PERSONAL DIGITAL LIBRARIES OF EMAIL

Benjamin M. Gross

University of Illinois Urbana-Champaign
Graduate School of Library and Information Science
University of California Berkeley
School of Information Management and Systems
bgross@uiuc.edu

In this paper, I describe the results of user interviews that reveal problems in current email systems, including role conflict, high cognitive overhead associated with organization and retrieval, inability to navigate conversations and difficulties in addressing messages. I also describe the design of a prototype system with an underlying message store for the email collection. Services layered on top of the message store address many problems described by users through improved support for identity and role management, authority control and query interfaces.

Many of the solutions described in this paper to improve management of personal email collections can also be applied to other digital library collections.

1. Introduction

1.1 Motivation

Collections of personal information are continually growing in size and importance. Electronic mail remains the dominant application for the Internet and is the most ubiquitous type of personal collection. According to the Messaging Online "Year-End 2000 Mailbox Report" the number of electronic mailboxes rose 67 percent from the 533 million in use at the end of 1999. Globally, the number of electronic mailboxes has grown to more than 891 million.

Over the last decade, the primary improvements to email have been in the integration of groupware functionality, rendering of multimedia content, remote access and security. Further research is needed in the areas of navigation, organization and retrieval within electronic mail collections.

In section 2., I describe the results of user interviews that reveal problems in current email systems, including role conflict, high cognitive overhead associated with organization and retrieval, inability to navigate conversations and difficulties in addressing messages. In section 3., I describe a prototype system that addresses many of these problems through improvements in the mes-

sage store, support for identity and roles, authority control and novel query interfaces. Finally, I discuss my conclusions, future work and related work.

2. Interviews

2.1 Methodology

I interviewed twelve users, who were an equal number of novice and expert computer users. Five users were male and seven users were female. All users had at least five years of experience with email. Their education ranged from high school diploma to Ph.D. candidate. The majority of users, seven, are employed in the information technology sector, while five users are in non-technical fields.

2.2 Role

People maintain numerous roles in everyday life including: spouse, employee, student, employer, member of organizations, etc. I found that most users maintain multiple email addresses in order to "act" in multiple "roles." For example, in my survey, the number of user email addresses ranged from at least two to dozens. Most users maintain separate email addresses for work and personal communications. The only exceptions are two users who did not use a computer or email in their work environment. Additional roles with separate addresses included organizational affiliations, multiple work roles, online shopping identities and pseudonymous identities. Users maintained multiple email addresses for other reasons including: status or prestige, Web accessible accounts for travel, to receive junkmail and to maintain a permanent forwarding address.

Even though many email applications allow users to select from a number of accounts to send or receive mail, users expressed difficulty in managing multiple email addresses. It is very important to many users to keep their roles and related email separate. For example, users consistently report being embarrassed by mailing a professional contact with a personal address. Users who experience a single case of role conflict altered their email usage to prevent the situation from happening again. For example, one user now maintains a separate role for her Ebay usage to separate her shopping role from her work role. She accesses her Ebay email through a Web interface and her work email through Netscape Communicator. This enforces a visual separation of the two roles so that it is difficult to make an error.

The most typical coping mechanism users have is to forward multiple addresses to a smaller number of addresses. Despite the total number of ad-

dresses, no user surveyed regularly checks more than three accounts. For example, the user with the largest number of addresses forwards all of his email to a single address. Users typically have more addresses that receive email than addresses from which they send email, because it is cognitively and technically easier to have more receiving addresses than sending addresses.

2.3 Organization, retrieval and navigation

Users manage their collections by creating categories and filing messages into them, moving messages from one category to another, duplicating messages and deleting messages. Most users categorize their email a small amount and sort or search a large amount. It is common for users to leave the bulk of their email in their inbox folder. [4] [16]

Nearly all classification mechanisms require users to place messages into fixed categories. In most systems, a message cannot exist in more than one category unless it is duplicated. This creates a burden on the user to choose the "correct" category to file messages under and to remember the category later in order to retrieve the message.

