the node view. If the length of the subject is longer than the width of the node view, the subject cuts off but the whole subject will be shown by resizing the view and also in a mouse-over tooltip. The box nodes are displayed in several rows if they do not fit in one row.

Each small blue square attached to the lower right corner of a node represents an attachment. Clicking on an attachment cue shows a list of attachments in text form. A mouse-over an attachment cue shows the filename of that attachment. A small white, yellow, or red square at the upper right corner of a node indicates that the message is annotated with normal, high, or very high priority respectively. The annotation view is accessible for reading or editing the annotation by double clicking the visual cue.

4.3.2.6 Thread View

The thread view has been designed to implement the threading feature. In the main view, we can only see messages in one folder at a time. As a result, when we view one thread we can only see those messages of the thread that are in the selected folder. There might be other messages from that thread scattered in other folders. To address this problem and simplify working with threads, we have designed the thread view to collect and display messages of a thread in one view regardless of folder. The thread view is accessible from the main menu or a pop-up menu in the list, greeked, and node views. The topic of the thread is shown in the window header.

The thread view (Figure 4-9) consists of list, node, and message views, similar to the main view. The list view displays a list of messages in the thread sorted by date and grouped by thread. Three columns in the list view show sender, date, and folder for each message. The indentation and visual icon of each row shows if the message is a root or a child in the thread. If the root message of a thread is deleted, the oldest message in the thread, which is a reply, is displayed as the first message of the tree. The list may be sorted by each column header; however, the ‘Sort Thread’ button in the toolbar revives the original tree view of the thread.

The message view displays the contents of the selected message with the header information (subject, sender, recipients, and attachments of the message). The node view here is a bit different from the main view since each frame node represents a folder and the box nodes inside the frame are those messages of the thread filed in that folder. With this view, we can see the folders containing the thread and the messages in each folder. Selecting a frame node
highlights all messages from that folder while selecting a box node highlights that single message in the node and list views. A mouse-over a box node shows the sender, subject, and date of the message. The attachments and annotation cues similar to those in the main view are used to visualize and access these attributes of a message. Unread messages are signified by using bold text in the list view and un-pressed box nodes in the node view. The window and the individual views inside the thread view are resizable.

4.3.2.7 Sender View

The sender view (Figure 4-10) has been designed to collect and display all messages to or from a correspondent regardless of folder. The sender view is accessible from the main menu or a pop-up menu in the list, greeked, and node views. The name of the correspondent is shown in the window header.

The design of the sender view is quite similar to that of the thread view. It consists of list, node, and message views similar to the main view. The list view displays the list of all messages for which that correspondent is the sender or among the recipients; messages to the person are colored grey. The list is sorted by date and clustered by thread. This is another opportunity to view all the conversations in which a correspondent has taken part. Three columns in the list view show subject, date, and folder for each message. The indentation and visual icons are used to
show whether a message is read, unread, the root or a child in a thread. The list may be sorted by date or folder in ascending or descending order. Clicking the ‘Subject’ column header will revive the original situation where messages are sorted by date and grouped by thread.

The message view displays the contents of a selected message with the header information (subject, sender, recipients, and attachments of the message). The node view is similar to the one in the thread view. A frame node represents a folder and the box nodes inside the frame are the related messages filed in that folder. Clicking on a frame node highlights all messages from that folder while clicking on a box node highlights that message in the node and list views. A mouse-over a box node shows the sender, subject, and date of the message. Attachments and annotations are supported as in the main view. Unread messages are bolded in the list view and indicated by un-pressed box nodes in the node view.

![Sender View](image)

**Figure 4-10 Sender View**

### 4.3.3 EzMail Functions

#### 4.3.3.1 Annotating a Message

One of the main challenges in email is the issue of reminders and to-dos. An email client should provide proper tools to help with these email-related tasks.

The ability to annotate a message allows a user to use it as a reminder or to-do. We have added an annotation view (Figure 4-11) to our prototype, available for each message from the
main view. The subject is shown in the window header and a text box allows entry of notes. A toolbar is provided to set the priority of the annotation to normal, high, or very high, save or delete the annotation, or cancel the action. Visual cues in node and greeked views are updated as soon as an annotation is added or removed.

Annotations are saved as a message property and are available whenever the user opens EzMail again, though they are not accessible in Microsoft Outlook.

![Annotation View](image)

**Figure 4-11 Annotation View**

### 4.3.3.2 Replying to Several Messages

Participants in conversations often want to reply to several related messages simultaneously, for example to gather the messages together and perhaps create a new thread. Since the messages need not be in the same thread, this allows gathering threads together. This also allows recipients of this new message to view the gathered messages from other participants. Most email clients support only 'divergent' conversations; topics may be added and refined, which differ from face-to-face discussions where participants pull ideas from different discussion threads to summarize or to create new ideas. In fact, drawing conclusions, reaching agreements, creating hypotheses, and summarizing results are the main objectives of many discussions. The ability to reply to several messages is a powerful function that adds considerable utility.

We have provided this functionality in EzMail. The user can select two or more messages in the list view and reply simultaneously to all of them. The reply view (Figure 4-12) displays the senders and subjects of the selected messages in ‘To’ and ‘Subject’ lines at the top pane. The original messages from all senders will also be displayed in the text box where the user can type the reply. The same view is used to reply to a single message.
4.4 Discussion

In this section we describe the information visualization methods used to develop EzMail and compare and contrast EzMail with the visualization systems reviewed in chapters 2 and 3.

4.4.1 Information Visualization Methods and Systems

Our email visualization research applied information visualization methods to visualize mailbox contents, thereby helping users gain insight into their messages.

Referring to the information visualization model (Figure 2-8) our Raw Data is a mailbox with email messages as a set of relations with a structural format. We used message IDs to represent data items and message properties such as sender, subject, date, topic (thread topic), type (root, parent, child, or single), annotation, and attachments as attributes with different values to be visualized.

The Visual Structures (Figure 2-8) used were ‘areas’ (messages in the node view), ‘lines’ (messages in the greeked view), and ‘points’ (attachments, and annotations). Visual properties that show relations among the data objects were proximity (visual proximity of messages sorted
by date, sender, or thread in the list, greeked, and node views), enclosure (thread clustering in the node view), position (indentation in the list and greeked views), size (message length in the greeked view), and color (attachment, annotation priority, and highlighting). Location Probes such as brushing and tooltips helped with viewing the properties and relations of the visual cues.

The final stage (Figure 2-8) was presentation of these visual structures in a multi-view interface that gives a kind of focus + context overview of the messages. The interaction paradigm is direct manipulation.

One of the information visualization systems we reviewed was SeeSoft [15], a software visualization system. It inspired our use of lines to represent messages in the greeked view. However, SeeSoft uses other methods of visualization such as folding and recursion that do not seem appropriate for email since software codes might be huge in comparison to email messages in a thread or folder.

The document visualization methods we reviewed mostly fall into two categories. The first group includes Zoom Browser [25], CZWeb [16], and Pad++ [4], whose main concern is the visualization of either a document collection or pages of a document. They represent pages as 'areas' that can be zoomed in and viewed simultaneously. The visualization in EzMail is somewhat similar to these document visualization systems. ‘Areas’ (frame nodes and box nodes) are used to visualize messages in the node view. However, we did not apply zooming to these visual cues since the contents are shown in another view (message view). One difference between these systems and ours is that email messages are normally not as long as documents and web pages. Another difference is that working with documents and web pages, the user may want to view many pages simultaneously but an email user usually only needs to view a few messages (e.g. messages of a thread) at the same time. Therefore, different issues may be considered for email visualization.

The second category of document visualization tools including ThemeRiver [22], ThemeView [59], and Galaxies [59] focuses on visualization of thematic variation of large collections of documents. To develop an analogous visualization for email, a thorough content analysis of the email messages would be required, and this was beyond the scope of this thesis but could be considered for future work. This kind of visualization could help with revealing the social connections of email messages and users.

Graph visualization was also needed in EzMail to visualize the relations among messages in a thread. The graph visualization methods we reviewed were mostly focused on large graphs.
However, in the case of email threads, simple graphical presentations suffice since the number of nodes (messages) in an email thread is small (80% of messages are in threads of size 5 or smaller [29]). We have used a simple layout to show the structure of messages by placing the frame nodes following each other that keeps the chronology of the messages and clustering the messages of a thread that reduces the allocated space to nodes.

### 4.4.2 Email Visualization Methods

Considering email patterns of use discussed by Venolia et al. [54], EzMail thread visualization seems best suited for triage (by grouping related messages together in the list and node views thus reducing the volume of items to be dealt with), task management (by providing annotations to serve as reminders and to-dos), and message retrieval (by thread and sender views to retrieve desired messages by having one message of the group). In addition, as we discussed in section 3.1.3, threading assists communication management, information management, and task management. Among the qualities that Kerr [29] suggests for a thread visualization (section 3.1.3.1), EzMail supports chronology of messages, stability of visualization, and scalability.

All the email thread visualization systems we reviewed, such as ReMail [46], Thread Arcs [29], Netscan [49], and Mixed-Model visualization [55], use tree structures to visualize threads. This is different from the frame nodes and box nodes that we have used in the node view. The node view presents thread grouping while the greeked view gives a sense of thread structure in the context of all messages. Line coding of email messages used in the greeked view is new and has not been used before. The visualization in the node view gives a clear presentation of messages in a thread while visualizing singleton messages as well. However our review of those papers suggests they were focused on thread visualization and appeared not to take folders into consideration. The systems may provide presentation of thread messages from all folders but the papers do not mention it.

The collection of functions we developed in EzMail has not been provided in any commercial email client that we looked at. Annotation capability does not exist in Microsoft Outlook and similar email clients. Rohall et al. have added annotation feature to their developing prototype. Replying simultaneously to two or more messages, as in EzMail, has not been proposed in the research that we reviewed, though we recently found that Eudora has a similar feature.
Other thread visualization tools such as Loom [12] and Conversation Map [48] present the social structures of Usenet conversations that might be investigated as part of the future work for EzMail. Another group of email visualization systems, such as Timestore [61] and TaskView [20] provide a time-based interface that is not similar to what we have in EzMail. The last group of email clients we studied offer automatic classification of messages such as neo [39] and Bifrost [2], which is not the focus of EzMail.

