Dynamic Coordinated Email Visualization

Simone Frau
Computing Laboratory,
University of Kent,
Canterbury,
Kent CT2 7NF, UK
sf31@kent.ac.uk

Jonathan C. Roberts,
Computing Laboratory,
University of Kent,
Canterbury,
Kent CT2 7NF, UK
j.c.roberts@kent.ac.uk

Nadia Boukhelifa
Computing Laboratory,
University of Kent,
Canterbury,
Kent CT2 7NF, UK
n.boukhelifa@kent.ac.uk

ABSTRACT

Many computer users receive hundreds (if not thousands) of emails per week; users often keep these emails and have many years of personal emails archived: users use their stored emails to manage appointments, to-do lists, and store useful information. In this paper we present an interactive email visualization tool (Mailview) that utilizes filter and coordination techniques to explore this archived data. The tool enables users to analyze and visualize hundreds of stored emails, it displays the emails on time-dependent plots enabling users to observe trends over time and perceive emails with similar features. Interaction is an important aspect of finding meaning within information, hence the tool utilizes focus+context views, dynamic filters, detail-on-demand techniques and coordinated views, finally, we discuss various methods that enable the system to be designed such that it can display hundreds of objects at interactive rates.

Keywords

Email visualization, information visualization, coordinated views, exploration, email archive

1. INTRODUCTION

The use and growth of email is staggering, even with the divergence and growth of other communication technologies such as the mobile phone and short messaging, still the use of email is growing. According to the University of California Berkeley "How much information?" 2003 evaluation, about 31 Billion emails were sent per day in 2002 and a prediction of 60 billion will be sent per day in 2006 [HMI03]. Many users store past emails for reference: archiving them in folders for future observation, they may store hundreds and thousands of emails in these archives; indeed the authors themselves have a joint archive of approximately 100,000 emails. They are stored such that information can be referenced for future use.

This personal email archive provides a rich and diverse dataset, and it is both interesting and potentially useful to analyze and visualize this information. First, visualization can help with information retrieval. There are known problems with email archiving [Whi96] that could be overcome by visualization. Such problems include, generating appropriate folder names, reconstructing these labels when they are needed to search for the required data, and being consistent in grouping similar material in the same folder, indeed, there is an ontology issue (as some mails could be legitimately stored under different categories). Second, visualization can be used to help the user analyze the information for trends or make observations. These observations could be utilized to control and influence work-patterns or to potentially benefit the effectiveness of spam-filters or automatic mail transfer (such as Exim) and archiving engines. For instance, if it was observed that the majority of spam messages arrived between a certain times of day then the spam filter could be dynamically adapted appropriately to catch more spam during that period.

Many email programs, like for example Outlook Express, Exmh, Mutt or Pine, besides showing the content of the emails display some features of the emails themselves; such as the receiving date and time, sender name, email length, and information about whether the email has attachments, etc. They are very good at displaying information about single emails, but do not traditionally depict aspects or trends of multiple emails. Furthermore, time is an important factor in emails: they are downloaded in chronological order, often stored in order, email

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

addresses may change when senders change employers, and minutes of meetings or mail-shots may be weekly occurrences. Thus, in contrast to current email visualization tools, the aim of our system is to visualize hundreds of emails displayed in a chronologically based visualization.

This paper describes the type of data that can be represented (section 2), related work (section 3), and our Mailview application (the remainder of the paper). Mailview provides a visualization of emails focused on showing the emails chronologically and it uses focus+context techniques and multiple views. It includes (a) an overview of the multiple emails represented by glyphs, (b) two zoomable and coordinated views depicting the arrival time, specifically the chronological order in which they arrived, and the relative size of the emails and (c) depicts specific details of selected single emails. Mailview is not meant to replace a normal email viewer, but to be used in partnership with their standard mail reader; Mailview may enhance the users understanding of their stored emails.

2. EMAIL DATA

Email data provides a rich and interesting source of information. Indeed, there are many attributes and statistics that can be calculated and visualized. We classify this information into four categories: meta-information, content, intra-email and inter-email.