Most users categorize email, at least in part, by the sender of the message (e.g. a folder for John Smith). [3] [4] The burden is on the user to either file all messages from an individual into a single category or to remember the name variants or email addresses in order to search for that individual later. Recategorization is time consuming because users must move each message to the new category. Often old categories are never fully removed after recategorization leading to "category drift." [1] [5] [14]

Users typically locate messages through sorting by name or date and then browsing to find the desired item. Users report that sorting columns is faster and easier than searching in most cases. A small number of users rely on the built in search function to locate messages.

Users want to search for messages by a person's name, not their email address. One difficulty is that there is no way to reference an individual consistently over time, as their email address and name may change. Email addresses change due to job changes, ISP changes and a host of other reasons. A 1998 International Data Corporation report estimated that twenty to thirty percent of all electronic mail addresses in the U.S. change annually. Searching for names is further complicated because people have additional email addresses for different roles.

Using sorting alone to locate message by an individual's name is problematic as name forms are not standardized and may have several variations. For example, when first and last names are inverted this causes blocks of messages from an individual to be separated in a sorted list. Nicknames, i.e. Bob and Will, have the same effect. The use of initials in names is relatively com-

mon for first and middle names. Names may change due to marriage or divorce. Finally, the name field may be missing entirely and just the email address may appear, or in some cases the email address is duplicated in the name field.

2.4 Navigating conversations

When we communicate with individuals, our interactions may be brief, where the conversation consists of only one message in each direction, or it may be a sustained interaction lasting for years. Because sent email is typically saved in a separate folder, a message and its response are hard to display together in most email clients. When reconstructing conversations, users typically must go back and forth between their sent mailbox, inbox and other folders in order to correlate messages. One user in my interview copies all of his sent email messages into his Eudora inbox so that he can see the parent and child messages together. Other users email themselves a copy of each message they send to achieve a similar effect. Only one user in the study never referred to his sent mail because he did not save it.

Conversations may include many responses and responses to these responses as well as multiple participants. An original message and its responses constitute a "thread" in email. Threads occur both across messages and within messages. Threading across messages is defined by messages having the same subject and headers that link the messages together. A single message may contain previous messages which have been quoted is also a thread. In my study only the most technically sophisticated users selected and relied on email applications that display threaded conversations across messages.

Email applications provide widely varying support for viewing and navigating threads. There are neither definitive standards for specifying thread data nor for quoting previous messages, so each vendor has their own implementation. Some email applications display threads across messages, which preserves the branching hierarchy of the parent-child relationships. Other applications such as Outlook, Outlook Express and Pine group messages together by subject which provides a rough simulation of threading across messages.

2.5 Addressing

Most modern email applications have a mechanism to store and retrieve email addresses. Users rely on a number of techniques to address messages, including address books, aliases, typing addresses in by hand, relying on the autocomplete feature and replying to previous messages.

Most users placed only their most frequently used addresses in their address book. Once an address is entered in the address book, the recipients' name may "auto complete," or expand after the first few characters are typed. Users

rely heavily on the auto-complete feature. Occasionally, auto-complete will choose an address that the user did not expect. Users report difficulty in using this feature to send email to recipients with more than one email address.

In some email clients, such as Outlook and Outlook Express, the recipients name is displayed without the email address, causing additional confusion. If the address book is automatically rather than manually populated with entries, there is a higher likelihood that the user will find unexpected auto-complete matches. Many users report confusion about how entries got into their address book. This confusion is due to the fact that in many configurations Outlook and Outlook express automatically populates the address book with the sender of any message to which the user replies.

There is little reliance on the address book aside from auto-completion. If an address does not auto-complete for a user, it is common to simply type the address by hand. Many users report that replying to an old message is faster than composing a new message and addressing it. Users will often reply to an old message to compose a new message by changing the subject and deleting the body of the message. One user keeps a separate email folder to store messages that contained contact information. She said that filing the message into this folder and retrieving it is faster and easier than entering information into the address book and searching for it.

Overall, most users made little use of nicknames or aliases. Some users enter aliases for most frequently used addresses since they are shorter to type. Others use separate aliases for the same person to distinguish between multiple email addresses for that person. Users report that these systems typically only work correctly in their workplace and not on a home or remote computer. A number of users report using Google as a way to locate addresses.

There is a temporal and geographic component to addressing. Users send email to different addressees depending on the time of day and location of the recipient. Many users who have multiple accounts do not have access to all of their accounts from each location. For example, one user who regularly works from home cannot read her work email at home due to corporate security restrictions. She simply asks her coworkers to send email to her personal address during the days she works at home.

