EmailMap: Visualizing Event Evolution and Contact Interaction within Email Archives

Sheng-Jie Luo* National Taiwan University
Li-Ting Huang* National Taiwan University
Bing-Yu Chen† National Taiwan University
Han-Wei Shen‡ The Ohio State University

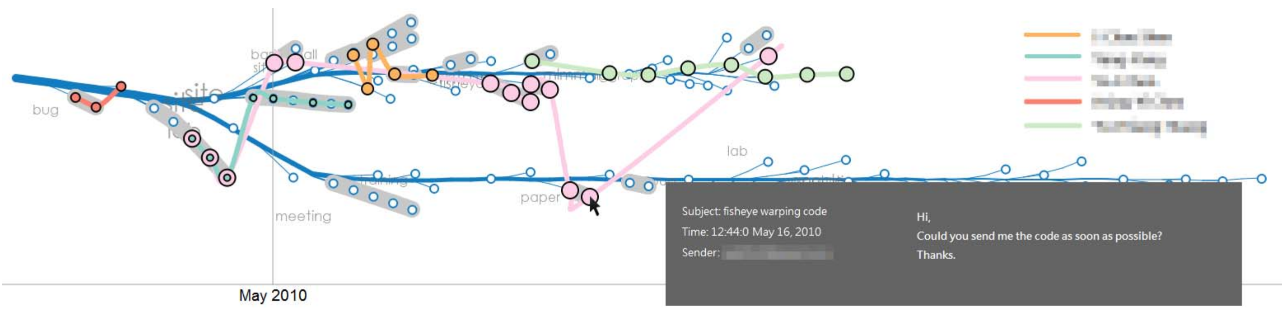

Figure 1: An example of EmailMap. The blue color flow depicts the event evolution in email archives of a time period, and the color tracks reveal the interaction between the email owner and his/her contacts. Notice that the names are blurred due to the privacy issue.

Abstract

Email archives contain rich information about how we interact with different contacts and how events evolve throughout time. Making sense of the archived messages can be a good way to understand how things evolved and progressed in the past. Although much work has been devoted to email visualization, most work has focused on presenting one of the two aspects of email archives: discovering the evolution of emails and events, or the relationship between the email owner and his/her contacts over time. In this paper, we present EmailMap, an email visualization which integrates the information of both events and contacts into a single view, enabling users to make sense of their email archives with complementary contextual information. Two visualization components are designed to portray complex information within the email archives: event flow and contact tracks. The event flow illustrates the evolution of past events, helping the users to grasp high-level pictures and patterns of their email archives. The contact tracks reveal the interaction between the email owner and his/her contacts.

Index Terms: I.3.8 [Computer Graphics]: Applications—; H.5.2 [Information Interfaces and Presentation]: User Interfaces—

1 Introduction

Email has become one of the most common communication tools in our daily life. As a result, it is very likely that most email users have large email repositories. As a major communication tool, email plays a critical role in our daily activities and our interaction with other people. In addition, email has evolved into a multi-purpose tool used for more than just sending/receiving messages [6, 24].

Our life described in email archives can be understood in two aspects: people and events. People signifies how we interact with our colleagues, friends, and families. Events represent what we encountered or what we have worked on, and how our life stories unfold over time. Previous email archives visualizations usually focus only on presenting one of the two aspects: the relationship of people (i.e., contacts) [3, 5, 7, 12, 14, 20, 22, 23], or the events that the email archives presented [8, 10, 16, 17, 19, 21]. Visualizations focus on people portray the relationship between the contacts over time, and visualizations focus on events display the relationship between the email threads, and how different email threads emerge and change. With these visualizations, users can only trace either the aspect of people, or the aspect of events, but not both. However, these two aspects are often interwoven tightly in our life story. Thus, providing the integration of both aspects is a better way to depict our life recorded in the email archives.