In summary, EzMail’s visualizations for singleton messages, group of messages, and attributes of messages in the greeked and node views are novel and different from other email visualization systems. The threading features of EzMail (thread view, replying to several messages, and clustering by thread in the list, greeked, and node views) are helpful in tracking, converging, and concluding conversations. The annotation feature assists EzMail user with task management.

However, to validate and improve our design ideas for EzMail, we conducted a small user survey to explore email users’ habits and preferences in more detail.
CHAPTER 5

EMAIL USER SURVEY

In order to explore how email users manage their messages, we conducted a small user survey asking the participants to complete a questionnaire and describe their general and specific email habits, preferences, and difficulties. The subjects were asked to describe their general email behavior, how they file and retrieve their messages, if they use automatic filtering to categorize messages, how they manage conversational threads, and how difficult they find management of attachments. This chapter discusses the methods and results of this user survey.

5.1 Design and Methodology

To study users' email management behavior, we prepared a questionnaire and conducted interviews with the participants. In designing the questionnaire we focused on areas of concern to us in our prototype design. The questionnaire consisted of five parts: general, foldering, threading, attachments, and email management.

The first part asked about general habits of users such as how often they check email, how much time they spend dealing with email, and how many messages they send and receive daily. The goal of the 'foldering' part was to determine the structure of users' email. We asked if they use folders to organize their email, how big their inbox and folders are, and how they manage filing and retrieving messages into and out of folders. The aim of the 'threading' part was to discover whether the subjects are familiar with email threads or conversations and use this capability. There were also some questions to see how much we can rely on subject line to find
threads. In the ‘attachments’ part we asked about the problems subjects have managing their email attachments. The last part covered the remaining problems and subjects’ suggestions regarding email management.

We selected our subjects from academia and industry. We were looking for people who had a large volume of sent, received, and archived email. There were six academics among our subjects: three professors and three students from the schools of Computing Science and Engineering Science at SFU. The rest (eight) were directors or managers from industry. Altogether, we had 14 subjects who mostly were high-volume users of email.

We conducted interviews with those who were available and sent the questionnaire by email to those who were not local. Each interview took about 45 minutes and was held in the subject’s office on her/his computer. Subjects were asked to open their email program to answer certain questions and at the same time, give us an opportunity to get an overview of their email structure. In some cases, a snapshot of their email folders and message structures was requested. However, supplying snapshots was entirely voluntary and there were no negative consequences if a subject wished not to supply such snapshots. In no case was the content of any email message requested.

During a session, the subject was provided with a copy of the questionnaire and could thus follow the questions easily while the interviewer was reading the questions to the subject and recording her/his answers. The subjects were encouraged to explain their responses as much as possible. In some cases, additional questions were asked based on the subjects’ replies to gain more insight into their usage of the email program. Right after each session, the subject’s answers were entered into a computer file to keep an electronic copy of the results.

We used the first subject as a pilot to determine the approximate time for the interview and to improve the questionnaire as well. In fact, we tried to improve the questionnaire after the first few interviews; remove questions that were not a real help to us or add questions that occurred as we gained experience with the preliminary results of the study.

5.2 Results

The goal of our user survey was to explore how the email users manage their messages, email folders, automatic filters, and conversational threads. We used the results of this survey to validate and refine the design ideas we developed for EzMail.
5.2.1 General

The first part of the questionnaire provided us general information about the behavior of subjects regarding their email usage. The first observation was the variety of email programs people use (Table 5-1). There were 14 different programs for our 14 subjects. Each user uses from 1-4 different programs based on the platform she/he happens to be using. All subjects use a client-based email program, 11 of 14 work with a Windows system, half use Microsoft Outlook, and 6 of 14 use at least one web-based email program in addition to their client-based email system. All industry subjects (eight) use a Windows email system and among them, 75% use Microsoft Outlook. Those who use text-based programs are from academia and comprise 67% of the academics subjects. Only one subject from academia uses a Macintosh system.

Most subjects (93%) check their email from different locations, basically home and office. This is more of a hassle for client-based email users who have to download their email onto every workstation on which they work with email. The advantage of a web-based email system is that the messages are kept on the server and handled once. Just one subject checks his email only in the office.

Half the subjects check for new mail more than once per hour (Figure 5-1). Some of these said they check their email constantly when at their computer. They receive automatic notification of incoming mail. Some users prefer not to be distracted by a visual or audible notification (Figure 5-2).
Table 5-1 Email Programs Popularity

<table>
<thead>
<tr>
<th>Email Type</th>
<th>Operating System</th>
<th>Email Program</th>
<th>Number of Users</th>
<th>Percentage</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client-based</td>
<td>Windows</td>
<td>Microsoft Outlook</td>
<td>7</td>
<td>50%</td>
<td>79%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Netscape Messenger</td>
<td>2</td>
<td>14%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lotus Notes</td>
<td>2</td>
<td>14%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Outlook Express</td>
<td>1</td>
<td>7%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bat!</td>
<td>1</td>
<td>7%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Neo</td>
<td>1</td>
<td>7%</td>
<td></td>
</tr>
<tr>
<td>Unix</td>
<td>Fine</td>
<td>2</td>
<td>14%</td>
<td>43%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mail tool</td>
<td>2</td>
<td>14%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mutt</td>
<td>1</td>
<td>7%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Elm</td>
<td>1</td>
<td>7%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macintosh</td>
<td>Entourage</td>
<td>1</td>
<td>7%</td>
<td>7%</td>
<td></td>
</tr>
<tr>
<td>Web-based</td>
<td>Webmail</td>
<td>3</td>
<td>21%</td>
<td>43%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yahoo</td>
<td>3</td>
<td>21%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hotmail</td>
<td>2</td>
<td>14%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-1 Email Checking Frequency

Figure 5-2 Automatic Notification
Most subjects spend a significant amount of their working time dealing with email (Figure 5-3) and spend more time on email at work than at home. Half the subjects spend 1-2 hours per day to handle their email. On average, users spend about two hours a day dealing with their email, about 25% of an 8-hour working day.

![Figure 5-3 Email Usage Time]

Figure 5-3 shows the results of our query regarding daily volume of sent and received email. The average number of sent messages is about 20 while that of received messages is 60.

![Figure 5-4 Number of Email Exchanged Daily]

5.2.2 Foldering

We asked both general and detailed questions about the structure of users' personal email folders and messages. We observed that folders are used in two different ways: one is to archive old or passive items and the other is to organize active email messages. Among the subjects, three do not really use email folders to categorize their messages but to archive their old items. They make use of the inbox to keep track of messages so they have bigger inboxes in comparison to others. One of them finds it “time consuming to create folders”. He is also afraid of creating
redundant folders. From time to time he dumps a group of messages into folders categorized by date. Among the rest, two subjects do not have any email folders since one of them saves messages in file folders and the other uses folders created by the email client (neo); there is thus no need for additional folders. Another subject finds ‘views’ in Lotus Notes more important than folders (though he uses folders) since he uses his email “as a workflow management and filing tool”. He defines views as “sorted/grouped displays of the entire email database that allows rapid finding of communications with correspondents and order them by sender, date, etc.”. Altogether 64% said that they use email folders: “to organize communications with a variety of people for a variety of purposes”, “with the hope of finding messages quickly”, “to organize mail into hierarchical categories”, “to decide what to read now vs. later”, “to group similar messages for later reference, responding, or deleting”, and “to identify ‘action’ or ‘urgent’ messages”.

Figure 5-5 shows the number of folders our subjects have, excluding inbox, sent, and trash folders. Approximately half the subjects have 20 or more folders.

![Figure 5-5 Number of Folders](image)

Regarding folder structure depth, 43% have only one level, another 43% have 2-4 levels and two subjects (17%) have a 4-7 level deep folder structure.

The maximum number of messages in a folder has a wide range (Figure 5-6). Its average is about 500. We included the two subjects who do not have any folders with zero number of folders.
We also asked subjects the total number of messages in all folders (Figure 5-7). This number was very high for those users who keep most of their messages. The average was about 5000 messages. Excluding the lowest (30) and highest (22,000) numbers, the average is about 4000 messages.

On average, subjects have a high number of messages (about 650) in their inbox (Figure 5-8). Users keep incomplete tasks, unfiled information, and ongoing conversations [57] in the inbox. There is also temporary information, recent messages, notification of upcoming events, and just plain “interesting stuff” as one user put it, in some subjects’ inboxes. Figure 5-9 shows the percentage of messages that subjects keep in the inbox in comparison to all of their messages. For example, the first column shows that 50% of subjects have less than 10% of their messages in the inbox. Apparently the users of folders have less number of messages in their inbox. Figure 5-10 shows that users with highest number of messages in the inbox are those who have fewer folders. As we note in the discussion section of this chapter (5.3), such users belong to the group of archivers.
Figure 5-8 Number of Messages in Inbox

Figure 5-9 Percentage of Messages in Inbox

Figure 5-10 Percentage of Messages in Inbox vs. Number of Folders
We asked subjects about the categories they use to create folders. Figure 5-11 shows the percentage of subjects that use sender, topic, project, date, organization, and other. The subjects mentioned threads, workflow, attachment size, and research groups in the 'other' category. A larger number of subjects (9 of 14) use sender and topic as criteria to create folders. EzMail's sender and thread views use the same categories to collect similar information in one place and make it easier for the user to find the desired information.

![Figure 5-11 Folder Categories](image)

We asked about methods used to retrieve messages from folders, with visual scanning and search being the two most popular methods (Figure 5-12). In the 'other' category, one subject mentioned viewing by thread. They only (usually) resort to search when they do not have enough information to find the message by visually scanning the list of messages. Therefore, our method of collecting information in one view for a sender or topic simplifies visual scanning and searching the results.