First there is the straightforward information about the mail itself, which is merely stored in the email header. This meta-information includes the senders email address (their familiar name may be retrieved from an address book), recipient email address, delivery date/time received date/time, the path of the email through various servers, content mime types and whether it contains attachments, subject line and the content of the email body. Not only is it interesting to investigate trends over every email received, but also it would be useful to filter and depict emails from specific senders, or by subjects.

Second, it is useful to view the content of the emails. The content of the email may be stored in different types (text, html, pdf etc), and the challenge of any mail viewer is to seamlessly view each of these different types. Moreover, although the average size of an email may be 59KB [HMI03] some emails are much larger. This means that many emails do not fit into a single screen, the information needs to be scrolled. Hence, there are opportunities to use focus+context techniques to abstract and better represent this text information.

Third, statistical information can be generated from each email individually. We name this intra-email analysis (within an email). Intuitively, we can easily calculate the length or size of the email, e.g. we can calculate the number of words or letters in the text, the number of pages, and the amount of memory it takes up. But, other statistical calculations can be made on a single email. For instance, word frequencies can be calculated to find common and recurrent words in the email, or emails can be classified based on an analysis of the content (such email classification is a necessary part of spam filtering).

Finally, observations and calculations can be made on a collection of emails. We classify this as being an inter-email analysis (between emails). Such analysis is often used to help navigate news articles. For example, some news reading programs display threaded information, depicting an initial post with all occurrences of replies to that post, more generally threaded-visualization depicts all objects that mention the same topic (thread) of information.

3. RELATED WORK

Electronic mail is a widely used communication medium. Its role, however, has expanded from a mere information carrier into a dynamic medium where users can have conversation threads, delegate tasks, plan meetings and exchange minutes and multimedia files. As a result of the growing usage of emails, users are overwhelmed by the increasing traffic on their mailbox and the varied nature of emails (from professional, personal and sales to spam and harmful emails). Again referring to the “How much information?” report [HMI03], by May 2003 almost 55% of emails sent were classified as spam. Consequently, researchers are continuously working on new interfaces to help users better manage their emails.

Much recent development, that the general public would be aware of, is the continued development of facilities that allow users to browse, manage and query emails more efficiently. Certainly, most modern mail viewers permit users to filter or highlight mail messages that conform to certain search criteria (such as displaying all the messages sent by a specific user, or on a particular subject). The tools allow the user to quickly and efficiently search through multiple folders and display exact and partial matches to the queries. Other tools or add-ons allow users to manage their mail, such as automatically archiving or deleting emails.

In the research domain, one important strand of email visualization research is the analysis and visualization of relationships between emails (detailed in section 2 as inter-email visualization and analysis). One important area of study focuses on clustering emails into meaningful groups. For example, Sudarsky and Hjelsvold [Sud01] make use

of the hierarchical nature of the domain names present in email addresses for the clustering criteria. While others have investigated modeling and characterizing email conversations (e.g. Venolia and Neustaedter [Ven03]).

Another area of research is visualizing these threads of conversation depicting, for instance, where users hold ongoing discussions by email. Similar techniques are applied to analyze and visualize social networks such as determined from Internet relay chat (IRC) and other forms of data. For example, Mutton [Mut04] infers associations between IRC participants based on parsing the conversation text. He utilizes graphs to visualize the inferred relationships, with the nodes being the participants and the edges the associations.

EmViz [Hec97] also uses a graph-based layout to visualize correspondence from email traffic within an organization. EmViz uses a cone-tree to depict the hierarchy of the organization. Additional information is annotated onto this reference structure, including, the quantity of emails sent/received by an individual is denoted by the size of the node, and the color of the edge depicts the frequency of peer-to-peer correspondence. Similarly, the aim of eArchivarius [Leu03] is to highlight existing communities of people. The eArchivarius tool visualizes and organizes collections of emails in various ways, one example demonstrates a cluster-based visualization, where each sphere glyph represents a person and the more emails two people exchange the closer the glyphs become. Colors can represent various attributes such as the topic (where the confidence of 'an email being correctly classified' is realized by the intensity of the color).

