Redesigning Email for the 21st Century
Workshop Position Paper

Steven L. Rohall
IBM T.J.Watson Research Center
One Rogers Street
Cambridge, MA 02142 USA
+1 617 693 1840
steven_rohall@us.ibm.com

ABSTRACT

Electronic mail has become the most widely used business productivity application. However, people increasingly feel frustrated by their email. They are overwhelmed by the volume, lose important items, and feel pressure to respond quickly. Though email usage has changed, our email clients largely have not.

Over the past several years, our research group, in conjunction with the Lotus Software group within IBM, has been investigating how people use email and how we might design and build a better email system. We have developed novel visualizations of the documents within mail databases to ease understanding and navigation. We have prototyped systems that incorporate synchronous communication and the ability to annotate individual messages. We have also considered how the structure of information within email might be exploited to provide better automatic summarization.

Throughout this work, I have come to believe that no single feature will solve the "email problem." Email usage is as individual as email users; everyone has their own, idiosyncratic way of dealing with electronic mail. While a new feature may benefit a large number of users, the most effective email system may need to include a large number of such features. The interesting research question, then, is how can a large number of new features be incorporated in email systems in a way that helps, rather than hinders, users.

Keywords

Email, threads, visualization, summarization, chat.

MOTIVATION

Electronic mail has become the primary business productivity application. It has emerged as the most-used communications tool in the US and Canada and, according to an Institute for the Future study, 97% of workers report using email daily or several times each week [12]. In fact, US workers average 49 minutes a day managing email, and 25% spend more than one hour per day on that task [7].

Despite our reliance on electronic mail, our tools for handling email have failed to keep pace [5]. First, workers feel **overwhelmed** by their email. The average user gets approximately 24 messages per day while "high-volume" users can easily get several hundred [11] [12]. Ironically, 34% of internal business messages were deemed "unnecessary" [7]. Second, people use their email inboxes to manage their tasks, yet they complain, **"things fall through the cracks."** Current organizational structures within email clients, such as folders, prove inadequate, especially for high-volume email users [17]. Finally, people feel pressure to be **more responsive** in replying to email messages, reporting that 27% of messages sent "require" immediate attention [7].

Email clients have changed little since they were first invented. Although today's email clients are more graphical with onscreen buttons, pull-down menus and rich-text display, they are essentially "souped-up" cousins of the email clients from thirty years ago. Most email clients today have the same set of features and organizational structures: multiple folders in which messages can be filed, a textual listing of the messages within a given folder, and the ability to preview a selected message. However, studies have shown that folder systems quickly degrade with the number of messages people receive [17]. Most people end up keeping all of their email in one large folder [1]. The content and use of email has also changed. In addition to traditional letters, email now consists of invitations, receipts, transactions, discussions, conversations, tasks, and newsletters, to name a few variations [5].

To address these problems, our research group has been investigating electronic mail. Our investigations have included user observations and interviews, design mockups, prototype implementations, and user evaluations. In this paper, I will describe our work on augmenting email with visualizations, synchronous awareness and communication, the ability to annotate messages, and automatic summarization of the content of messages.

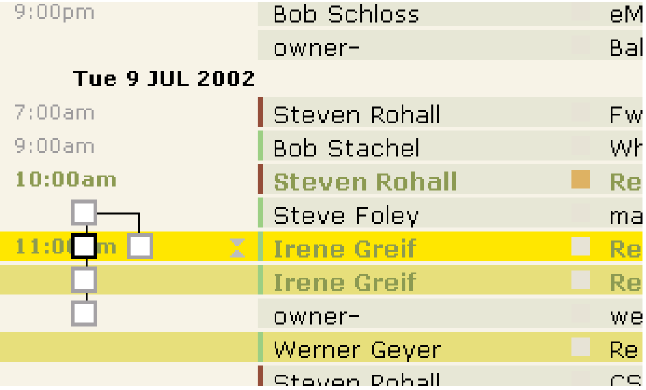
VISUALIZATIONS

Email has several numeric attributes (e.g., message size, number of attachments) that are easy, but not very useful, to visualize. Therefore, we have had to look deeper into the structure of electronic communications to determine salient features that should be visualized. We have focused on three such features: message threads, time, and the content of documents.

Message Threads

The idea of message (or conversation) threads is borrowed from discussion databases. At its simplest, a conversation thread in electronic mail represents the series of replies to a message and the replies to those replies. Several systems have attacked the problem of visualizing threads within discussion databases [4][15][16]. Some email systems also have support for message threads. Typically, they display a textual representation of the thread tree using a Windows Explorer-like tree control.

