電子郵件資料中事件演進模式與人際互動關係之視覺化研究

黃莉婷* 羅聖傑* 陳炳宇†
國立臺灣大學
*{soidid, forestking}@cmlab.csie.ntu.edu.tw †robin@ntu.edu.tw

ABSTRACT
電子郵件資料中有大量關於人們如何和連絡人互動，以及生活中的事件如何演進、發展的資訊。理解與組織這些資料可以幫助人們了解並且綜觀過去的生命體驗。儘管過去已有不少研究著力於電子郵件的視覺化，但大部分的研究主題仍集中在下列兩者之一：理解電子郵件中的事件發展；或是人們之間互動關係的變化，尚未有研究能提出完整的方式整合這兩項資訊。在本研究中，我們提出了 EmailMap — 一個將電子郵件視覺化的系統。在此系統中，我們整合事件發展和連絡人互動關係的資訊至一個單一的系統介面，讓使用者能夠透過這兩個互補的資訊，充分地理解隱藏在大量電子郵件資料中的脈絡和資訊。我們設計了兩種視覺化元素來呈現這些資料：事件流（event flow）和聯絡人追蹤（contact tracks）：事件流以流線方式呈現過去事件之演變，幫助使用者理解資料較為全觀性的特徵與結構；聯絡人追蹤則用以呈現人們之間互動的情況。最後，我們透過質性使用者測試，驗證了 EmailMap 系統之有效性，說明了此一系統能夠幫助使用者整合過去事件之演變與人們互動關係之變化，提供更豐富的回想經驗。

Categories and Subject Descriptors
K.6.1 [**Management of Computing and Information Systems**]: Project and People Management—*Life Cycle*; K.7.m [**The Computing Profession**]: Miscellaneous—*Ethics*

General Terms
Design

Keywords
電子郵件視覺化、電子郵件資料、事件發展、聯絡人互動、分類、事件流、聯絡人追蹤

1. INTRODUCTION
Email is one of the most popular applications in our daily life[1]. With many free email services offering increasing storage and decreasing cost, email users tend to save most of messages they received [5]. Therefore, it is very likely that most email users have large email repositories as time goes by. As a major communication tool, email plays a critical role in our daily activities and our interaction with other people. No matter it is a research team with overseas members working closely on a project, a company announcing new policies, an international coordinator drafting a new cooperating program, or a meeting organizer sending meeting appointments, it is not unusual to find email as the prominent tool in delivering the messages. In addition, email has evolved into a multipurpose tool used for more than just sending messages [32] [8].

[1]http://www.radicati.com/?p=7261

With its diversified usages, email has become a passive life-logging medium. Unlike keeping a blog or a diary, email has recorded our life without people adding additional efforts. Therefore, to understand email archives can be a good way to understand how things evolved and progressed in the past. However, to traverse thousands of emails can be tedious. It is also difficult to make sense of large amount of data. As present email client does not offer an efficient solution, it is crucial to provide a tool that can facilitate making sense of the rich life stories lain in the mess email archives.

Our life (described in email archives) can be understood in two aspects: people and events. People relate to how we interact with our colleagues, friends, and families. Events represent what tasks we have been working on, and how different events evolved during the past. Most previous systems for handling email archives usually focused only on presenting one of the two aspects: the relationship of people (i.e., contacts) [31] [30] [20] [16] [28] [9] [2] [6], or events the email archives presented [22] [24] [29] [27] [14] [22] [10]. Visualizations focused on people portray the relationship between the contacts over time. On the other hand, visualizations focused on events display the relationship between the email threads, and how different email threads emerge and change over time. With these visualizations, users can only trace either the aspect of people, or the aspect of events, but not both of them.

However, these two aspects are often interwoven tightly in our life story. Our interaction with different people often relates to different events. For example, when collaborating on a project, we work closely with the project members. Moreover, when reminiscing about the past, the relationship of people and the relationship of events can offer complementary hints. For example, we can remember more details of an event by remembering how we interacted with the people involved, and vice versa. Thus, providing the integration of both aspects may be a better way to describe our life described in the email archives.