Thread Arcs [Ker03] is another graph-based tool that visualizes relations between emails. In this representation the threads are chains of emails where each one (except the root) is a reply to another belonging to the chain. Arcs link each 'child' (an email being the reply to another one) to its 'parent'' (the email the child replies to) showing the connections among them and the progress of a conversation. The user can interact selecting any email in the thread.

The emphasis of the aforementioned related work focuses on intra-email visualization, however the focus of our work in this paper is on visualizing the personal email archive especially displaying the emails on time related plots. So far, there has not been much research investigating methods to visualize this archive. Jovicic [Jov00] does describe a system to visualize personal email data and indeed discusses that time is important. In fact, email data and metadata has a temporal nature, and thus visualizing the mailbox can benefit greatly from the already existing visualization tools for temporal information.

We are in agreement with Jovicic [Jov00], who states that users tend to ignore time as a crucial factor in email communication. She added that most emails tend to have personal and informational qualities. When an email involves personal events, activity, people involved and place of the event are typically well remembered explicitly [Jov00]. This is often known as episodic memory, which details the human ability of remembering things that happened at a particular time and place. Jovicic, discussing work by Friedman [Fri93], mentions that the memory reconstruction relies on 'temporal cycles' which are used to estimate the time elapsed since the event and to provide a frame of reference within which an event can be placed. Jovicic plots the mails on periods of 'days' and 'weeks' particularly highlighting the weekends. In another study Begole et al [Beg02] monitors computer activity minute-by-minute in order to establish rhythms. Begole et al display computer activity timelines coupled with information about the location of the activity, online calendar appointments, and email activity. This linkage helps find patterns of individuals according to time of day, location and day of the week.

Temporal based charts have been used to visualize historic, and other chronological data in other datasets. Email data has much similarity with this information and thus we briefly mention other related temporal/historic visualization research. For instance, Weber et al [Web01] describe some of the popular visualization tools for time-series data; these include sequence charts, point charts, bar charts, line graphs and circle graphs. One of the most utilized techniques to visualize temporal data across various fields is timelines.

A timeline is a linear visual representation of time-varying events. According to Kumar et al [Kum98], the earliest use of timelines in the published literature can be traced to William Playfair. Timelines have been used to display historical information very efficiently. Extensions to this technique resulted in 3D dynamic timelines [Kul96], spiral timelines [Web01], RiverThemes [Hec97] and lifelines [Pla96]. Moreover, Kumar et al [Kum98] presented a framework and interface for representing temporal information. Finally, Karam [Kar94] suggested a model to automate and generalize timelines.

4. DESIGN & IMPLEMENTATION

4.1. Data Gathering and Preparation

Various email readers store the emails in different forms (from human readable files with one file per message, tagged file including multiple messages, or proprietary databases). In this paper we assume that

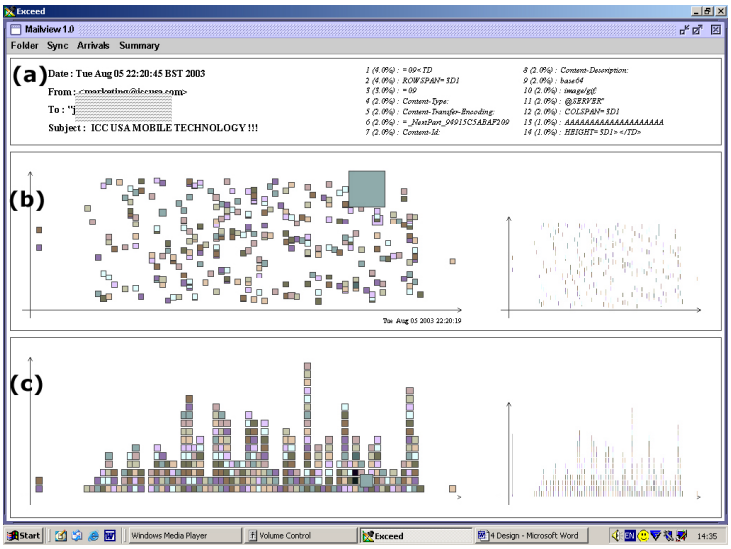

Figure 1 A screenshot of Mailview, depicting emails from the spam directory from our personal email archive.