3. System design

3.1 An email message store

Many of the limitations of current email systems discussed in the previous sections can be traced to limitations in the data structure in which the messages are stored. The prototype system I am developing has an underlying mes-

sage store with services layered on top that provide substantial improvements for organization, retrieval, addressing and navigation.

The message store in the prototype is a relational database. A full text index is a basic service provided on top of the database. The database schema supports end-user supplied metadata which is also indexed. The end-user metadata is used to create categories of messages. Users create categories in the system through queries on the database, rather than filing messages into a certain location on the file system.

A simple query based interface is sufficient for most queries. An advanced interface is available for creating complex queries, as well as for editing queries. Filters, queries and folders are interchangeable within the system. A user may save a query as a standing query, which then appears as a traditional folder in the interface. For example, if a user wants to create a category for a mailing list, she can query for the mailing list address and save the category as a folder. User studies are needed to determine the appropriate interface and methods of user interaction.

This service greatly simplifies the categorization process. The benefit is that users no longer have to manually move messages in order to categorize them. Instead of filing messages into fixed categories, users may create additional categories on the fly. Categories are simply views on portions of the collection which allow messages to be in multiple and overlapping categories. Categories may contain other sub-categories that emulate the folder hierarchy in traditional mail applications.

Few electronic mail systems take advantage of basic information retrieval techniques that can improve both precision, recall and can reduce retrieval and classification time. In most current systems, the search function iterates over the specified collection each time until it finds the desired keyword. In the prototype, I implement full text indexing with Boolean searching. A full text index of the message store allows query results to be returned quickly. Boolean queries are useful for advanced users to narrow down results in large collections. Results may be further narrowed through iterative queries. Clustering and proximity searching is left for a future version.

3.2 Role and identity management

Most modern email applications allow users to check email from multiple accounts and to send email from multiple email addresses. This functionality is generally referred to as “roles” or “personalities.” However, this function only allows the user to select the from: header in the mail message from a predefined list. There is no matching of identities and roles or pairing the roles for senders and receivers. Finally, these identities and roles cannot be used for organization or retrieval.

The prototype system includes a notion of an "individual" comprised of multiple facets: name forms, email addresses, roles, contact information, notes, etc. Each individual has a locally unique identifier within the email collection that allows senders and recipients to have a persistent "identity." This form of authority control is useful for mapping multiple entries into one entry for the purposes of retrieval. [13]

The most common method for users to file messages is by sender. For this reason, the prototype system automatically generates a category for each identity in the collection. This significantly reduces the amount of categorization for most users and the cognitive overhead associated with remembering multiple name forms for a single person. A disadvantage is that users must associate every email address to an identity. New addresses are recognized automatically through similarities in names and email addresses. User testing will determine whether the time and effort saved in categorization outweighs the costs of associating email addresses.

By attaching role information to an identity, the system can perform "role matching." For example, if a user sends a message using a personal role to someone who is both a friend and a coworker, the application will use the recipient's personal email address by default. More complex matching can be achieved through the use of temporal and geographic facets.

Another advantage of using a canonical identity, rather than a series of email addresses is that it improves the reconstruction and display of threads. The system is able to display an entire conversation with any individual, including messages both sent to and received from that person.

3.3 Time based classification

One common user practice is to categorize messages by ranges of dates and times. In the prototype, simple time based categories are available by default, for example, email received today, this week, this month or this year. Other ranges can be added without difficulty. The advantage is that time based categories can be combined with and overlap with other categories (queries). The prototype provides a simple but powerful, interface for selecting time based queries. The user is able to select dates or date ranges on a calendar that are translated into queries. For example, it is simple to find all email from Bob Jones from January to February 2002 by selecting the identity and time range from the interface.

4. Conclusions and future work

I conducted user interviews that revealed problems in current email systems, including role conflict, high cognitive overhead associated with organiza-

tion and retrieval, inability to navigate conversations and difficulties in addressing messages. The prototype system addresses many of these problems through improvements in the message store, support for identity and roles, authority control and novel query interfaces.