In this paper, we present EmailMap, a visualization that helps users associate the relationship of people and the relationship of events. We design a new way to understand personal email archives by integrating the contact relationship and email relationship into a single view, in which one relationship is used as the context providing more information to the other. Specifically, emails are grouped into a set of hierarchical events, and illustrated as an event flow over the temporal coordinate. Related email messages can be browsed along the same branch of the flow. In addition, each contact is visualized as a smooth color-coded track that connects all the emails related to the email owner. With the aid of this visualization, users are able to explore both the longitudinal relationship with their contacts as well as the relationship of emails/events at the same time, thus getting a more holistic understanding of the life stories lain in their email archives. Our work has the following contributions. (1) **An visualization system with an integrated event flow and contract tracks view** that depicts the evolution of past life events and the interaction patterns between people. (2) **An email clustering method** that groups a large amount of email threads in the email archives into meaningful chronological life events according to the email content and contacts. (3) **An optimization-based layout algorithm** that computes a smooth event flow out of large-scale email archive data.

2 EmailMap Design

In EmailMap, we were mainly interested in integrating the relationship of people (i.e., contacts) and the evolution of events lain in personal email archives. By providing these two aspects in one sin-

*e-mail: {forestking,soidid}@cmlab.csie.ntu.edu.tw
†e-mail:robin@ntu.edu.tw
‡e-mail:hwshen@cse.ohio-state.edu

978-1-4799-2873-6/14 $31.00 © 2014 IEEE
DOI 10.1109/PacificVis.2014.36

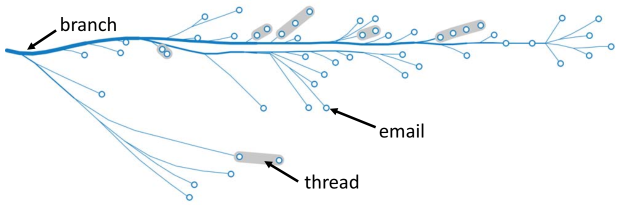

Figure 2: An example of event flow, which consists of three visual elements: branch, email, and thread.

gle view, we hope to reveal interesting patterns of how people and events are interwoven over time.

2.1 Dual Design Focus

To the best of our knowledge, most of the email visualization techniques focus on only one data dimension, either the relationship of people or the relationship of events. There are only a few studies [9, 15] considered both. Therefore, we decide to adopt a dual design focus in EmailMap to facilitate tracking both people and events, granting the owner of the email archives (i.e., the ego) the freedom to choose his/her main goal, and providing the two aspects as complementary information. In EmailMap, emails are grouped as an event flow shown as the background, and the contacts are depicted as curved tracks going through the emails they have participated in. The horizontal axis represents time progressing from left to right.

2.2 Events as Flow

As shown in Figure 2, we represent the events from the email archives as an event flow. An event refers to a set of emails discussing about the same task, activity, or topic. Each email is represented as a circle. The flow (from left to right) goes from the first email to the last, which reveals the evolution and relation of these events.

A straightforward approach to visualize related emails is to adopt the concept of email thread, which is defined as a series of messages sharing the same subject, while the prefixes such as ”Re:” and ”Fw:” are ignored [19]. Because there are often hundreds of threads in an email archive, if we directly visualize these threads, it will be difficult to obtain a high-level view of the data. Therefore, we view each email thread as a basic event component, and group the email threads according to both content similarity and participating-contacts similarity. To provide a better overview, threads are represented as a thick gray line going from the first email to the last of the thread. Thread lines also characterize the event flow, revealing how many different conversations were going on, and whether there are many long conversations or the opposite. The time interval between two emails of a thread indicates whether the thread is under intense discussion or not. The total length of a conversation indicates the duration of a certain subject.

The flow can be split into a number of branches. Each branch shows how one event flow evolves into two or more sub-events, and when this splitting happens. The thickness of a branch encodes the relative importance of the flow. Because it is challenging to accurately measure the importance of each email message, we define the importance as the number of email messages that belong to the current flow. As email archives could easily go up to a large number of messages, adopting a linear mapping of the weights could cause huge differences of the line thickness. The dramatical change in line thickness could lead to unnecessary visual clutter near the major flows. To address this problem, we define the thickness of a flow to be the square root of the importance.