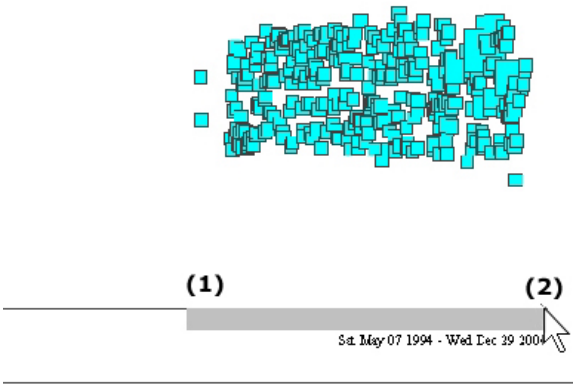

Figure 2 The user can zoom into a particular area either by selecting a bounding region directly on the visualization, or by dragging the mouse along an axis line (position 1 to 2) as shown by this screenshot of Mailview.

the emails are archived in individual files, the files may be grouped together in folders, that the files are MIME encoded (the Multipurpose Internet Mail Extensions format allows enriched content such as images, audio, and attachments) but they are stored in essentially text documents.

Gathering and preparing data consists of first choosing the mails to be viewed, then scanning the mails to create a data-structure containing an abstract representation of the mail data required for visualization. The data-structure is hierarchically organized mirroring the folders hierarchy, and each node corresponds to an email and includes some of its features, such as sender, receiver and so on. It also contains fields concerning the emails presentation (such as layout coordinates). Besides extracting some basic fields (date, sender, receiver and subject), we calculate the size of the email, and scan the whole body of the email to obtain a frequency analysis of the most common words and their percentage.

4.2. Visualization

The overarching design was to depict the emails in plots that demonstrated temporal attributes. Hence we display the emails in various temporal based scatter plots that can be scaled and zoomed. Figure 1 shows Mailview.

There are three main layers in the tool, the upper layer (a) depicts details about a specific email (sender, recipient, date, time, and the frequency analysis) the second (b) and third (c) layers depict two views: one context view that shows an overview of the whole display (right), and the other is a zoomed view (left). Each of the plots display dates (days, weeks and months) along the bottom (the x-axis). The plot in the middle layer (b) displays time along the y-axis (from 00:00 at the bottom to 24:00 midnight at the top) and the plot on the bottom layer (c) represents a stacked bar-chart representation (with the y-axis representing the quantity of emails that day).

Each email is represented by a glyph (the user can choose the glyphs) either vertical lines, circles or squares. The relative size of the email is realized by the size of the glyph (with larger emails being realized by larger glyphs), the emails are also colored, the color depends upon the folders in the archive and is automatically allocated, but the user can edit the allocated colors.

We took a design decision to not display too many labels, as there could be potentially hundreds of emails being display, there could be hundreds of labels each denoting an email: which would quickly become unreadable. However, labels are important and a balance must be met. Thus, text information about both the axis and particular details of the highlighted email get displayed as the user brushes over an email glyph. In fact, the use of glyphs enables a large amount of information to be displayed in a small space (which was one of the original goals). Obviously, it would be useful to link a regular email browser to this visualization (or to embed this into such a browser) and again coordinate everything together. However, we leave this