![Figure 5-12 Methods of Retrieving from Folders](image)

Some subjects find it relatively easy to retrieve messages from folders (Figure 5-13). It gets difficult when the user does not remember the right keyword to look for. The challenge is
finding the right folder. Based on our observations, subjects who use text-based email programs, such as Pine, or have lots of email find it more difficult.

![Difficulty of Retrieving from Folders](image)

**Figure 5-13 Difficulty of Retrieving from Folders**

We asked our subjects if they use automatic filtering to categorize their messages and 42% of the replies were negative. The rest use it for filtering spam or messages from specific senders or mailing lists.

The subjects gave some useful suggestions on foldering such as: “using a kind of color scheme to separate folders and distinguish archive folders”, “linking related messages from different folders, those from the same higher level folder but filed in different subfolders”, and “having simple google-like search”. Again, what we have done in the thread and sender views is linking messages related by subject or sender from different folders.

### 5.2.3 Threading

We asked about the methods used to track a conversation or thread and how difficult this tracking was. Three of the subjects do not use threading. The rest do not find it very difficult (Figure 5-14). A possible explanation is that they usually follow recent conversations that might still be available from the inbox. Figure 5-15 shows methods of tracking a thread. We included three subjects who do not use threading by adding a ‘none’ column. ‘Grouping’, by which we mean collecting messages of a thread and presenting them to the user, was not very common, though it was often felt to be helpful. Only two subjects (not Outlook users) make use of grouping to follow a thread. Microsoft Outlook and Netscape Messenger have an option to view messages of a folder by conversation. Outlook has another feature to ‘find all related messages’ to a selected message (messages in one thread) in folders that the user selects. We believe that either the feature allowing finding messages from all folders does not exist for the email programs our subjects use or if it does, the users are not familiar with this feature since it may not be easily
accessible or visible as in Outlook. The list view in EzMail provides messages grouped by thread. The thread view groups messages of the same subject in another view. By providing visual proximity, these features help with grouping messages from a thread and making them visible to the user.

![Figure 5-14 Difficulty of Threading](image)

In the threading section of the questionnaire, we included questions to study users' habits in replying to messages. The replies to these multiple-choice questions were chosen from 'rarely' (1) to 'quite often' (5) or 'always by default' (6).

An important problem with threading is the incomplete set of messages of a thread because of 'missing' messages, either as a result of users choosing to delete some received messages or not saving sent messages. We found that almost all subjects save sent messages 'always by default' (5.79). This can assure us that we have usually access to sent messages of a thread unless the user occasionally deletes them.

On average, the subjects include the original message in their reply 'quite often' (5.07). This is an advantage that can be used to detect to which message a message is a reply. We did not
make use of it since it needs complex text matching strategies and is beyond the scope of this thesis.

Most subjects are satisfied with the auto-completion capability of their email client for email addresses. But they ‘often’ (3.29) reply to a message instead of creating a new message to avoid retyping the recipient’s email address. When their reply is not relevant to the original message, they ‘usually’ change the subject (3.86) and delete the original message (3.79). The habit of replying to an unrelated message increases the number of unreliable messages. By unreliable here we mean those messages that may be interpreted as part of a thread, though they are not. That is why we do not use the ‘In Reply To’ header of a message to find a thread in EzMail. We cannot completely rely on the subject line either, though the users tend to change the subject when it is irrelevant. To increase the reliability of a thread in EzMail, we also check the sender and recipients of messages of a thread to be consistent. We make sure that the sender of a child message is among the recipients of the parent message. If this is not the case, we conclude that the messages do not belong to one thread despite sharing the same topic.

Among the suggestions our subjects made about threading were “collecting sent and received messages together”. This is actually what we do in the thread and sender views since we collect messages from all folders including the sent items folder. Two of the subjects complained about “losing the link to the thread when they file it away”. By reading messages of a thread from all folders in EzMail this will not be a problem. Another suggestion was “grouping messages of a thread in a temporary folder” which is accomplished in EzMail by viewing messages of a thread in the thread view. Another subject finds a sender view in Lotus Notes to view messages by sender quite satisfying. This is also a confirmation of the usefulness of the sender view in EzMail. One of the subjects who uses a text-based email client finds meeting scheduling, which is a multi-persons thread, quite difficult.

5.2.4 Attachments

Most subjects find it easy in general to deal with attachments (Figure 5-16). The main complaint is dealing with very large attachments since this tends to overrun email quotas. One subject suggested an “automatic compacting of attachments”. Moreover, opening large attachments with dial-up connections is painful especially since client-based email programs usually do not provide information about the email and attachments before downloading. According to one of our subjects, a preview of senders and size of messages before downloading is helpful. Another complaint was versioning of the attachments. It is hard to follow the versions...
of attachments through several email messages. A solution is needed to control the version of attached documents.

![Graph](image1)

**Figure 5-16 Difficulty of Attachment Management**

We also wanted to know if people leave attachments mostly with messages or save them separately into file folders. Figure 5-17 and Figure 5-18 show the results.

![Graph](image2)

**Figure 5-17 Saving Attachments to File Folders**

![Graph](image3)

**Figure 5-18 Leaving Attachments with Messages**
5.3 Discussion

Foldering is used for two different purposes: archiving old or passive items and organizing active messages. This suggests categorizing our users as archivers and organizers. Archivers are those who keep their messages in the inbox and have a few folders to archive their old items from time to time. Organizers are those who have a well-structured folder system and use folders on a regular basis to organize their messages. This categorization may be compared to that of Whittaker and Sidner [57] who categorized users based on their foldering activity to no-filers, spring cleaners, and frequent-filers. The frequent filers are basically the same as organizers who make frequent use of folders. No-filers who have huge inboxes are somewhat similar to archivers. We did not encounter any spring cleaner among our subjects; those who have a foldering system but do not use it efficiently. Our archivers category is also different from that of Mackay [34]. She identified the users who keep all their messages and make use of folders as archivers.

We observed that some users are still not really familiar with threading and its benefits, possibly because threading features of current email clients are not yet well-developed. Therefore, implementing threading features should improve the usage of an email client.

We found users of email clients with the capability of viewing messages by sender, date, or other attributes (e.g. Lotus Notes and neo) quite satisfied with this feature, perhaps because it reduces the need to create folders. This validates the design of the thread and sender views in EzMail whose aim is viewing messages by topic and sender.

5.4 Summary

Our user survey found similar behaviors, problems, and complaints to those noted in previous research (sections 3.1.1 and 3.1.2):

- Email is one the main activities of the users. On average people spend 25% of their working time on email and check their email quite often (1/hour or more).
- People use quite a diverse range of email programs. Individuals may even use several different programs depending on operating system, location, type of email messages (personal versus official), and other factors.
• All our subjects use at least one client-based email program. Some also use a web-based email program. Microsoft Outlook is a popular email client, especially among people in industry.
• Using folders is prevalent but still not easy even for those who have a well-structured folder system. This is in agreement with [13] and [57] who noted difficulties in foldering.
• There are still users who are reluctant to use folders because they find it time-consuming and inefficient. They prefer to keep their messages in their inbox and find messages by visual scanning and search.
• On average, number of messages in the users’ inboxes seems high but those who use folders extensively have fewer messages in their inbox.
• Sender, topic, and project are the most common criteria to create folders.
• Visual scanning and search are the most used methods to retrieve messages from folders. Sort is the next in the list. Ducheneaut and Bellotti [13] found sort to be more popular than search. Apparently people use search more often now than before.
• Automatic filtering is used primarily to filter spam and messages from newsgroups.
• We found that users often reply to a message instead of composing a new message to avoid retyping or looking for the recipient email address. This habit increases the number of messages that may be misinterpreted as messages of a thread.
• Saving sent messages and including the original message in the reply happen quite often by default which could help in implementing threading.
• The main problems of attachments are size and versioning. This is in agreement with previous findings. [13][54]
• The results of this user survey are a confirmation of “email overload” phenomenon first declared by Whittaker and Sidner [57].

The results of this user survey and participants’ comments suggested EzMail’s design was at least a step in the right direction, and we felt justified in moving on to a more formal evaluation.
CHAPTER 6

EZMAIL USER STUDY

The goal of this user study was to evaluate the usability and usefulness of EzMail. By conducting controlled user sessions asking the subjects to perform certain representative tasks with EzMail and Microsoft Outlook, we determined whether the new visualizations and functions of EzMail help users manage their email threads and messages more effectively.

6.1 Design and Methodology

We used a between-subject experimental method in which each subject was assigned to a different condition. Microsoft Outlook served as the control and EzMail was the experimental condition. The goal was to compare the performance of subjects under these two conditions.

Each session of the user study consisted of a pre-session questionnaire, a training phase, a set of tasks, and a post-session questionnaire. The pre-session questionnaire was to determine the subject’s general email habits. The training session familiarized the subject with either Outlook or EzMail and provided a short practice period. Then the subjects were required to perform a set of on-screen tasks based on a scenario and answer the task-related questions. The session concluded with a qualitative post-session questionnaire to explore the subject’s opinion about EzMail or Outlook. The approximate time for each session was 45-60 minutes.

The tasks were designed based on a real scenario in a way that guided the subject through different aspects of EzMail and similar features in Outlook. A monitor program was designed to
present the tasks and the task-related questions to the subject. The time to complete a task and answer the questions following the task and the responses to those questions were recorded. A set of messages based on the scenario was created by the experimenter. The rest of the mailbox was populated with additional messages (about 1000) and folders (20-30) to make a more realistic context.

We wanted to evaluate EzMail visualizations for individual messages (message itself, attachments, annotations, and size of message), visualizations for group of messages (grouped by threads and senders), and new functions (adding annotations to messages, and replying simultaneously to more than one message). The dependent variables were the time to complete an assigned task, the number of task errors (incorrect or incomplete tasks), and the number of errors in task-related questions (incorrect response or no response).