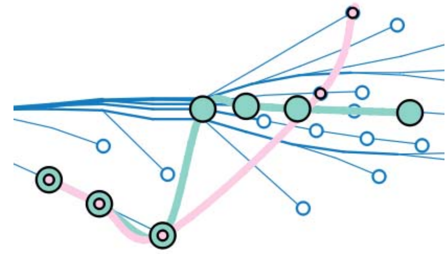

Figure 3: The contacts are represented as curved tracks. When two tracks intersect, the participating node is encoded by concentric rings with their corresponding color keys.

2.3 Contacts as Tracks

To provide the context information, inspired by [1], a contact is shown as a curved track going through the emails that he/she has participated in. Tracing a contact track, we can tell the email exchange frequency between the contacts and ego. Also, by identifying how many different event flow branches this contact traverses, we can see if the ego has contacted with him/her at only one event, or has close interaction across multiple life events. The horizontal range of a contact track indicates the duration of a contact relating to a certain subject. Moreover, the color is utilized to distinguish different contacts. The emails that a contact track traverses are represented as a circular nodes with the color. When two or more contact tracks intersect, the participating node is encoded by concentric rings with their corresponding color keys. Figure 3 shows the interaction of two contacts.

3 Constructing Hierarchical Email Structure

To construct the event flow, similar emails should be grouped together to form a flow for users to trace. As was stated before, email thread is a commonly adopted concept that groups email messages into a series of related messages. Therefore, we first group email messages into email threads, and then apply similarity analysis techniques to group these threads into a hierarchical email structure.

Although general similarity analysis of document content has been well studied in the field of information retrieval, it is not suitable for email messages, which include the contact information such as sender, receivers, and CC (carbon copy) receivers. Simply applying traditional similarity algorithms based on document content would exclude the similarity of contact information. Therefore, we developed a similarity measurement approach that integrates the similarity of content and participating-contacts of the emails, where the former one is calculated using TF-IDF [18] and the latter one is defined as $|N(d_1) \cap N(d_2)|/|N(d_1) \cup N(d_2)|$, where $N(d_i)$ is the set of participating-contacts of message d_i. As a result, the email message similarity is calculated by the weighting combination of the two similarity measures.

Now we describe the grouping technique that incorporates the email message similarity. The event flow is designed to enable users to browse several similar conversations under a high-level event concept. Adopting flat clustering would eliminate the different levels of similarity, which may prevent one from getting useful information in a local view. Therefore, we adopted hierarchical clustering as our basic concept, which enables a dynamic control of final grouping numbers. This flexibility could be used to process email archives with different attributes (with many long or short conversations), providing a minimum-cluttering and meaningful visualization.

Binary tree is a widely-adopted structure to present the structure of hierarchical clustering. However, if we simply apply a binary tree as our clustering structure, we would fail to encode the time information embedded in the emails, which is a prominent feature that should be preserved and well-considered. Considering the time factor, we categorize the grouping of two similar email threads into three conditions: (1) *Temporally overlap:* Two similar email threads with temporally overlap (i.e., the latter one starts while the former one has not ended yet) might indicate that the latter one was triggered by the former one. To encode the possible derived-from relation, we add the latter one as a branch from the former one. In other words, the latter one is visually encoded as an event derived from the former one. (2) *Non-overlap, temporally close:* When two email threads have no overlap and are temporally close, it is relatively vague if one is derived from the other. Therefore, we create a new branch node linked to both of them to depict that the event has split into two sub-events closely related to each other. (3) *Non-overlap, temporally apart:* When two email threads have no overlap and are temporally far away from each other, it might be the case that these two threads are related to another bigger event. Therefore, we connect both threads to another trunk. The second and third cases are distinguished by a time threshold parameter. We set this parameter to 1-day as default.

4 COMPUTING EVENT FLOW LAYOUT

To draw the event flow with smoothly branching lines for aesthetics and readability, we apply an optimization technique to layout the hierarchical email structure obtained from the previous section as a smooth flow over temporal coordinate. To compute the layout, we formulate a number of visual constraints into an objective function, and find the unknown positions of nodes that minimize the function.