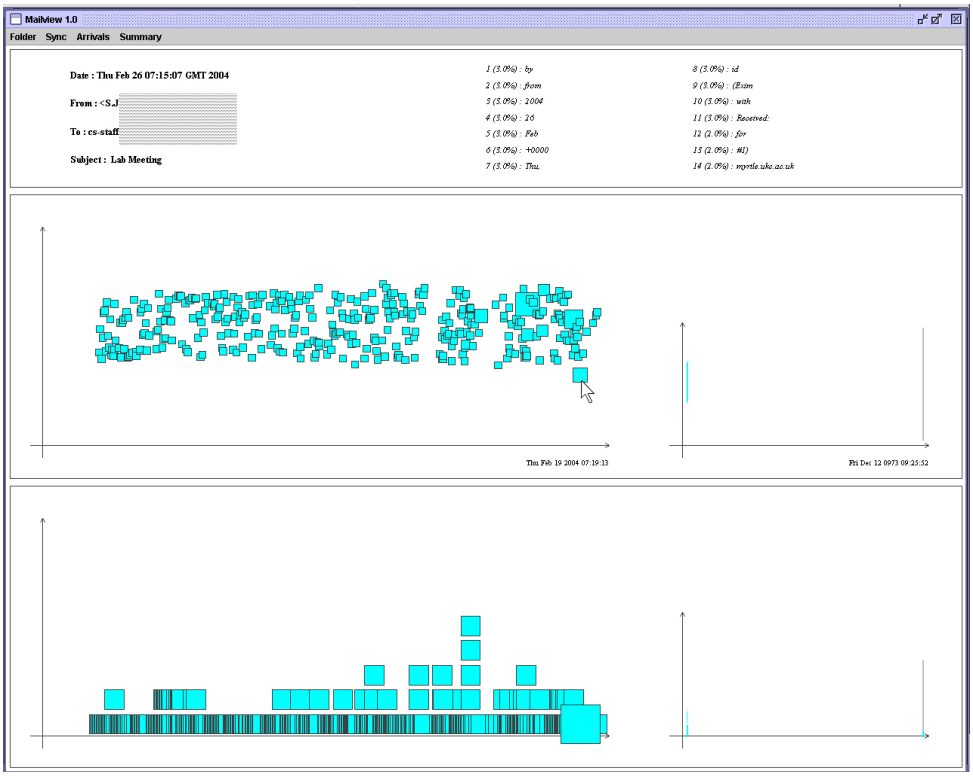

Figure 3 This screenshot shows Mailview displaying emails of laboratory meetings. After a quick observation it is easy to see that most of the emails have arrived during traditional work hours (8.30am to 5.30pm).

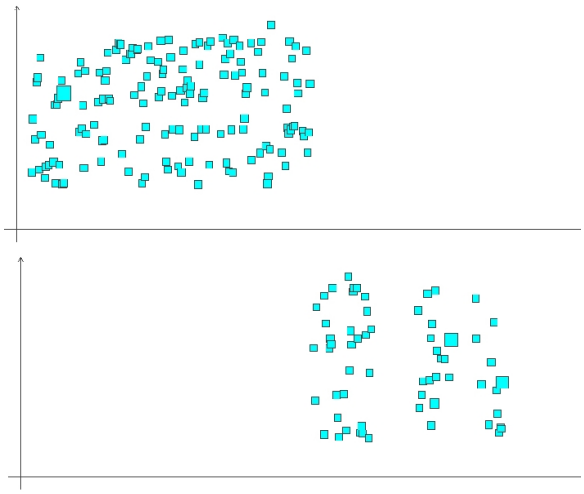

Figure 4. This figure shows a subset of the data from Figure 3, (Upper) depicting the emails that have been sent by one member of staff, and (Lower) by a second member of staff. The visualization clearly depicts when the first member of staff left and the second one took over the role of sending out the regular meeting minutes.

integration of the traditional mail reader and the Mailview visualization for future work.

In fact, the tool was developed using Java, and although Mailview currently includes four plots, glyphs and mappings, each of the plots are inherited from the same abstract class, because they have similar properties and attributes, hence the system is easily extensible.