One of our prototypes supports threads of email messages by providing a visualization of the thread tree when any message is selected [13][14]. At the same time, other messages in the thread are shown with a secondary highlighting color. The secondary highlights show the user which messages are related to the selected one (Figure 1.). Hovering over the nodes in the tree view provides summary information of those messages and clicking on a node automatically scrolls the list view to the selected message. This automatic scrolling is very useful when a thread has gone on for several days and older messages are no longer visible on screen.

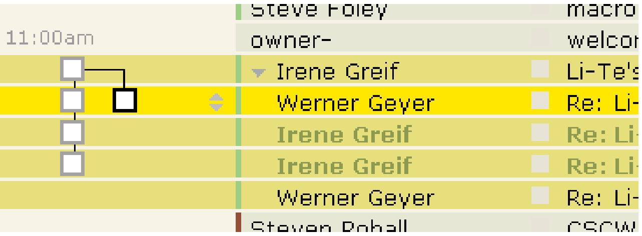


9:00pm	Bob Schloss	eM
	owner-	Bal
Tue 9 JUL 2002		
7:00am	Steven Rohall	Fw
9:00am	Bob Stachel	Wh
10:00am	Steven Rohall	Re
	Steve Foley	ma
11:00am	Irene Greif	Re
	Irene Greif	Re
	owner-	we
	Werner Geyer	Re
	Steven Rohall	CS

Figure 1. The client in use, showing the thread map on the left, and online awareness indicators next to each name.

Much research has been devoted to the visualization of trees. However, the bulk of that work has concentrated on the display of trees with hundreds or thousands of nodes. Our problem is simpler: message trees, from our observations, are not very deep nor very bushy. This allows us to use a simple graphical representation of the message thread.

This prototype also supported the ability to "gather" a thread. When a thread is gathered, all of the individual messages are collapsed into one item in the list view. This item appears in the list at the place where the most recently received individual message was located and, as new messages in that thread are received, the gathered thread moves to those positions. Moving the gathered thread ensures that the user will always have access to the entire thread of messages whenever a new message is received, even if the thread has been inactive and has scrolled out of view. At any time, the user may "scatter" a thread and replace the single item with the individual constituent messages in their proper places within the list (Figure 2).



	Steve Foley	macro
11:00am	owner-	welcor
	Irene Greif	Li-Te's
	Werner Geyer	Re: Li-
	Irene Greif	Re: Li
	Irene Greif	Re: Li
	Werner Geyer	Re: Li-
	Steven Rohall	CSCW

Figure 2. A gathered thread.

Time and Timelines

One problem with the thread visualization described above that became apparent was that, while it was easy to determine parent-child relationships among the messages, it was difficult to determine time relationships among the messages. For example, two messages would clearly be seen to be first-generation replies to another message. However, the visualization in the prototype falsely implied that they were received at the same time when, in fact, they might have been received at very different times. This observation led us to consider alternate visualizations of the thread tree that also incorporated time (Figure 3).

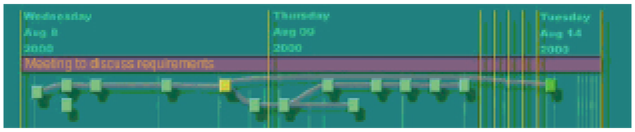


Figure 3. Message tree with timeline.

In this design, the vertical lines represent day boundaries. The text in the middle band is the subject of the thread. The color-coding of the nodes represents the relationship of the message senders to the recipient (e.g., derived from a corporate directory). Time is non-linear in this display; days with little or no activity are shown compressed.

While other systems have used time as an organizing principle of the user interface, they have concentrated on using a time-based visualization for displaying the entire mailbox, perhaps broken down by sender [8][9]. These sorts of visualizations tend to have large gaps at times when no messages are received that limit their usefulness. By limiting our time-based display to an individual thread and incorporating a non-linear display of time, the proposed visualization avoids these problems.

Document Content

We have also explored visualizing document content through the use of reduced-resolution overviews. The idea is that people can quickly see which messages contain images or the meeting agenda, for example, by the look of the message even if they are not able to read the actual contents of the message. One of our thread map designs displayed the low-resolution thumbnail of a message when the user hovered over a node in the map. In this way, the user could "see" the message before navigating to it.

Although other systems have used reduced-resolution document overviews in their user interface [2], email, in particular, would benefit from such visualization. Email has structure that other client software has failed to exploit. With an overview, people can quickly pick out different types of email (e.g., agendas, online purchase receipts, corporate-wide announcements). Automatic classification of this sort has proven error-prone [5].