Formally, we denote the input hierarchical email structure by $\mathbf{T} = \{\mathbf{V}, \mathbf{E}\}$, where $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_n\}$ is a set of n nodes, $\mathbf{v}_i = (\mathbf{v}_{i,x}, \mathbf{v}_{i,y}) \in \mathbb{R}^2$, and $\mathbf{E}$ is the connecting edges. The event flow has the following four different types of nodes: (1) **Root node** that is the root of the event flow with no parent node. (2) **Email node** which stands for an email message, which has a time stamp to indicate when the email was sent. (3) **Branch node** which has multiple children. (4) **Subdivision node** which has one parent node and one child node, but does not represent an email message. We denote the sets of the above types of nodes by $\mathbf{V}_r, \mathbf{V}_e, \mathbf{V}_b, \mathbf{V}_s$, respectively. In addition, to encode the thickness of the flow, each node has a weight ($w_{\mathbf{v}_i}$) indicating the number of email nodes of the subtree rooted at the node. To obtain a smooth flow layout, we iteratively subdivide an edge and add a subdivision node at the center of the edge if it is longer than a predefined length l_ε.

4.1 Objective Function

We introduce a number of constraints that can capture the properties of a good flow, and compute a set of node positions $\mathbf{V}'$. Specifically, we model the constraints by smoothness cost, occlusion cost, time stamp cost, and flow direction cost.

Smoothness cost. To make the event flow go as smoothly as possible, we encourage the connecting edges to have similar directions. Therefore, we define a smoothness cost for each node which has a parent node and at least one child node. Moreover, if an edge branches into a thicker edge and a thinner edge, it is more pleasing if the thicker edge is straighter, allowing users to easily trace the main branch flows. To implement the idea, we minimize:

$$\Omega_s = \sum_{\mathbf{v}'_i \in \mathbf{V}'_\varepsilon} \sum_{\mathbf{v}'_c \in C(\mathbf{v}'_i)} w_{\mathbf{v}'_c} ||s_{ip}(\mathbf{v}'_i - \mathbf{v}'_{P(\mathbf{v}'_i)}) - s_{ic}(\mathbf{v}'_c - \mathbf{v}'_i)||^2, \quad (1)$$

where $\mathbf{V}'_\varepsilon$ is the set of nodes which have a parent note and at least one child node, $C(\mathbf{v}'_i)$ is the set of $\mathbf{v}'_i$'s children, $P(\mathbf{v}'_i)$ is the parent node of $\mathbf{v}'_i$, $s_{ip} = ||\mathbf{v}_c - \mathbf{v}_i|| / (||\mathbf{v}_c - \mathbf{v}_i|| + ||\mathbf{v}_i - \mathbf{v}_{P(\mathbf{v}_i)}||)$ and $s_{ic} = 1 - s_{ip}$ ensure that the length proportions of neighboring edges do not change during optimization. The weight $w_{\mathbf{v}'_c}$ encourages the thicker branch to be straightened more. Figure 4 shows the comparison of the optimization results with/without the weight $w_{\mathbf{v}'_c}$. Notice that with the weight, the main branch would be straightened more (Figure 4(b)).

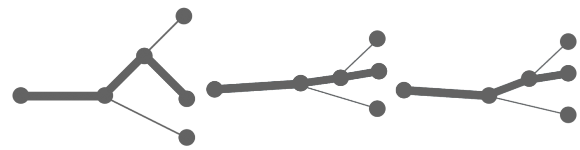
(a) (b) (c)

Figure 4: The results of optimizing the energy function Ω with and without the weight $w_{\mathbf{v}'_c}$. Notice that the thickness of each branch indicates the value of $w_{\mathbf{v}'_c}$. (a) The input tree structure. (b) The result with $w_{\mathbf{v}'_c}$. (c) The result without $w_{\mathbf{v}'_c}$.