4.3. Interaction, Coordination & Filtering

As the user brushes over the plots so the underlying glyph is highlighted, the glyph doubles in size, summary details of the appropriate email are updated in the top view (Figure 1a), and the same email (represented in the other window-view) is also simultaneously highlighted. In fact, the user can choose which views are coordinated together to allow the users to compare disparate parts of the plots. Figure 1 shows some emails, which are designated as spam from our email archive; the plot demonstrates that spam arrives throughout the day, and throughout the week.

Furthermore, users can zoom into any area. Zooming can be controlled either by dragging out a bounding box directly on the plot (allowing the user to change the date and time range together), or by dragging the mouse along an axis line (allowing the user to zoom into either a date range or a time range

independently) as shown in Figure 2. Again, these views can be coordinated together to allow the same focused area to be visualized in each view. The zoomed area is also depicted in the overview plots.

Every operation that has an effect to the display (such as zooming, changing what is coordinated, or changing the appearance of the glyphs) is stored in a history list. The user can undo any operation that they have made; this encourages the user to try out scenarios and makes comparison easier.

The system allows emails to be filtered and selected so trends about particular senders or subjects can be spotted. This is different to the above zooming and selection operations, which enable the user to see trends in time. For instance, the user may wish to observe repeating patterns such as to see the email of weekly meetings, or occurrences when employees have left a company. Filtering is achieved through selection. The email is selected when the user clicks on a glyph, this fixes the current detail information, thus when the user thereafter brushes over a detail field such as subject or sender, only those emails that have been sent by that sender are displayed (all the others are filtered out); the user can then select another field and the information is thereafter constrained by two fields.

We demonstrate this filtering mechanism through a simple example. Figure 3 shows some emails that relate to laboratory meetings, they are all stored in one folder in the archive. From an initial observation, it is easy to notice that the emails mostly arrive between 8.30am and 5.30pm. There are some outlyers, including one sent at 7.15 am (under the displayed cursor in the screenshot of Figure 3). The user can now brush over the elements to discover different aspects of the emails. In browsing this dataset we discovered that most of the emails on the left-hand-side were sent by one member of staff, and those to the right by another. This discovery is shown in Figure 4 and in fact demonstrates the time when one secretary left and another one joined the laboratory. At any stage the user can undo the operation to explore another constraint or scenario.

4.4. Performance challenges and solutions

With hundreds of emails it takes time to refresh the display. The bottleneck seemed to be calculating positions at runtime, so in order to speed up the refresh rate we cache the positions in the hierarchical data-structure. This way the positions only need to be recalculated when a range change occurs. Further problems occurred with brushing at interactive speeds (especially finding the closest email to the mouse pointer). Creating a grid, containing a two-dimensional array of lists of emails, solved this. Since we set the cell size to the maximum radius of the shapes, we knew for sure that if we are hovering with the mouse over a shape its center must be either in that cell or in one next to it.

If the feedback is coordinated among windows, then corresponding emails need to be highlighted in the other diagrams. Even though emails datastructure contains a unique identifier field that makes them unambiguously distinguishable, it still would take time to search for the corresponding element to highlight. The chosen solution was to develop a new class, similar to the coordination space of Boukhelifa et al [Bou03], to manage the coordination. Practically, this class stores a list of the email unique identifiers as an associated array, so if on a check the element contains -1, there is no similar email in that diagram, otherwise the number contained will represent the position the email has in the hierarchical datastructure.

5. SUMMARY & FUTURE WORK

We have successfully developed a visualization tool that displays email archive data. The tool enables users to see trends and details of emails within time and date plots. Users can interact, zoom and filter the information in a coordinated exploratory environment. We believe chronological information is important within effective email data perception. Most of our program's functionalities have been developed to be as extendable as possible (both for visualization and coordination), such that the tool can be further developed in the future. Although we have tested the tool on various users, we have yet to accomplish a full user study. We plan to do this in the near future.

There is much functionality, other views and techniques that could be added to the system. In fact, we believe aggregation is an important extension that is missing from other tools and also missing from Mailview at present, such ideas are important and have been used by Larsen et al [Lar96] and Begole et al [Beg02]. Additionally, we know that users do not often remember exact dates of events (as discussed in the related work, section 3) rather they remember periods of time, and hence it would be useful to allow the user to explore the data through a rich set of aggregation commands.