We conducted a pilot study with four subjects to test the experimental design before the formal sessions. Two of the pilot subjects tested EzMail and the other two worked with Outlook. This testing indicated minor changes to the design of the monitor program were needed. For the main study, we obtained 12 subjects, four female and eight male, by sending email to SFU students in the schools of engineering and computing science. Half the subjects tested Microsoft Outlook and were required to have no experience with this program. We set this condition to balance the requirements for Outlook users and EzMail users by ensuring all users were unfamiliar with the environment they were going to use.

6.1.1 Scenario

Several requirements had to be met in designing the user study. First of all we wanted the subject to participate in a threaded conversation since threading was a main emphasis of EzMail. To create a realistic context for the conversation, a realistic scenario was needed. At the same time, the tasks were to fulfill the goal of the study that was testing the new features and functions of EzMail. For Outlook users we had to consider more or less similar features to accomplish the tasks. Therefore, based on these ideas and requirements, we developed a scenario and a set of related tasks to let the subjects participate in a thread, explore the desired features, and perform the required functions in either EzMail or Outlook. The tasks were all part of an integrated scenario and hence were consecutive, i.e. subjects had to complete one task in order to proceed to the next one. To help a subject who failed to complete a task, the answer to each task was provided in the description of the following task.
The main idea of the scenario was to apply for a graduate award. Since all subjects were graduate students, they would understand the context of and be familiar with such a task. The subject played the role of a graduate student, Mary Nelson for a female subject or Bob Smith for a male subject. The scenario was outlined as follows:

You, as a graduate student in engineering science department receive an email from Raj Pabla, the graduate secretary, about ‘Graduate awards and scholarships’ and how to apply for them in December. The application deadline is January 20th. Among the documents needed for the application are your transcripts and a reference letter from your supervisor. You forward the message to John Dill, your supervisor, to ask him for the reference letter and at the same time put an annotation on Raj’s message to remind yourself that you have done this.

Back to school in January, you would like to apply for an award. Since the deadline is coming, you have to find the original message from Raj. Rereading the message, you find out that there is another short message from Raj with the list of scholarships attached. You have to find that message as well.

You also read the annotation on Raj’s message and find out that you should look for John’s reply to your inquiry about a letter of reference. You are not sure in which folder you have saved his message. You may as well change the annotation to remind yourself to ask for your transcripts from Raj.

You find John’s email in another folder named ‘office’. He has asked if there is a template for the reference letter. You remember that Raj has sent an email regarding reference letters recently. You want to find the message but can’t remember where you have filed it.

Finding Raj’s message in ‘office’ folder, you find out that she has given a hyperlink that contains templates for reference letters. You want to send the link to John, but at the same time, you need to confirm with Raj that the same format is fine for this application as well. You would like to send one reply to both John’s and Raj’s email messages including their original messages, for Raj to read John’s question and approve the template and for John to get the format of the reference letter.
6.1.2 Tasks

Based on this scenario, six tasks were formulated. We also created 11 messages related to the tasks. Three messages were filed in the inbox and the rest in a folder named ‘office’. In performing these tasks, the subject would make use of the new visualizations and functions of EzMail and similar features in Outlook and participate in a thread to apply for graduate awards. The subject was given a brief overview of the scenario.

Task 1 asked the subject to find the message from Raj Pabla about ‘Graduate awards and scholarships’ and how to apply for them. The sender, the subject, the folder, an overview of the content, and approximate date was given to the subject. The questions for Task 1 required the subject to find additional properties of the message like being part of a thread and being annotated. EzMail users could benefit from thread grouping in the list view and thread visualizations in the greeked and node views to find that the message was part of a thread. In Outlook, the user could ‘view messages by conversation’. If the subject could find correctly that this message was part of a thread then it would be easy to see that the message was the root message of the thread. The third question asked the subject if the message had been annotated. We previously annotated the message of Task 1.

Task 2 was to open and read the annotation on the message and to determine its priority. Since Outlook does not provide the capability of annotating a message, we added the annotation to the end of the message by editing the message text. To edit a message in Outlook, the user must double click on the message to open it in another window and then select ‘Edit Message’ from Edit menu. EzMail has visual cues in the greeked and node views to represent an annotation. Task 2 also required the subject to edit the annotation.

The goal of Task 3 was to use the size of the message as a property and hence to make use of the greeked view to find the message. The subject was required to find a short email from Raj with the list of scholarships attached. The message was part of a thread in the same folder and from the same sender as of first message. Looking for attachments of the message was also included in this task.

Task 4 asked the subject to find another message in the thread. The point was to access messages of a thread by having one of them (e.g. message of Task 1). This message was from a different sender (John Dill), in a different folder (office), but with the same topic as Raj’s message. EzMail’s thread view can easily help view all messages in the thread from all folders. In Outlook, the user can view all messages related to a given message by right clicking the message.
and selecting 'find all related messages'. However, this is a complex search and the user must select different options to complete the search. To assure that the subject had found the right message, we asked about its content and required the subject to identify the folder in which the message had been filed.

The next task (Task 5) was finding messages from a sender by using a message from this correspondent. This was the message from Raj about the format of a reference letter and was saved in the 'office' folder. It had the same sender as the message from Task 1 but a different topic, and was in a different folder. In EzMail, the sender view can help locate the messages from a sender. Outlook has a similar feature to 'find all messages from the sender' by right clicking on a message. But still the user has to select the folders in which to search.

The last task (Task 6) was trying to send a reply to two messages simultaneously. We wanted the subject to send the template for the reference letter to John and at the same time to Raj, asking if the same format was acceptable for the scholarship application. The subject was asked to send one reply to the messages found in Task 4 and Task 5, so that Raj could read John's question about the reference letters and approve the template, and John could get the format. In EzMail, the user can select more than one message in a folder using 'shift' or 'ctrl' and reply to them simultaneously. Outlook does not have such a feature and the user has to copy and paste the content of the messages and email addresses in the reply message.

6.2 Results

6.2.1 Pre-Session Questionnaire

The pre-session questionnaire for this user study was part of the questionnaire we designed for our user survey. We included those questions that asked for general information about the subject's email habits and preferences. This questionnaire was the same for EzMail and Outlook subjects. We will see how the answers may correlate to the results of the user study.

Our subjects were not heavy users of email. All use a web-based email client. Only four use a client-based email client as well. On average, they spend about 1-1.5 hour daily on email and most of them check their email 1-7 times a day. They send 5-10 and receive 15-20 email messages daily. The number of their email in the inbox is from 10 to a few thousands. Some of them do not use folders at all and those who do, have at most 15 folders. Topic, project, and sender are the most popular criteria for selecting folders. The subjects retrieve messages from
folders mostly by visual scanning and sorting; searching is less common. Our subjects are not very familiar with threads. For most of them, we had to explain what a thread is. Even for those few who work with threads, visual scan and sort are still used to find messages in a thread. Our subjects often have to look for messages from a sender, but annotating a message and replying simultaneously to two or more messages are not frequent activities. Some of the subjects were not familiar with the meaning of annotation.

6.2.2 Statistical Analysis

Our experiment involved two sets of small samples in which measurements were taken of two dependent variables (time to perform tasks and number of errors) under two conditions, EzMail and Outlook. The difference between the means of the two samples indicates the difference in performance. We used a t-test for task performance times to estimate the confidence level of the results. The results are based on two-tailed tests and a 10% α-level (90% confidence interval). This level is more appropriate for exploratory studies with a limited number of subjects than the conventional 5% used in confirmatory studies. We assumed a normal distribution for task performance times. For number of errors a U-test was performed.

Table 6-1 displays the mean value, standard deviation, mean difference, the ratio of mean difference to Outlook mean, t, and p values for completion time of each task. It shows that the average time for completing Task 1 (finding a message), Task 2 (annotating a message), Task 4 (finding messages in a thread), Task 6 (replying to >1 message), and all tasks (total time) are significantly different for the two sample groups. In each case, the mean value for EzMail is less than Outlook. The percentage of mean difference indicates the improvement of performance time for EzMail users in comparison to Outlook users. For example, the total time improved by 41% with EzMail over Outlook. Completion time for Task 3 and Task 5 are not significantly different at the level of p=0.05. Figure 6-1 displays these results as a graph. Error bars display the standard error of the mean.
Table 6-1 Task Completion Time

<table>
<thead>
<tr>
<th>Item</th>
<th>Mean (s)</th>
<th>Std. Dev. (s)</th>
<th>Mean Diff. (s)</th>
<th>Percent of Mean Diff.</th>
<th>t</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EzMail</td>
<td>Outlook</td>
<td>EzMail</td>
<td>Outlook</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 1: Find message</td>
<td>103</td>
<td>199</td>
<td>50</td>
<td>63</td>
<td>96</td>
<td>48%</td>
</tr>
<tr>
<td>Task 2: Annotate message</td>
<td>128</td>
<td>299</td>
<td>80</td>
<td>104</td>
<td>172</td>
<td>56%</td>
</tr>
<tr>
<td>Task 3: Find message in thread</td>
<td>134</td>
<td>191</td>
<td>50</td>
<td>80</td>
<td>58</td>
<td>30%</td>
</tr>
<tr>
<td>Task 4: Find message in thread</td>
<td>273</td>
<td>444</td>
<td>132</td>
<td>157</td>
<td>171</td>
<td>39%</td>
</tr>
<tr>
<td>Task 5: Find message from sender</td>
<td>138</td>
<td>166</td>
<td>67</td>
<td>108</td>
<td>29</td>
<td>17%</td>
</tr>
<tr>
<td>Task 6: Reply to &gt;1 message</td>
<td>173</td>
<td>307</td>
<td>74</td>
<td>126</td>
<td>135</td>
<td>44%</td>
</tr>
<tr>
<td>Total time</td>
<td>948</td>
<td>1607</td>
<td>241</td>
<td>276</td>
<td>659</td>
<td>41%</td>
</tr>
</tbody>
</table>