Therefore, in this paper, we present EmailMap, a visualization that helps users associate the relationship of people and the relationship

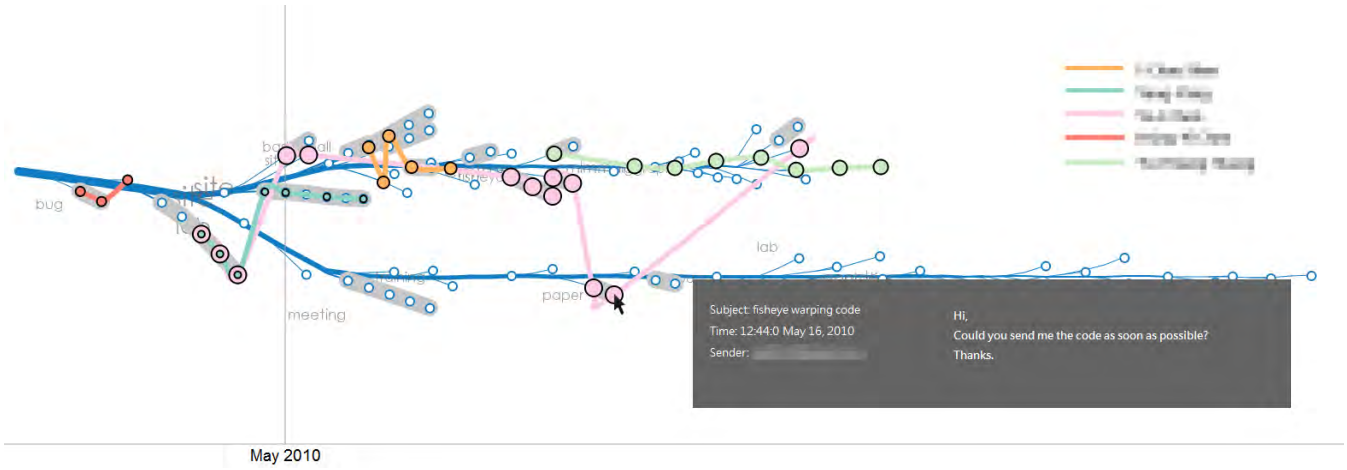

Figure 1: An example of EmailMap of a selected personal email data, the blue color flow depicts the event evolution in email archives of a time period. The color tracks reveals the interaction between these contacts and ego. Notice that the names are unmarked for privacy issue.

of events. We purpose 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 a 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 him/her. 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. To better understand how the visualization facilitates users' comprehension toward their email archives, we conducted a qualitative user study assessing the information it provides to users. Results show that EmailMap effectively helps users to reminisce the details of past cooperation and the evolution of life events. It also enables users to get an overall picture and pattern of their email archives that was unclear to them before. Specifically, our work presents the following contributions.

- **An visualization system with event flow and contact tracks** that facilitates the interactive exploration of email archives with an integral context information depicting the evolution of past life events and the interaction patterns between people.
- **An email clustering method** that groups massive email threads in the email archives into meaningful chronological life events according to email content and participated contacts.
- **An optimization-based layout algorithm** that computes smooth event flow out of large-scale email archive data.

2. RELATED WORK

Email Visualization. Several studies have explored visualizing email archives, however, they have mainly focused on either the relationship of people [31] [30] [20] [16] [28] [9] [2] [6], or the presentation of events/email threads [22] [24] [29] [27] [14] [22] [10]. Visualizations focused on people have aimed to portray the relationship between people (i.e., contacts) over time. On the other hand, visualizations focused on events/emails have displayed the evolution of events, or the relationship between emails.

Themail [31] showed the dyad relationship by visualizing keywords that characterize one correspondence with each contact the best. With the aid of text analysis, the content of email archives summarized the dyad relationship. By scanning the keywords, one can get a general idea of how the relationship has changed over time without going through the haystack. PostHistory [30], an ego-centric visualization of email archives used a calendar panel and a contacts panel in its interface. The contacts panel showed the overall importance of the contacts. Once a contact was selected in the contacts panel, the corresponding emails that were sent by this person in the calendar panel were highlighted, revealing the email change frequency between the ego and the selected contact.