SYNCHRONOUS AWARENESS AND MESSAGING

Instant messaging, or "chat," has become an increasingly important application for communication. Products such as Lotus Sametime from IBM have brought instant messaging into the workplace. Traditional use of instant messaging has a serious limitation, however: unless a user is in your "buddy list," it is difficult to monitor that person's status or initiate communication.

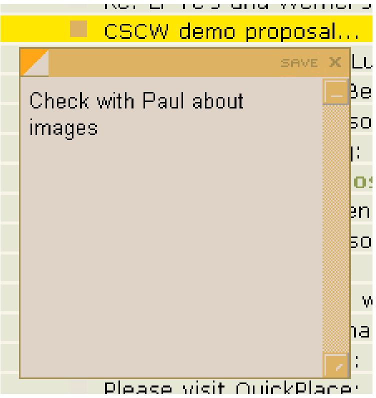
Several of our prototypes addressed this limitation by providing a "live names" feature where names in the "From," "To," and "CC" fields show the user's online status and allow one to start a chat simply by clicking on the person's name. (See the red and green icons next to the senders' names in Figure 1.) In a business sense, what is important is not that a person is your "buddy," but rather that you are able to get them the information they need in a timely fashion. In a large company, this context-based awareness opens up the possibility of chatting in reply as easily as composing an email reply.

Further, the chat transcript in our implementation is saved as a part of the thread of messages. In fact, a user can reply with email or chat based upon the urgency of the communication. Threads of conversation may move between synchronous and asynchronous modes fluidly.

ANNOTATIONS

Another shortcoming of current email applications is the lack of a feature for annotating email messages. Annotations serve two purposes. First, they serve as reminders for the user much in the same way that physical notes attached to documents help remind people of related information. Second, in situations where more than one person is sharing an inbox, the annotations allow people to communicate with each other about the work (e.g., "check with me before responding to this" or "spoke to her on the telephone—no need to follow up").

Our ReMail prototype annotation feature has two parts. One is the ability to place a colored icon on a message without any annotation text. For example, all messages with red icons might need immediate attention. In addition, the user is able to add text to the annotation; the icon changes to indicate that there is text included (Figure 3). These icons could be used to sort and categorize messages.


CSCW demo proposal...

Check with Paul about images

Please visit QuickPlace:

Figure 3. An annotation attached to an email message. The orange "dog ear" in the upper left is the user-specified color.

AUTOMATIC DOCUMENT SUMMARIZATION

While much of our work has focused on improving the design of email clients by adding useful features or new visualizations, we have also investigated ways to help users by providing summaries of email messages with the hope of reducing the amount of time it takes to "triage" their inboxes or to find a message for which they are searching.

There has been much research into document summarization and there are commercial products that perform summarization by extracting key words, phrases, or sentences. The problem with using commercial summarizers on email is that they were developed to summarize long, well-authored documents. Email messages are typically neither long nor well authored.

On the other hand, email messages do have structure that can be exploited. First, email messages in the corporate environment tend to concentrate on people and events. Adding special support for finding names and dates would be extremely useful. Second, message threads themselves can provide additional context and background information that can help off-the-shelf summarizers perform better on short, informal email messages. Others have investigated summarizing threads in discussion databases [6], a job that is somewhat simpler since threads in discussion databases are typically complete. In email, threads are often incomplete since messages may not have been saved or may have been deleted.

We developed a prototype email summarizer that preprocessed the messages in a thread and passed them to a commercial summarization system [10]. The preprocessing included removing quoted history text and email headers to create a new, synthesized document for summarization. By removing these features, we ensured that the summarizer did not weigh them too heavily. At the same time, the prototype looked for proper names and dates in both the message headers as well as the message body. The date extraction software utilized regular expressions to find date terms such as “tomorrow” or “last week” and then translated such terms into numeric dates.

While our initial results were encouraging, there is still much work to be done on the exploitation of the unstructured information found in electronic mail. For example, on short messages, the summary was often about the same length as the original message, possibly creating more work for the user rather than less. On longer messages, which the system summarized better, the summaries still tended to be too long. This prevented the system from displaying the summaries optimally in the user interface. For example, we wanted to display the summary when the user hovered over a message subject, but this was not possible. In addition, reading the summaries could be difficult since connecting words and phrases had been eliminated.

OTHER INVESTIGATIONS

Other research in our group has looked at social networks in email, defining taxonomies of email types, collaboration around shared inboxes, and email on mobile devices. Electronic mail continues to be a primary focus of our group and we are presently prototyping a second generation of our email client that incorporates the best features of our prior work and serves as a platform for testing new ideas.