Figure 6-1 Task Completion Time

Table 6-2 displays the mean value, standard deviation, mean difference, the ratio of mean difference to Outlook mean, $U$, and $p$ values for the responses to task-related questions. It shows that the number of errors (no response or incorrect response) for task-related questions is significantly less in EzMail than Outlook ($p=0.093$).
Table 6-2 Task-Related Questions

<table>
<thead>
<tr>
<th>Item</th>
<th>EzMail</th>
<th>Outlook</th>
<th>EzMail</th>
<th>Outlook</th>
<th>Mean Diff.</th>
<th>Percent of Mean Diff.</th>
<th>U</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td># of responses</td>
<td>9.7</td>
<td>9</td>
<td>0.82</td>
<td>1.55</td>
<td>-0.67</td>
<td>7.4%</td>
<td>22</td>
<td>0.589</td>
</tr>
<tr>
<td># of incorrect responses</td>
<td>0</td>
<td>2</td>
<td>0.00</td>
<td>2.10</td>
<td>2</td>
<td>100%</td>
<td>33.5</td>
<td>0.009</td>
</tr>
<tr>
<td># of no responses</td>
<td>0.3</td>
<td>1</td>
<td>0.82</td>
<td>1.55</td>
<td>0.67</td>
<td>67%</td>
<td>22</td>
<td>0.589</td>
</tr>
<tr>
<td># of errors (incorrect + no response)</td>
<td>0.3</td>
<td>3</td>
<td>0.82</td>
<td>1.79</td>
<td>2.67</td>
<td>89%</td>
<td>29</td>
<td>0.093</td>
</tr>
</tbody>
</table>

We also compared the way the subjects performed the tasks for EzMail and Outlook. The results are shown in different forms in Table 6-3, Table 6-4, and Figure 6-2. Table 6-3 has six columns for each group and task listing the number of individuals who completed the task, left it incomplete, performed with error, recovered from error, or met the time limit. The last column displays the number of errors, which is the sum of incomplete and incorrect tasks. We have six tasks for six subjects in each group so altogether we have 36 task-subjects if we consider it as a variable. The last row of the table shows the total number in each column for this variable. For example, 29 out of 36 task-subject were complete for EzMail while this number is 19 out of 36 task-subject for Outlook.

We ran a U-test to evaluate task performance. Table 6-4 displays the mean value, standard deviation, mean difference, the ratio of mean difference to Outlook mean, \( U \), and \( p \) values for task performance. The number of completed tasks in EzMail is significantly more than for Outlook \( (p=0.041) \) and the number of task errors is significantly more in Outlook than EzMail \( (p=0.093) \).

Figure 6-2 displays the last row of Table 6-3 in a graph. It shows the percentage of subjects who performed the tasks with or without success. Error bars display the standard error of the mean. We see that the subjects were more successful performing the tasks with EzMail than Outlook.
### Table 6-3 Task Performance – Raw Data

<table>
<thead>
<tr>
<th>Item</th>
<th>Group</th>
<th>Complete</th>
<th>Incomplete</th>
<th>Incorrect</th>
<th>Recovered Errors</th>
<th>Time Limit</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Find message</td>
<td>EzMail</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Task 2: Annotate message</td>
<td>EzMail</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Task 3: Find message in thread</td>
<td>EzMail</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Task 4: Find message in thread</td>
<td>EzMail</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Task 5: Find message from sender</td>
<td>EzMail</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Task 6: Reply to &gt;1 message</td>
<td>EzMail</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>EzMail</td>
<td>29</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>19</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>

### Table 6-4 Task Performance – Statistical Analysis

<table>
<thead>
<tr>
<th>Item</th>
<th>Mean EzMail</th>
<th>Std. Dev. EzMail</th>
<th>Mean Diff.</th>
<th>Percent of Mean Diff.</th>
<th>U</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td># of complete</td>
<td>4.83</td>
<td>3.17</td>
<td>-1.67</td>
<td>53%</td>
<td>31</td>
<td>0.041</td>
</tr>
<tr>
<td># of incomplete</td>
<td>0.17</td>
<td>1.00</td>
<td>0.83</td>
<td>83%</td>
<td>18.5</td>
<td>0.937</td>
</tr>
<tr>
<td># of incorrect</td>
<td>0.00</td>
<td>0.50</td>
<td>0.50</td>
<td>100%</td>
<td>24</td>
<td>0.394</td>
</tr>
<tr>
<td># of recovered errors</td>
<td>0.67</td>
<td>0.50</td>
<td>-0.17</td>
<td>34%</td>
<td>19.5</td>
<td>0.818</td>
</tr>
<tr>
<td># of time limit</td>
<td>0.33</td>
<td>0.83</td>
<td>0.50</td>
<td>60%</td>
<td>25</td>
<td>0.310</td>
</tr>
<tr>
<td># of errors (incomplete + incorrect)</td>
<td>0.17</td>
<td>1.50</td>
<td>1.33</td>
<td>89%</td>
<td>28.5</td>
<td>0.093</td>
</tr>
</tbody>
</table>

79
For each task, there was a feature in EzMail to help the subject complete the task easier. There were similar features in Outlook for some of the tasks. Table 6-5 lists these features and the percentage of subjects who used them in the test. There is also a column showing how the rest of the subjects accomplished the task. Outlook users mostly made use of search, sort, and visual scanning to find messages. Search in some cases did not return the message the user sought. Apparently Outlook threading features (view by conversation, find all related messages, or find all messages from a sender) were not easily accessible for the users since they did not use them.

In EzMail, the node and list views could be used for the first two tasks. For the third task we had hoped the greeked view would provide an additional visualization to the user for size of the message. However, no subject made use of this view and they completed the task using the list and node views. This may be why the completion times for Task 3 were not significantly different for the two groups. However, the thread visualization helped the user in finding the message in EzMail. For Task 4, most of the subjects made use of the thread view as we expected, which was encouraging for us. Only one subject used the sender view and could not complete the task in the assigned time. Task 5 was finding messages from a sender where the sender view could help and was used by most. However, if there were many messages from a sender, then finding one message even in the sender view was not trivial, especially without a search tool available. This made the difference between completion times for Task 5 not statistically significant for EzMail and Outlook. For Task 6 we told the subjects during training that they...
could select more than one message. Most of the subjects (four) used this feature to reply to two messages. The rest (two) used copy and paste.

Table 6-5 Features for Tasks

<table>
<thead>
<tr>
<th>Item</th>
<th>Group</th>
<th>Feature to be used</th>
<th>Percent used it</th>
<th>Feature the rest used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Find message</td>
<td>EzMail</td>
<td>Node or list view</td>
<td>100%</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>View by conversation</td>
<td>17%</td>
<td>Search-Sort</td>
</tr>
<tr>
<td>Task 2: Annotate message</td>
<td>EzMail</td>
<td>Node or list view</td>
<td>100%</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Task 3: Find message in thread</td>
<td>EzMail</td>
<td>Greeked view</td>
<td>0</td>
<td>List or node view</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Task 4: Find message in thread</td>
<td>EzMail</td>
<td>Thread view</td>
<td>83%</td>
<td>Sender view</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>Find related messages</td>
<td>0</td>
<td>Search-Sort</td>
</tr>
<tr>
<td>Task 5: Find message from sender</td>
<td>EzMail</td>
<td>Sender view</td>
<td>67%</td>
<td>List view</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>Find messages from sender</td>
<td>0</td>
<td>Search-Sort</td>
</tr>
<tr>
<td>Task 6: Reply to &gt;1 message</td>
<td>EzMail</td>
<td>Select 2 messages</td>
<td>67%</td>
<td>Copy &amp; paste</td>
</tr>
<tr>
<td></td>
<td>Outlook</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

6.2.3 Post-Session Questionnaire

The post-session questionnaire provided us with subjective opinions of the participants regarding the two systems. We asked subjects about the difficulty of the activities they performed through the test and what they liked most and least about EzMail or Outlook. The EzMail questionnaire had a few specific questions on the usefulness and ease of use of its different views and functions and whether the users liked the visualizations. These questions were not applicable to Outlook.

The original levels of rating that we used in the questionnaire assigned the best case to 1 and the worst case to 5. We redefine this rating to 1 being the worst and 5 the best rating. Therefore, we consider \(6-(\text{user rating})\) for each case.

Figure 6-3 compares the mean values for rating of ease of task performance in EzMail and Outlook (on a scale of 1: very difficult to 5: very easy). Comparing the mean values from Figure 6-3, subjects rated EzMail easier to use than Outlook for finding messages in a thread, finding messages from a sender, and annotating a message. The only function that had identical ratings for both EzMail and Outlook was replying to more than one message. Actually the two
EzMail users who did not use the specific feature of this tool for selecting two messages and replying to them rated it as difficult. It is possible that had they used this feature they would have found this task easier.

![User Rating Chart]

**Figure 6-3 Ease of Task Performance**

However, a U-test did not show a significant difference in user ratings for finding a message from a sender and replying to more than one message (Table 6-6). Annotating a message is rated significantly better for EzMail than for Outlook ($p=0.002$), and for finding a message in a thread the significance level is $p=0.093$.

**Table 6-6 Ease of Task Performance**

<table>
<thead>
<tr>
<th>Task</th>
<th>Mean EzMail</th>
<th>Mean Outlook</th>
<th>Mean Diff.</th>
<th>$U$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find message in a thread</td>
<td>4.7</td>
<td>3.7</td>
<td>1.0</td>
<td>29</td>
<td>0.093</td>
</tr>
<tr>
<td>Find message from sender</td>
<td>4.3</td>
<td>3.8</td>
<td>0.5</td>
<td>23.5</td>
<td>0.394</td>
</tr>
<tr>
<td>Annotate a message</td>
<td>4.8</td>
<td>2.2</td>
<td>2.6</td>
<td>35.5</td>
<td>0.002</td>
</tr>
<tr>
<td>Reply &gt;1 message</td>
<td>3.3</td>
<td>3.3</td>
<td>0.0</td>
<td>18</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 6-4 and Figure 6-5 show how EzMail users rated its different views and functions for usefulness and ease of use (on a scale of 1: not at all to 5: very much). Again, the ratings are quite encouraging. Generally, they found EzMail's features useful and easy to use. The only view that users found less useful was the greeked view that they did not really use.
Figure 6-4 Usefulness of EzMail Features

Figure 6-5 Ease of Use of EzMail Features

Figure 6-6 shows that EzMail users liked its visualizations for a single message, messages of a thread, and messages in a folder (on a scale of 1: not at all to 5: very much).