Some other studies have investigated on email rhythms. Perer *et al.* [20] analyzed the temporal rhythms, revealing how people interacted with their contacts. Mandic and Kerne [16] visualized the chronological intimacy pattern by using color and shape to characterize the intimacy level of each email. Users are enabled to see how they spend time and energy on people with different intimacy throughout time. Tyler and Tang [28] focused on the temporal pattern of email usage, enabling users to understand their contacts' response pattern. Xobni [2] and Rapportive [3] provide widgets that show integrated information of users' contacts. In addition to the one-one relationship, several research has devoted to explore the overall interaction patterns lain in collaboration [9] [2] [6], as well as finding a key person among the long contact list [19] [21]. To facilitate managing the contacts, research such as MUSE [11] and ContactMap [18] has provided an automatic grouping of similar contacts.

While the aforementioned visualizations have provided various ways to understand the relationship between people, one cannot know how events have evolved and how different email threads related to each other. Many work has been proposed to depict the relationship of email threads [22] [24] [29] [27]. ThreadArc [14] adopted a tree visualization technique to display the relationship between emails while maintaining chronology. Related emails were connected by arcs, showing the reply-to relationship as well as how different threads evolved over time. From this visualization, one could observe and compare different threads according to their qualities, such as the size of the thread or the number of responses per mes-

[2]http://www.xobni.com/
[3]http://rapportive.com/

sage. With visualizations focused on emails, the relationship between the emails can be easily understood. Nevertheless, one has no clue to the relationship between contacts. Rohall *et al.* [22] proposed a reduced-resolution document overview to help users locate the events they are searching for. Frau *et al.* [10] designed a temporal-plot of emails, enabling users to observe the trends of emails over time and perceive the emails with similar features.

In spite of much having been done in visualizing email archives, with previous visualizations, people could only trace either the relationship of people or the events the email archives presented at a time, but not both of them. Thus, a better visualization that enables people to trace both of the two aspects at the same time is needed.

Some work has proposed similar idea as ours: bring the relationship of people and the evolution of events together. In MUSE [11], different cues were provided to guide users to explore their own email data. Group cues, name cues, and sentiment cues can be used to remind users both their relationship with others and some life events. However, while MUSE focused on providing useful cues, they did not visualize how people and events interwove together clearly. Kang *et al.*presented NetLens [13], a system aimed to help people making sense of huge data by providing both identity and topicality. While they had similar concept as ours, NetLens was more of a query-based system which employed the people panel (represented identity) and the message panel (represented topicality) as separated views, which differ from our integral visualization design that shows both aspects in one single view.

Except for the work that has been done in the field of email visualization, some work from related fields also has proposed the similar concept. Smith and Fiore [26] illustrated the structure of discussion threads in newsgroups, and encoded people with different roles with different glyphs representation. However, one cannot see how the threads of different discussions related to each other. Moreover, while the members of a topic in a newsgroup will always get each reply thread, email works differently. The recipients could change as email threads go on. For example, one might forward a group message he/she got to another people who did not get the same message. Zhu and Chen [33] presented a communication-garden system, which characterized each thread with its post, participants, and duration as petals, leaves, and flowers. They also use the similar technique to decorate each person with his/her posts, topic-participated in, and the time that he/she has joined the topic. While it provided (event) threads and people as complementary information, they did not integrated it in one single view.

Timeline-based Visualization.

As previous work has identified the importance of temporal information in emails [22] [32], we also choose to integral people and event information into a timeline-based visualization. Themeriver [12] used a flow-like visualization to depict the topics changed over time. Textflow [4] extended the similar idea, and improved on showing the merge and split of topics. Dork *et al.* [7] developed a system of following and exploring large-scale online conversations. Rose *et al.* [23] presented a system which linked essential content from streaming data, showing how the document changed as the story developed. However, these systems have focused on revealing the topic evolution from mass text data, and did not take the relationship between people into consideration.

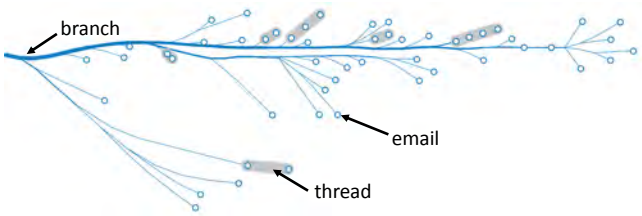

Figure 2: An example of event flow, which consists of three visual elements: branch, email, and thread.