REFERENCES

[Beg02] J.B. Begole, J.C. Tang, R.B. Smith, and N.Yankelovich, “Work rhythms: analyzing visualizations of awareness histories of distributed groups,” Proc ACM CSCW'02, pp.334-343, 2002. New Orleans.

[Bou03] N. Boukhelifa, J.C. Roberts, and P.Rodgers, “A Coordination Model for Exploratory Multi-View Visualization,” Proc International Conference

on Coordinated and Multiple Views in Exploratory Visualization (CMV 2003), pp.76-85. J.Roberts, ed., IEEE, July 2003.

[Fri93] W.J. Friedman, “Memory of time for past events,” Psychological Bulletin 113(1), pp.44-66, 1993.

[Hec97] B. Heckel and B.Hamann, “Emviz - a visual e-mail analysis tool,” Proc New Paradigms in Information Visualization and Manipulation Workshop, pp.36-38, 1997. Las Vegas, Nevada USA.

[HMI03] “How much information? 2003,” www.sims.berkeley.edu/research/projects/how-much-info-2003/

[Jov00] S.Jovicic, “Role of memory in email management,” Proc ACM CHI 2000, Interactive posters, pp.151-152, 2000. The Netherlands.

[Kar94] G.M. Karam, “Visualization using timelines,” Proc International Symposium on Software Testing and Analysis ISSTA, T.Ostrand, ed., pp.125-137, 1994.

[Ker03] B.Kerr, “THREAD ARCS: An Email Thread Visualization,” in IEEE Symposium on Information Visualization, pp.211-218, 2003. Seattle, Washington.

[Kul96] R.L. Kullberg, “Dynamic timelines: Visualizing the history of photography,” Proc ACM CHI 96 Conference on Human Factors in Computing Systems, VIDEOS: Visualization, pp.386-387, 1996.Vancouver, Canada.

[Kum98] V. Kumar, R. Furuta, and R.B. Allen, “Metadata visualization for digital libraries: Interactive timeline editing and review,” in DL'98: Proc ACM International Conference on Digital Libraries, pp.126-133, 1998. Pittsburgh, USA.

[Lar96] S.F. Larsen, C.P. Thompson, and T.Hansen, “Remembering our past,” in Studies in autobiographical memory time in autobiographical memory, D.C.Rubin, ed., Cambridge University Press, 1996.

[Leu03] A.Leuski, D.W. Oard, and R.Bhagat, “eArchivarius: Accessing Collections of Electronic Mail,” Proc ACM SIGIR Conference on Research and Development in Information Retrieval, Demos, p.468, 2003. Toronto, Canada.

[Mut04] P.Mutton, “Inferring and visualizing social networks on internet relay chat,” Proc Information Visualization, IEEE Computer Society, 2004. London, UK.

[Pla96] C.Plaisant, B.Milash, A.Rose, S.Widoff, and B.Shneiderman, “Lifelines: Visualizing personal histories,” Proc ACM CHI 96 Conference on Human Factors in Computing Systems Papers: Interactive Information Retrieval, pp.221-227, 1996.

[Sud01] S.Sudarsky and R.Hjelsvold, “Visualizing electronic mail,” Proc Information Visualization IV'02, pp.3-9, IEEE Computer Society, 2002. London, UK.

[Ven03] G.D. Venolia and C.Neustaedter, “Understanding sequence and reply relationships within email conversations: a mixed-model visualization,” Proc ACM CHI, pp.361-368, 2003.

[Web01] M. Weber, M. Alexa and W. Müller “Visualizing timeseries on spirals,” Proc InfoVis'01, pp.21-28, IEEE Computer Society, 2001.

[Whi96] S.Whittaker and C.Sidner, “Email overload: Exploring personal information management of email,” Proc of ACM CHI 96, pp.276-283, 1996.