Figure 6-6 Ratings of EzMail Visualizations
The last two questions asked the users what they liked most or least about the tool they used. The main positive comment about Outlook, which was common among subjects, was the organization of email messages into folders and subfolders that provided management of a large number of messages. They complained about its complexity that prevented them from using all its features. Most did not find the search tool very useful for finding a message.

All EzMail subjects liked the idea of threading. They found grouping by thread, thread view, and node view helpful in finding messages. One user found the “clean simple layout and design of EzMail” appealing. Another user found EzMail “a good organized offering of different views”. One of the users mentioned its “ease of learning”. Two users commented on the node view that gave the opportunity “of an easy and fast scan of email” and “to view information about a message such as annotation and attachments”. Another participant found “a separate view for messages from a sender quite useful”. Two users liked the idea of a greeked view.

Among missing features in EzMail, subjects identified a search tool and the ability to customize the workspace. Needed interface refinements included a marker on field titles to show the direction of sort, and a reply button to make replying easier. One subject complained that the visual buttons for annotation and attachments in the node view were small and the greeked view was a bit crowded to click on a message. A problem with EzMail was that sorting messages based on a field resulted in loss of selection of the selected message.

6.3 Discussion

Our subjects were graduate and undergraduate students. Based on their responses to the pre-session questionnaire, they mostly use web-based email programs. Since they usually do not have a dedicated computer at work, a web-based email client is generally more appropriate for them. They do not send or receive a large number of messages and most are not very familiar with threads (half said that they do not work with threads). This group of users thus differed from the participants of our user survey who were mostly people from industry with a high volume of email and familiarity with threads. Since EzMail supports threading, it is more helpful for high-volume email users rather than low-volume users.

Another observation was the trade-off between the amount of training and subject performance. Determining the amount of training to give subjects was a difficult task. Following several discussions, we decided simply to give subjects a brief overview of functions available with Outlook and EzMail following which they had a brief practice session (up to 10 minutes).
and could ask questions before undertaking the experimental tasks. This way, we also tested the ease of learning of the features which the user study results suggest was higher for EzMail than Outlook. Not surprisingly, informal comments and observations suggest that performance would have been better with a longer training session.

We observed that working with EzMail helped subjects who were unfamiliar with the idea of threads quickly pick up the idea of threading. Most of the features of EzMail related to tasks were used by users except for the greeked view. Outlook users did not look for any threading feature. This suggests that threading features are not easily accessible for the Outlook users.

Since EzMail was designed as a companion tool to run in conjunction with the user’s email client, it has only a fraction of Outlook’s features. However, subjects assigned to EzMail were not allowed to access any of Outlook’s functions (e.g. search), which made it more challenging for EzMail users to perform the tasks.

Qualitatively, our observations of subjects’ behavior suggested Outlook users found it more difficult to complete the tasks than EzMail users. This suggests that EzMail’s features are more visible and easier to use than those of Outlook. We acknowledge however that Outlook, being a complete personal data management system, including being a complete email client, is more difficult to explore in a short time for a new user.

6.4 Summary

The user study we conducted to measure user satisfaction and performance using EzMail was quite positive. Based on the statistical results, we can state that performance of EzMail users is significantly better than Outlook users. EzMail users, despite its limited set of features, completed the tasks faster and with fewer errors than Outlook users. The average task performance time and number of errors with tasks and task-related questions were significantly less for EzMail users than Outlook users. According to the statistical results, EzMail is significantly more helpful in finding messages in a thread, annotating a message, and replying simultaneously to more than one message. Based on the subjective results of the post-session questionnaire the users’ overall impression of EzMail was positive. They found the specific features and views of EzMail useful and easy to use. They also liked different visualizations of EzMail for single messages, messages of a thread, and messages in a folder.
Overall, the user study results suggest that the threading features, visualizations, and functions of EzMail help users work more effectively with their email messages and threads when compared with similar tasks using Microsoft Outlook.
CHAPTER 7

CONCLUSION

7.1 Summary

In this thesis, we developed and tested EzMail, an email companion tool with new visualizations and functions. Subjects using EzMail performed better than those using Outlook in dealing with email messages and threads.

We started with an analysis of previous email research to study email patterns of use, reported problems, and proposed solutions. Email, being used for a variety of functions, is inevitably overloaded. Managing email messages, conversations, tasks, and documents needs a great deal of effort from the email user. Email clients need much improvement to help users more effectively handle their email. Threading is one of the proposed methods that can help with management of information, conversations, and tasks in email. Based on this background, we developed design ideas to implement threading features in an email visualization tool. Our approach was to apply information visualization methods to visualize different aspects of a mailbox and help user gain more insight into the information.

At the same time, we designed a questionnaire and conducted an email user survey to further examine email users' habits, preferences, and difficulties. Based on the results, we proposed a classification of email users based on their foldering habits. We categorized our subjects as archivers who use folders for archiving old messages and organizers who tend to organize their active messages in folders. The user survey also helped us better understand the
current state of the art in threading. We also found that viewing by sender, date, or other attributes of messages is helpful. Altogether, the results of this survey confirmed that our preliminary ideas were in the right direction.

So we designed EzMail with multiple views to present the content, context, and structure of email messages. We developed novel visualizations for individual messages, message properties, and messages in a thread. Viewing by thread was implemented to help users get an improved overview of conversations and their messages, status, and participants. The idea of using greeked text to represent a high-level overview of email messages is a new idea. Message annotation incorporated in EzMail may serve as a reminder or to-do and help with task management. Replying to several messages support convergent conversations that may lead to creation of new ideas in a discussion.

A follow up user study provided a comparative evaluation of EzMail with respect to Microsoft Outlook. It showed significant improvement in user performance for finding messages in a thread, annotating a message, and replying simultaneously to several messages using EzMail over Outlook.

7.2 Research Contribution

We developed EzMail as a multi-view interface for email messages to help with communication management, information management, and task management. It provides contextual information and conversational history for email messages by presenting them as part of their conversational threads. It presents a high level overview of the linear incoming stream of messages as well as an integration of traditional file organization of messages with a graphical hierarchy that reveals the linkages and relationships among messages. Message annotation and replying to several messages were incorporated to help with managing conversations and tasks.

A user survey led to recognition of email management behavior of users. An evaluative user study resulted in validation of improvement in user performance and satisfaction using EzMail. We put considerable effort and care into the design of an appropriate scenario with representative tasks. The result however was a set of tasks and associated questions which engaged the student subjects and helped improve the validity and relevance of the experimental results. Overall, this was an an important part of our research contribution.

In summary, the contribution of this thesis is the design, development, and evaluation of new features and functions in an email prototype to assist with email management.
7.3 Future Work

Next steps would include both major steps to provide more robust versions of EzMail and minor improvements to the interface and features of EzMail.

An ambitious task would be revealing the patterns of social connections among email users and messages. To accomplish this feature, content analysis methods should be used to explore the content of email messages.

More reliable threading algorithms must be developed to employ as many of the thread-related headers of messages in order to recognize threads more accurately. Content analysis and text matching methods would be helpful in using ‘quoted messages’ to find messages of a thread. Visualization of detailed relationships among messages in a thread should also be supported in EzMail.

An alternative method to test EzMail views is to present the views to the user one by one and compare the performance results to see which view is more helpful. The features and visualizations proven useful may then be incorporated as add-ins to an email client to provide more convenient methods for email management.

Since the subjects for the user study were relative novices for both EzMail and Outlook, the study results emphasize the learnability of the two interfaces. An interesting extension would be to provide significantly longer training times, and to study and compare the results for experienced users.

Adding a toolbar to present the main functions, implementing a ‘search’ tool, and providing ‘help’ facilitates the usage of EzMail. Color coding greeked messages can be used to highlight specific message attributes, and to identify new part of a message from the history of previous messages in the message content. The idea of greeked attachments similar to greeked messages can be helpful in visualizing size of attachments and thus cleaning up the mailbox. Another improvement would be visualization of messages in the node view with more informative structures whose shapes or sizes represent message properties.

Our initial investigation confirms several other investigators’ findings that email is an increasing burden that none of us can escape. Better tools are needed. We have developed one approach to this, and have demonstrated its validity. Clearly however, this is an area where much more needs to be done to help reduce the burden of increasing volumes of, and increasing set of tasks represented by, email.
APPENDIX A

EzMail User’s Guide

General

1. For EzMail to access email, your default email client must be Microsoft Outlook. To set the default email client, go to Control Panel, click on Internet Options, click Programs tab and set the default email program to Outlook.

2. At EzMail startup, a dialog pops up asking to choose the profile name. Select your default profile and click Ok.

3. First your new (unread) messages, if there are any, are retrieved from the Inbox.

4. A security message pops up asking for your permission to access Outlook. Check the box (Allow access ...) and click yes to get your messages. You can allow access for up to ten
minutes. This message will show up during EzMail work and you have to renew the permission.

![Security Message]

**Main View**

5. There are five interconnected views in EzMail main window to display the messages. You can resize the views by holding the left mouse button down on the common edges and moving the mouse.

6. Right mouse click on each message in the list, greeked, or node view pops up a menu with four items (thread, sender, annotation, and reply) to open the related views in new windows. They are also accessible from the main menu.

7. The main menu has three commands: File, View, and Tools.

8. The ‘Log On’ and ‘Log Off’ under File are for entering and exiting EzMail. Logon happens automatically when you open the application. The ‘Log On’ command is just usable if an error occurs in logon process.