Occlusion cost. The email nodes should be prevented from being occluded by other nodes for readability. In addition, it is more visually pleasing if the edges are clearly separated. Therefore, the occlusion cost is designed to ensure that all nodes keep a predefined distance d_ε from other nodes. Specifically, the occlusion cost is defined as:

$$\Omega_o = \sum_{\mathbf{v}'_i \in \mathbf{V}'} \sum_{\mathbf{v}'_j \in \mathbf{V}', \mathbf{v}'_i \neq \mathbf{v}'_j} \Omega_o(\mathbf{v}'_i, \mathbf{v}'_j), \quad (2)$$

where

$$\Omega_o(\mathbf{v}'_i, \mathbf{v}'_j) = \begin{cases} (d_\varepsilon - ||\mathbf{v}'_i - \mathbf{v}'_j||)^2, & \text{if } ||\mathbf{v}'_i - \mathbf{v}'_j|| < d_\varepsilon \\ 0, & \text{otherwise.} \end{cases}$$

Time stamp cost. The email nodes should locate on the position of the temporal coordinate that corresponds to their sent time stamps. Therefore, we penalize the distance between the nodes' x-coordinates and their target x-coordinates. Specifically, the time stamp cost is defined as:

$$\Omega_t = \sum_{\mathbf{v}'_i \in \mathbf{V}'_e} ||\mathbf{v}'_{i,x} - X_{\mathbf{v}_i}||^2, \quad (3)$$

where $X_{\mathbf{v}_i}$ is the target x-coordinates. Notice that the time stamp cost is only used to constrain the email nodes.

Flow direction cost. Event flow was designed to be drawn from left to right. To prevent the flow from bending backward, we enforce the x position of a node to the right of its parent. Thus, we introduce the flow direction cost as:

$$\Omega_f = \sum_{\mathbf{v}'_i \in \{\mathbf{V}'_e, \mathbf{V}'_b, \mathbf{V}'_s\}} \Omega_f(\mathbf{v}'_{i,x}, P(\mathbf{v}'_i)_{,x}), \quad (4)$$

$$\Omega_f(\mathbf{v}'_{i,x}, P(\mathbf{v}'_i)_{,x}) = \begin{cases} 1, & \text{if } \mathbf{v}'_{i,x} \leq P(\mathbf{v}'_i)_{,x} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where $P(\mathbf{v}'_i)_{,x}$ is the x-coordinate of $\mathbf{v}'_i$'s parent node $P(\mathbf{v}'_i)$.

The total objective function for the optimization is a weighted sum of the cost terms defined above: $\Omega = w_s\Omega_s + w_o\Omega_o + w_t\Omega_t + w_f\Omega_f$. We weight the time stamp cost and flow direction cost more strongly compared to the smoothness cost and occlusion cost because they aim to mimic hard constraints. The weights are determined by experimenting with different values and inspecting the results. Although a large ranges of weights work well, we used the weights of $w_s = 1, w_o = 10, w_t = 100$, and $w_f = 100$ for generating all results.

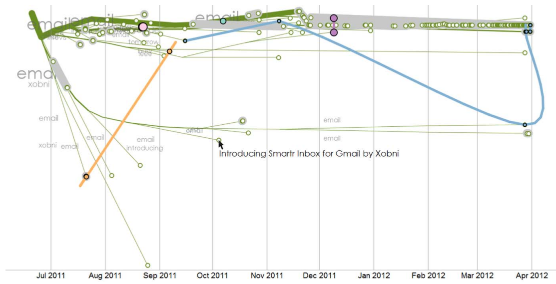

Figure 5: The visualization of EmailMap project.

4.2 Optimization

In this section, we describe how we minimize the objective function Ω to solve for the positions of all the nodes, say $\mathbf{V}'$. The steepest descent method is applied to minimize the objective function, which iteratively moves the positions with a lower energy. Formally, at each iteration the nodes' positions are updated as $\mathbf{V}'_{t+1} = \mathbf{V}'_t - \varepsilon\Delta\Omega(\mathbf{V}'_t)$, where ε scales the step of the gradient vector $\Delta\Omega$. To find an adequate ε, a step-doubling line search strategy is adopted. Starting from the point in $\mathbb{R}^{2n}$ defined by $\mathbf{V}'_{i-1}$, it takes steps along the gradient direction, and doubling step length until the objective function does not decrease, then choose the one with lowest energy.