In the future, I will conduct performance evaluations, including user studies, to test improvements made in the prototype system. Performance evaluations will include precision, recall, speed and efficiency comparisons. User studies will include an analysis of organization versus retrieval time, ease of use and effectiveness of various interfaces.

5. Related work

A number of research and enterprise email systems have been built on top of databases or other indexed data structures, including Lotus Notes, Microsoft Exchange, Novel Groupwise, HP OpenMail and Compaq CRC Pachyderm. However, many of these systems require additional programming in order to expose or implement many of the features discussed in this paper. [6] [12] A number of third party applications, such as Altavista Discovery, Enfish Onespace and Glimpse, can be used to index email collections. [15]

The Lifestreams system stores records in a database as a time ordered stream and all presentation is time based. The system provides tools to select and navigate time periods. [7] [8] [9]

The TimeStore system is designed to help users locate messages in their mail collection through a time based display that helps the user reconstruct temporal cues to select the correct message.[2] [10] [11]

References]

[1] David Abrams, Ron Baecker, and Mark Chignell. Information archiving with bookmarks: Personal web space construction and organization. In Proceedings of ACM CHI'98 Conference on Human Factors in Computing Systems, pages 41–48, New York, NY, 1998. Association for Computing Machinery.

[2] Ron Baecker, Kellogg Booth, Sasha Jovicic, Joanna McGrenere, and Gale Moore. Reducing the gap between what users know and what they need to know. In Proceedings of the 2000 International Conference on Intelligent User Interfaces, Architecture and Experience, pages 17–23, 2000.

[3] Olle Bälter. Electronic mail from a user perspective: Problems and remedies. Master's thesis, Royal Institute of Technology, IPLab, NADA, KTH, 10044 Stockholm, 1995.

[4] Olle Bälter. Electronic Mail in a Working Context. PhD thesis, Royal Institute of Technology, IPLab, NADA, KTH, 10044 Stockholm, 1998.

[5] Hilary D. Burton. Famulus revisited: Ten years of personal information systems. Journal of the American Society for Information Science, 32(11): 440–443, November 1981.
[6] James Donahue and Willie Sue Orr. WALNUT: Storing electronic mail in a database. Technical Report TR-CSL 85-9, Xerox Palo Alto Research Center, April 1986.
[7] Scott Fertig, Eric Freeman, and David Gelernter. Finding and reminding reconsidered. SIGCHI Bulletin, 28(1), January 1996.
[8] Scott Fertig, Eric Freeman, and David Gelernter. Lifestreams an alternative to the desktop metaphor. In Proceedings of ACM CHI'96 Conference on Human Factors in Computing Systems – Conference Companion, pages 410–411, 1996.
[9] Eric Freeman and David Gelernter. Lifestreams: A storage model for personal data. SIGMOD Record (ACM Special Interest Group on Management of Data), 25(1), 1996.
[10] Aleksandra Jovicic. Retrieval issues in email management. Master's thesis, University of Toronto, 2000.
[11] Sasha Jovicic. Role of memory in email management. In Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems, volume 2, pages 151–152, 2000.
[12] Jack Kent, Douglas B. Terry, and Willie-Sue Orr. Browsing electronic mail: Experiences interfacing a mail system to a DBMS. In François Bancilhon and David J. DeWitt, editors, Very large data bases: 1988, 14th VLDB, Los Angeles, USA, August 29–September 1: proceedings of the Fourteenth International Conference on Very Large Data Bases, pages 112–123, Los Altos, CA 94022, USA, 1988. Morgan Kaufmann Publishers.
[13] F. W. Lancaster. Vocabulary Control for Information Retrieval. Information Resources Press, Arlington, VA, second edition, 1986.
[14] Ann Lantz. Heavy users of electronic mail. International Journal of Human-Computer Interaction, 10(4): 361–379, 1998.
[15] Udi Manber and Sun Wu. GLIMPSE: A tool to search through entire file systems. In USENIX Association, editor, Proceedings of the Winter 1994 USENIX Conference: January 17–21, 1994, San Francisco, California, USA, pages 23–32, Berkeley, CA, USA, 1994. USENIX.
[16] Steve Whittaker and Candace Sidner. Email overload: Exploring personal information management of email. In Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems, volume 1 of PAPERS: Collaborative Systems, pages 276–283, 1996.