9. To close the program click ‘Logoff’ from ‘File’ menu.

10. From ‘Thread view’ and ‘Sender view’ under View you can access the related views for a selected message. You should have selected a message before clicking these items.

11. There are two commands under Tools. ‘Receive’ is used to receive new messages from the Inbox. ‘Reply’ opens the reply view for the selected message or messages.

**Folder View**

12. The folder view displays your personal email folders.
13. Clicking on a folder name lists its messages in the list view.

**List View**

14. The *list view* lists messages of a selected folder sorted by date and grouped by thread.

15. Unread messages are shown with bold font.

16. The icons represent ‘read’ or ‘unread’ for singleton messages and a ‘reply’ or ‘root’ message in a thread.

17. When an unread message is selected and read, the visualizations are updated.

18. The status bar at the bottom of the view shows the number of messages in the selected folder.

19. Clicking on a message updates other views. The same message is highlighted in the greeked and node views and the contents are shown in the message view.

20. You can select more than one message by holding shift or ctrl and clicking the left mouse button on the message. The selected messages are highlighted and their contents are shown in cascaded text boxes in message view.

21. Clicking on ‘Date’ and ‘From’ column headers will sort the messages by that field ascending or descending. Clicking the subject field will sort messages by date while grouped by thread. The greeked and node views are updated at the same time.

**Message View**

22. The *message view* shows the contents of a message.

23. The header of the message displays the recipients and attachments of the message.

**Greeked View**

24. The *greeked view* displays a reduced-resolution image of messages’ content to give an overview of their structure.

25. Each message is encoded as a line whose length represents the length of the message.

26. A thicker line shows an unread message.

27. A mouse over the line will show the message sender, subject, and date.

28. Clicking on a message highlights the same message in other views.
29. An annotation is represented by a small vertical line at the upper right corner of the message line. White, yellow, or red shows the normal, high, or very high priority of the annotation. Double clicking on the cue will open the annotation view.

30. Each attachment is represented by a small vertical blue line at the lower right corner of the message line.

**Node View**

31. Each frame node shows an individual message or a thread root message. The title of the frame node is the subject of the message.

32. Replies within a frame are shown with box nodes.

33. Bold text of frame nodes or un-pressed box nodes show unread messages.

34. Placing the mouse pointer on each node pops up more information about the message.

35. Clicking on a message highlights the same message in other views.

36. Each small blue square at the lower right corner of a node shows an attachment. Clicking on the cue displays a list of attachments. Placing the mouse over the cue will show the name of that attachment.

37. A white, yellow, or red small square at the upper right corner of the node indicates that the message has an annotation with different priorities (normal, high, very high respectively). Double clicking on the box will display the annotation view where you can edit the text. You can also right click on an annotation icon and edit or delete it.

**Thread View**

38. This view collects and displays messages of a thread from all folders.

39. The caption of the view is the topic of the selected message.

40. There are list and node views in this window.

41. The list view displays messages sorted by date and grouped by thread.

42. Clicking on the column headers of the list view will sort the messages by that field ascending or descending. The ‘Sort Thread’ button will revive the original view sorted by date and grouped by thread.

43. Each frame node in the node view represents a folder. Clicking on the frame node highlights all messages from that folder.
44. Each box node in the node view represents a message. The attachments and annotation cues are also present similar to the main view.

**Sender view**

45. This view collects and displays messages to or from a correspondent from all folders. The messages to the correspondent are shown with gray color.

46. The caption of the view is the sender of the selected message.

47. There are list and node views in this window. Their properties are identical to the *thread view*.

**Annotation**

48. With this view you can add annotation to any message.

49. Type the text and click ‘Save’.

50. Clicking ‘Delete’ will remove the text. Clicking ‘Cancel’ will not save any changes.

51. You can change the priority of the annotation to high or very high. The default is normal.

52. In any case you should save changes.

53. The node and greeked views are updated to show the added or removed annotation.

**Reply**

54. You can reply to a selected message in this view.

55. A window may show up to resolve the recipient address if there is no address or more than one address for the person in the address book.

56. The recipient and subject are shown at the top.

57. Clicking ‘Send’ will save the message in the Outbox folder of Outlook. Anytime a send/receive is performed in Outlook the message will be sent.

58. You can also reply to more than one message by selecting several messages in the list view.

59. All the recipients and subjects will be shown in the reply view.
APPENDIX B

EMAIL USER SURVEY

Questionnaire

General

1. What kind of organization/department do you work for/study at?
2. What is your position and profession?
3. What programs do you use to read email?
4. Do you check your email at home, office/school, or both?
5. How often do you check your email?
   - < 1 time / day
   - 1-7 times / day
   - about 1 time / hour
   - >1 time / hour
6. Do you get automatic notification of new mail? Visual, audible, or both?
7. How much time do you spend daily to deal with your email? (please mark on the following axis.)

   0  ½  1  2  3  5  hour
8. How many messages do you send per day? (please mark on the following axis.)

9. How many messages do you receive per day? (please mark on the following axis.)

Foldering

10. Do you use folders to organize your messages? If yes, how does foldering help you in managing your email?

11. What kind of messages do you keep in your inbox? (please mark all that apply.)
   - incomplete tasks
   - unfiled information
   - ongoing conversation
   - others

12. How many messages do you typically have in your inbox? (please mark on the following axis.)

13. How many folders (including subfolders) do you have (other than inbox, sent, and trash)?

14. How deep is your folder structure (the maximum number of levels)?

15. What is the maximum number of messages in a folder?

16. What is the total number of messages in all folders (not including inbox, sent, and trash)? (please mark on the following axis.)

17. Based on what criteria do you choose your folders? (please mark all that apply.)
   - sender
18. On a scale of 1(rarely) to 5(quite often), how often do you feel the need to file a message in several folders (one copy and a bunch of links)?

19. How do you retrieve messages from folders? (please mark all that apply.)
   - search tool
   - sort messages
   - visual scanning
   - others

20. On a scale of 1(easy) to 5(difficult), how difficult is it to retrieve messages from folders? Why is it difficult? What capabilities are needed to simplify the task?

21. Do you use automatic filtering to categorize your messages? If yes, how does it help you in managing your email (other than spam filtering)?

22. What other problems do you encounter managing folders?

23. What other capabilities are needed to improve the foldering feature?

**Threading**

24. How do you track a thread or conversation? (please mark all that apply.)
   - search tool
   - sort messages
   - visual scanning
   - grouping
   - others

25. On a scale of 1(easy) to 5(difficult), how difficult is it to track a thread or conversation? Why is it difficult? What capabilities are needed to simplify the task?

26. When you reply to a message in a conversation, on a scale of 1(rarely) to 5(quite often) or 6(always by default), how often do you:
   - save your sent messages?
• include the original message in your reply?

27. On a scale of 1(rarely) to 5(quite often) or 6(always by default), how often do you save your sent messages? How do you make use of the sent items?

28. When you want to create a new mail, on a scale of 1(rarely) to 5(quite often), how often do you reply to a message instead, to avoid typing the receiver's email address?

29. When your reply to a message is not relevant to that message, on a scale of 1(rarely) to 5(quite often), how often do you:
   • change the subject of your reply?
   • delete the original message in your reply?

30. On a scale of 1(rarely) to 5(quite often), how often do you feel the need to send one reply to more than one message (gather 2 or more threads together)?

31. What other problems do you encounter managing threads?

32. What other capabilities are needed to improve the threading feature?

Attachments

33. On a scale of 1(easy) to 5(difficult), how difficult is it to manage the attachments?

34. On a scale of 1(rarely) to 5(quite often), how often do you:
   • save the attachments into file folders?
   • leave the attachments attached to the mail message in email folders?

35. On a scale of 1(rarely) to 5(quite often), how often do you feel the need to forward attachments separately (without the message)?

36. What other problems do you encounter managing attachments?

37. What other capabilities are needed to simplify attachment management?

Email Management

38. What other problems do you encounter using your email client?

39. What other functions would you like your email client be able to do (simple calendar, meeting scheduler, etc.)?
# User Survey Results