Initial layout. The iterative optimization requires an initial guess. In our proof-of-concept implementation, we generate the initial layout by adopting a rule-based strategy. The rules capture a number of aesthetic criteria as follows. First, thick branches are expected to contain more email messages, which should be clearly separated and displayed. To achieve this, we set initial y coordinate according to each branch's weight. Second, as email threads are regarded as the basic component of the event flow, messages belong to the same email threads will be given the same y-axis coordinate to maintain their closeness in space. The spacial closeness makes the threads easier to be tracked, as well as avoid visual cluttering. Third, branch nodes, which indicate the split of events, should be easy to identify. Therefore, the children nodes (which represent the sub-event paths) should avoid to have the same y-coordinate as their parent node (which represent a major event.) Finally, as we aim to preserve the temporal information of email messages, the x-coordinate of each email node is fixed at the corresponding time.

Dealing with large-scale email data. Users usually have a large amount of email messages over a long period of time in their archives. As a consequence, we can expect an event flow with a large amount of nodes to optimize. However, to solve a large amount of nodes' positions is technically intractable and inefficient. Hence, we introduce two strategies to deal with this problem: coarse-to-fine optimization and sliding window optimization.

The goal of the first strategy is to solve a rough high-level event flow layout by optimization, followed by subdividing the input hierarchical email structure to get a finer structure and computing its initial layout. Second, the input hierarchical email structure may contain hundreds of nodes and thus is inefficient even the coarse-to-fine strategy is applied. Therefore, rather than globally solving the optimization problem, we propose the sliding window optimization strategy. Specifically, starting from a certain temporal coordinate (e.g., the root node), we compute the energy only for the nodes located in a local temporal window and then shift the window forward and backward. The window size is inversely proportional to the subdivision level of the event flow. The two strategies are applied together to optimize an input hierarchical email structure.

5 Results

Figure 5 shows the visualization of all the mails related to the development of the EmailMap project. As can be seen from the figure, this project started around June, 2011 and completed in the end of March, 2012. The frequency of message exchange has increased since mid-November, 2011. A dramatic rise was found at the end of March, 2012. This project was clustered into two main flows: the upper one contains the discussion of the design and implementation of EmailMap overtime, while the lower one includes the discussions of noteworthy related works. For example, the hovered on message was sent when we were trying out Xoboni [1]. Several participants were highlighted. The blue contact track indicates that this person was involved for a relative longer period of time compared to the orange one. Both of them were professors who gave us advice from time to time, but did not continue to participate in the work later on. The pink, cyan, and purple contacts were other researchers we consulted or who provided us some related information.

6 Limitation, Conclusion and Future Work

While much work has been investigated on email classification [2, 4, 11, 13], it is almost impossible to construct a perfect email clustering method that can match every user's mental model of how he/she understands and classifies his/her email archives. When making sense of email archives, people might also utilize memories that lie outside the email archives to create a rich context information that helps to better cluster the email messages.

We identified the following future research directions. First, a more sophisticated email clustering algorithm is needed. There is still much to be done on understanding how users mentally cluster their email archives into not only a flat classification, but also, an evolutionary structure that can be mapped to real life experiences and memories. We are also searching for ways to integrate more details of the dyad relationship (between contacts and ego) in the visualization, and at the same time, keeping the design intuitive and easily understandable. Other layout algorithms could also be provided to improve both the aesthetics and computational efficiency.

As email plays a prominent role in people's communication and collaboration, it contains rich information for reminiscing and understanding of the past. However, most tools aiming at presenting email archives can provide only one of the two aspects important to the understanding of emails, preventing people from getting a structural comprehension of the closely related evolution of both events and the interaction between people. In this paper, we integrate the two important aspects of email archives into a single visualization. By integrating the event evolution and the interaction between people throughout time, users are enabled to make sense of their own data with complementary context information. By offering a novel approach of making sense of email archives, not only do we provide a new step for reminiscing and understanding the overall pattern of email archives, but also raise awareness for the difficult task of integrating the various information that lies in email archives.

Acknowledgements: The authors wish to thank the anonymous reviewers for their detailed comments. This work was supported in part by the National Science Council of Taiwan under grant NSC100-2628-E-002-036-MY3, and also by the Excellent Research Projects of the National Taiwan University under grant NTU102R7725.