CONCLUSION

The problems with electronic mail are well documented. In fact, it is impossible for us to demonstrate one of our prototypes without people relaying their personal email horror stories. One person, for example, who is in several standards organizations and is also a line manager told us that he receives between 300 and 400 messages a day. Clearly things have gotten out of hand.

We as researchers, designers, and developers can do much to improve the situation. Our group has looked at numerous aspects of electronic mail including visualization, navigation, annotation, synchronous communication, and summarization. We are naïve, though, if we think that piling a bunch of new features, all of which look “cool” independently, onto existing email clients will solve the problem. The data is overwhelming the user; we cannot afford to have our email applications exacerbate the problem.

At the same time, there are as many ways of dealing with email as there are users of email; a feature or set of features that works for one group of users may be less useful to another group of users. On the other hand, to produce a viable product, companies will be forced to include an increasing number of features so that their applications will appeal to the largest number of users. The real challenge going forward, then, is one of integration design. How can we incorporate new features in ways that help all users by revealing the features that are needed when they are needed and hiding the ones that are not? How can we provide flexibility in our applications to switch among feature sets as a user’s role changes? These are difficult questions that our research group is actively pursuing.

REFERENCES

1. Bälter, O., “Electronic Mail in a Working Context,” doctoral dissertation, Royal Institute of Technology, Stockholm, Sweden, 1998.

2. Bederson, B. and J. Hollan, “Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics,” *Proceedings of UIST’94*, Marina del Rey, CA, November 2-4, 1994, pp. 17-26.

3. Cadiz, J.J., L. Dabbish, A. Gupta, and G.D. Venolia, “Supporting Email Workflow,” Microsoft Research Technical Report # MSR-TR-2001-88.

4. Donath, J., K. Karahalios, and F. Viegas, “Visualizing Conversations,” *Proceedings of HICSS-32*, Maui, HI, January 5-8, 1999.

5. Ducheneaut, N. and V. Bellotti, “E-mail as Habitat,” *Interactions*, 8(5), Sept.-Oct. 2001, ACM, pp. 30-38.

6. Farrell, R.; P.G. Fairweather; and K. Snyder, “Summarization of discussion groups,” *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, pages 532–534, Atlanta, GA, 2001.

7. “The Spam Within: Gartner Says One-Third of Business E-Mail is ‘Occupational Spam’,” Gartner, Inc. press release, April 19, 2001.

8. Jovicic, S., and R. Baecker, “Time-Based Archiving and Retrieval of Email,” Workshop on History-Keeping in Computer Applications, Human-Computer Interaction Laboratory, University of Maryland, December 3, 1999.

9. Kudo, M., M. Tanaka, and Y. Koseki, “Information Visualization for Electronic Mail Management,” *Proceedings of Visual’97*, San Diego, CA, December 15-17, 1997.

10. Lam, D., S.L. Rohall, C. Schmandt, and M.K. Stern, “Exploiting Email Structure to Improve Summarization,” poster session, CSCW’02, New Orleans, LA, November 16-20, 2002.

11. Levitt, M., “Email Usage Forecast and Analysis, 2000-2005,” IDC Report # W23011, September 2000.

12. “Pitney Bowes Study Reveals Increased Use of Electronic Communications Tools Among North American and European Workers,” Pitney Bowes press release, August 7, 2000.

13. Rohall, S.L., D. Gruen, P. Moody, and S. Kellerman, “Email Visualizations to Aid Communications,” *Late Breaking, Hot Topic Proceedings of the IEEE Symposium on Information Visualization*, San Diego, CA, October 22-23, 2001, pp. 12-15.

14. Rohall, S.L. and D. Gruen, “ReMail: A Reinvented Email Prototype,” demo session, CSCW’02, New Orleans, LA, November 16-20, 2002.

15. Sack, W., “Conversation Map: A Content-Based Usenet Newsgroup Browser,” *Proceedings of IUI’00*, New Orleans, LA, January 9-12, 2000, pp. 233-240.

16. Smith, M., J. Cadiz, and B. Burkhalter, “Conversation Trees and Threaded Chats,” *Proceedings of CSCW’00*, Philadelphia, PA, December 2-6, 2000, pp. 97-106.

17. Whittaker, S. and C. Sidner, “Email Overload: Exploring Personal Information Management of Email,” *Proceedings of CHI’96*, Vancouver, B.C., April 13-18, 1996, pp. 276-283.