<table>
<thead>
<tr>
<th>Fields</th>
<th>Users</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email checking frequency (per hour)</td>
<td>Min</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.13</td>
<td>0.63</td>
<td>1</td>
<td>1</td>
<td>0.13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.63</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.97</td>
</tr>
<tr>
<td>Email usage time (h)</td>
<td>Min</td>
<td>0.5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.5</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2.32</td>
</tr>
<tr>
<td># of sent email</td>
<td>Min</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>10</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>40</td>
<td>10</td>
<td>30</td>
<td>15</td>
<td>25.21</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>3</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>30</td>
<td>15</td>
<td>10</td>
<td>50</td>
<td>50</td>
<td>30</td>
<td>40</td>
<td>15</td>
<td>25.21</td>
</tr>
<tr>
<td># of received email</td>
<td>Min</td>
<td>10</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>40</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>30</td>
<td>60</td>
<td>200</td>
<td>50.71</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>15</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>40</td>
<td>60</td>
<td>300</td>
<td>66.07</td>
</tr>
<tr>
<td><strong>Folders</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of folders</td>
<td>Min</td>
<td>25</td>
<td>1</td>
<td>73</td>
<td>38</td>
<td>50</td>
<td>4</td>
<td>13</td>
<td>42</td>
<td>180</td>
<td>0</td>
<td>12</td>
<td>14</td>
<td>500</td>
<td>0</td>
<td>73.23</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>25</td>
<td>2</td>
<td>73</td>
<td>38</td>
<td>230</td>
<td>4</td>
<td>13</td>
<td>42</td>
<td>180</td>
<td>0</td>
<td>12</td>
<td>14</td>
<td>500</td>
<td>750</td>
<td>134.50</td>
</tr>
<tr>
<td>Depth of folder structure</td>
<td>Min</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>2.43</td>
</tr>
<tr>
<td>Max # of email in a folder</td>
<td>Min</td>
<td>10</td>
<td>10</td>
<td>300</td>
<td>73</td>
<td>500</td>
<td>20</td>
<td>1000</td>
<td>94</td>
<td>3</td>
<td>0</td>
<td>200</td>
<td>300</td>
<td>500</td>
<td>800</td>
<td>272.14</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>110</td>
<td>15</td>
<td>300</td>
<td>73</td>
<td>500</td>
<td>20</td>
<td>1000</td>
<td>94</td>
<td>4000</td>
<td>0</td>
<td>200</td>
<td>1200</td>
<td>1000</td>
<td>1500</td>
<td>715.14</td>
</tr>
<tr>
<td># of email in inbox</td>
<td>Min</td>
<td>350</td>
<td>150</td>
<td>30</td>
<td>10</td>
<td>5</td>
<td>1000</td>
<td>750</td>
<td>50</td>
<td>100</td>
<td>700</td>
<td>500</td>
<td>3900</td>
<td>100</td>
<td>25</td>
<td>547.86</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>350</td>
<td>800</td>
<td>100</td>
<td>15</td>
<td>7</td>
<td>2000</td>
<td>750</td>
<td>100</td>
<td>500</td>
<td>1000</td>
<td>1000</td>
<td>3900</td>
<td>100</td>
<td>3900</td>
<td>100</td>
</tr>
<tr>
<td>Total # of email in folders</td>
<td></td>
<td>1000</td>
<td>30</td>
<td>4000</td>
<td>600</td>
<td>5000</td>
<td>100</td>
<td>5000</td>
<td>1000</td>
<td>10000</td>
<td>1500</td>
<td>500</td>
<td>10000</td>
<td>8000</td>
<td>22000</td>
<td>4,909.29</td>
</tr>
<tr>
<td>Fields</td>
<td>Users</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>Mean</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------------------------</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-------</td>
</tr>
<tr>
<td>Total # of email in folders + inbox</td>
<td></td>
<td>1350</td>
<td>830</td>
<td>4100</td>
<td>615</td>
<td>5000</td>
<td>2000</td>
<td>5750</td>
<td>1100</td>
<td>10000</td>
<td>2500</td>
<td>1500</td>
<td>14000</td>
<td>8100</td>
<td>22200</td>
<td>5,646.07</td>
</tr>
<tr>
<td>Percentage of email in inbox</td>
<td></td>
<td>25.93%</td>
<td>96.39%</td>
<td>2.44%</td>
<td>2.44%</td>
<td>0.14%</td>
<td>100%</td>
<td>13.04%</td>
<td>9.09%</td>
<td>5.00%</td>
<td>40.00%</td>
<td>66.67%</td>
<td>27.86%</td>
<td>1.23%</td>
<td>0.90%</td>
<td>13.69%</td>
</tr>
<tr>
<td>Need to file in &gt; 1 folder</td>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>-</td>
<td>1</td>
<td>5</td>
<td>1.82</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difficulty of retrieving from folders</td>
<td>Min</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2.23</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2.46</td>
<td></td>
</tr>
<tr>
<td>Threading</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difficulty of threading</td>
<td></td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1.91</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save sent messages?</td>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5.79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Include original message?</td>
<td></td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5.07</td>
<td></td>
</tr>
<tr>
<td>Reply a message?</td>
<td></td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3.29</td>
</tr>
<tr>
<td>Change subject?</td>
<td></td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>3.86</td>
</tr>
<tr>
<td>Delete original message?</td>
<td></td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>3.79</td>
</tr>
<tr>
<td>Need to reply &gt; 1 message</td>
<td></td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1.93</td>
</tr>
<tr>
<td>Attachments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difficulty of management</td>
<td></td>
<td>5</td>
<td>2</td>
<td>2.5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2.32</td>
</tr>
<tr>
<td>Save to folders?</td>
<td></td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3.21</td>
</tr>
<tr>
<td>Leave with message?</td>
<td></td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3.43</td>
</tr>
<tr>
<td>Need to forward attachments?</td>
<td></td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1.83</td>
</tr>
</tbody>
</table>
APPENDIX C

EzMail User Study

Pre-Session Questionnaire

1. What programs do you use to read email?

2. How often do you check your email?
   - < 1 time / day
   - 1-7 times / day
   - about 1 time / hour
   - >1 time / hour

3. How much time do you spend daily to deal with your email? (please mark on the following axis)

   \[
   \begin{array}{cccccc}
   0 & \frac{1}{2} & 1 & 2 & 3 & 5 \text{ hour}
   \end{array}
   \]

4. How many messages do you send per day? (please mark on the following axis)

   \[
   \begin{array}{cccccc}
   0 & 10 & 30 & 50 \text{ messages}
   \end{array}
   \]
5. How many messages do you receive per day? (please mark on the following axis)

0 10 30 50 messages

6. How many messages do you typically have in your inbox? (please mark on the following axis)

0 10 50 100 500 1000 messages

7. Do you use folders to organize your messages? If yes, what is the approximate number of your folders (including subfolders and excluding inbox, sent, and trash)?

8. Based on what criteria do you choose your folders? (please mark all that apply)
   - sender
   - topic
   - organization
   - project
   - date
   - others (please name)

9. On a scale of 1(rarely) to 5(quite often), how often do you use each of these methods to retrieve messages from folders?
   - search tool
   - sort messages
   - visual scanning
   - others (please name)

10. On a scale of 1(rarely) to 5(quite often), how often do you perform any of these activities?
    - Finding messages in a thread
    - Finding messages from a sender
    - Annotating a message
    - Replying simultaneously to 2 or more messages
11. Do you work with threads? If yes, how do you track a thread or conversation? (please mark all that apply)
   • search tool
   • sort messages
   • visual scanning
   • grouping
   • others (please name)

User Study Tasks

Scenario

Pretend that you are Mary Nelson, a graduate student in the Engineering department, and you want to apply for a graduate student award. You will be working with your email messages and folders to send, receive, file, and retrieve email from your mailbox. You will specifically be working with an email thread.

Task 1 – Finding a Message

Raj Pabla, the graduate secretary, sent a message to you as a graduate student some time in December about “Graduate awards and scholarships" and how to apply for them. Find the message in the inbox.

Task 1 – Questions

1. Is this message part of a thread?
   - Yes
   - No

2. If yes, does this message initiate the thread or is a reply?
   - Initiates the thread
   - Is a reply

3. Is the message annotated?
   - Yes
   - No
Task 2a – Annotating a Message

The visual cues show that there is an annotation on the message you just opened. Open the annotation.

Task 2a – Questions

4. What is the annotation about?
   - Application deadline
   - Transcripts
   - Reference letter

5. Is this a high or normal priority annotation?
   - High
   - Normal

6. Which view helped you find the annotation?
   - List view
   - Greeked view
   - Node view
   - Thread view
   - Sender view

Task 2b – Annotating a Message

Edit the annotation you just opened to remind yourself to email Raj and ask for your transcripts.

Task 3a – Finding Messages of a Thread

Reread the original message from Raj. Look for information in that message about where to find the list of scholarships.

Task 3a – Questions

7. Where can you find the list of scholarships?
   - On ENSC web page
   - In another email
   - In the office
Task 3b – Finding Messages of a Thread

Raj subsequently sent a short email with the list of scholarships attached. Find the message.

Task 3b – Questions

8. How many attachments does the message have?
   - None
   - One
   - Two
   - Five

9. Which view helped you the most in finding the message?
   - List view
   - Greeked view
   - Node view
   - Thread view
   - Sender view

Task 4 – Finding messages of a Thread

The annotation reminded you that you had asked John Dill, your supervisor, for a reference letter. He must have replied to you. Find his message (it might be in another folder rather than the inbox).

Task 4 – Questions

10. What has John asked for?
    - A template for reference letter
    - Your transcripts
    - The list of scholarships

11. Which view helped you the most in finding the message?
    - List view
    - Greeked view
    - Node view
    - Thread view
    - Sender view
12. In which folder did you find the message?
   - Inbox
   - John
   - Office

Task 5 – Finding Messages of a Sender

John has asked for a template of the reference letter. You remember that Raj sent an email regarding "reference letters" recently (not related to this thread). Find her message.

Task 5 – Questions

13. Which view helped you the most in finding the message?
   - List view
   - Greeked view
   - Node view
   - Thread view
   - Sender view

14. In which folder did you find the message?
   - Inbox
   - John
   - Office

Task 6 – Replying to More than One Message

You found John's and Raj's messages in the 'office' folder. Now you want to send the template for the reference letter to John. At the same time you need to ask Raj if the same format is ok for the scholarship application. Send one reply to both messages, so that Raj can read John's question and approve the template, and John can get the format.

EzMail Post-Session Questionnaire

1. On a scale of 1(very easy) to 5(very difficult), how easy was to accomplish the following tasks?
   - Finding messages in a thread
   - Finding messages from a sender
Annotating a message
- Replying simultaneously to 2 or more messages

2. On a scale of 1(very useful) to 5(not at all useful), how do you rate each of these features of EzMail?
- Thread view
- Sender view
- Greeked view
- Node view
- Annotation view
- Replying simultaneously to 2 or more messages

3. On a scale of 1(very easy) to 5(very difficult), how easy is the usage of each of these features of EzMail?
- Thread view
- Sender view
- Greeked view
- Node view
- Annotation view
- Replying simultaneously to 2 or more messages

4. On a scale of 1(very much) to 5(very little), how did you like the following features of EzMail?
- Visual representation of a single message
- Visual representation of messages of a thread
- Visual representation of messages of a folder

5. What did you like most about EzMail?

6. What did you like least about EzMail?

Outlook Post-Session Questionnaire

1. On a scale of 1(very easy) to 5(very difficult), how easy was to accomplish the following tasks?
• Finding messages in a thread
• Finding messages from a sender
• Annotating a message
• Replying simultaneously to 2 or more messages

2. What did you like most about Outlook?

3. What did you like least about Outlook?
LIST OF REFERENCES