References

[1] B. Alper, N. H. Riche, G. Ramos, and M. Czerwinski. Design study of lineSets, a novel set visualization technique. *IEEE TVCG*, 17(12):2259–2267, 2011.

[1] http://www.xobni.com/

[2] R. Bekkerman, A. Mccallum, and G. Huang. Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. Technical Report IR-418, Univ. of Massachusetts, 2004.

[3] R. P. Biuk-Aghai. Visualization of interactions in an online collaboration environment. In *Proc. CTS '05*, pages 228–235, 2005.

[4] V. R. Carvalho and W. W. Cohen. On the collective classification of email "speech acts". In *Proc. ACM SIGIR '05*, pages 345–352, 2005.

[5] J. S. Donath. Visual who: animating the affinities and activities of an electronic community. In *Proc. ACM MM '95*, pages 99–107, 1995.

[6] N. Ducheneaut and V. Bellotti. E-mail as habitat: an exploration of embedded personal information management. *interactions*, 8(5):30–38, 2001.

[7] D. Fisher and P. Dourish. Social and temporal structures in everyday collaboration. In *Proc. ACM CHI '04*, pages 551–558, 2004.

[8] S. Frau, J. C. Roberts, and N. Boukhelifa. Dynamic coordinated email visualization. In *Proc. WSCG '05*, pages 187–193, 2005.

[9] S. Hangal, M. S. Lam, and J. Heer. MUSE: reviving memories using email archives. In *Proc. ACM UIST '11*, pages 75–84, 2011.

[10] B. Kerr. Thread arcs: an email thread visualization. In *Proc. IEEE InfoVis '03*, pages 211–218, 2003.

[11] A. Leuski. Email is a stage: discovering people roles from email archives. In *Proc. ACM SIGIR '04*, pages 502–503, 2004.

[12] M. Mandic and A. Kerne. Using intimacy, chronology and zooming to visualize rhythms in email experience. In *Ext. Abst. ACM CHI '05*, pages 1617–1620, 2005.

[13] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual search and name disambiguation in email using graphs. In *Proc. ACM SIGIR '06*, pages 27–34, 2006.

[14] A. Perer, B. Shneiderman, and D. W. Oard. Using rhythms of relationships to understand e-mail archives. *JASIST*, 57(14):1936–1948, 2006.

[15] A. Perer and M. A. Smith. Contrasting portraits of email practices: visual approaches to reflection and analysis. In *Proc. AVI '06*, pages 389–395, 2006.

[16] S. Rohall, D. Gruen, P. Moody, and S. Kellerman. Email visualizations to aid communications. In *Posters IEEE InfoVis '01*, 2001.

[17] A. Saad and R. Dimitrios. Email threads: A comparative evaluation of textual, graphical and multimodal approaches. *IJ Comp.*, 3(2):238–250, 2009.

[18] G. Salton, editor. *Automatic text processing. Section 10-1, The Vector Space Model*. Addison-Wesley, 1988.

[19] S. Sudarsky and R. Hjelsvold. Visualizing electronic mail. In *Proc. IV '02*, pages 3–9, 2002.

[20] J. R. Tyler and J. C. Tang. When can i expect an email response? a study of rhythms in email usage. In *Proc. ECSCW '03*, pages 239–258, 2003.

[21] G. D. Venolia and C. Neustaedter. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *Proc. ACM CHI '03*, pages 361–368, 2003.

[22] F. B. Viégas, D. Boyd, D. H. Nguyen, J. Potter, and J. Donath. Digital artifacts for remembering and storytelling: Posthistory and social network fragments. In *Proc. HICSS '04*, volume 4, pages 40109.1–40109.10, 2004.

[23] F. B. Viégas, S. Golder, and J. Donath. Visualizing email content: portraying relationships from conversational histories. In *Prof. ACM CHI '06*, pages 979–988, 2006.

[24] S. Whittaker and C. Sidner. Email overload: exploring personal information management of email. In *Prof. ACM CHI '96*, pages 276–283, 1996.