3. 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 single view, we hope to reveal interesting patterns of how people and events interwoven over time, such as:

- How different people played parts in different events? Did they devote much time and energy in one single event, or did they participate in several events?
- How contacts interacted in events? Did they join the same event(s) as co-workers, or did they never "encounter" each other in emails?
- What are the rhythms of interaction within different dyadic relationship with ego (the owner of the email archives)?
- What is the overview of the email archives? How different events evolve and change over time? What is the time span and when is the busy time of email exchange?

3.1 Dual Design Focus

To the best of our knowledge, most of the email visualization designs focused on only one data dimension, either the relationship of people or the relationship of events. In our first design iteration, we explored the pros and cons of only using the contacts or the email threads as our design focus. It turned out that either of them could only achieve half of our goal. By using the contacts as a design focus it was easier to trace people and the relationship over time but difficult for telling the evolution of events. On the contrary, by using the email threads as a design focus had the opposite effect.

Therefore, we decide to adopt a dual design focus in EmailMap to facilitate tracing either people or events, granting the owner of the email archives (i.e., the ego) the freedom to choose his/her main focus, and providing the two aspects as complementary context information. In EmailMap, emails are grouped as a event flow shown in the back, 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.

3.2 Events as Flow

As shown in Fig. 2, we represent the events lain in email archives as an event flow. Each email is represented as a circle in it. The flow (from left to right) goes from the first email to the last, which reveals the evolution and relation of these events in the past.

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, where the prefixes such as "Re:"

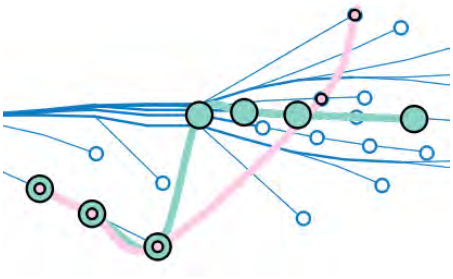

Figure 3: The contacts are represented as curved tracks. When two tracks intersect, the participating node is encoded by concentric rings with their corresponding color keys.

and "Fw:" are ignored [27]. However, there are often hundreds of threads in email archives. If we directly visualize these threads, it will be difficult for users to make sense and get a high level concept 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 participated-contact similarity. To provide a better overview, threads are represented as a 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 more long conversations or the opposite. The intervals in between each email thread show how a given conversation goes with time – if it is an intensive one or a loose one. 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 evolved into two or more sub-events, and when this splitting happened. 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 either lead to the unnecessary clutter near the major flows. To address this problem, we define the thickness of a flow to be square root of the importance.

In order to make the flow more comprehensive, keyword hints can be adopted as road signs that guide users during the trace of events. An effective way to visualize keywords is to place word clouds next to important branches. However, it is not suitable in our scenario because branches would be near to each other, and thus cause visual clutter. Therefore, we extract the important keywords from the email archives, and then lay the most important keyword next to the branch. To prevent from cluttering, we only lay the keyword when a branch is important, i.e., the weight of a branch decreased compared to it's previous edge and the weight exceeds a predefined threshold value, which is allowed users to adjust. In addition, the keywords are laid as a gray color word in the background for not being clutter. Their sizes are proportional to the branches' weights for users to be able to trace the main branches.

3.3 Contacts as Tracks

In order to provide the context information of contacts, a contact is showed as a curved track going through the emails that he/she has participated in. Tracing a contact track, we can tell the email exchange rhythms 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 the contact at only one event, or has intense the contact across different life events. The total length 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. Fig. 3 shows the interaction of two contacts.

3.4 Interaction

Contact Interaction. In email archives, it is common to see hundreds of contacts. However, not all contacts stand the same importance. To minimize visual clustering, contact tracks are shown in semi-transparent, and users can control the contact panel to decide which contact tracks to be displayed.

Smart Zooming. Email archives often spend several years. To facilitate a holistic overview as well as a more detail local view, zooming and panning are provided. When zooming, only the x-axis which mapped the interval of real time data are adjusted. By keeping the y-axis fixed, it prevented the event flow pattern becomes too crowded to see.

Hovering. Hovering on an email node displays the content of the email. This enables users to make connections between the event flow structure with their email message memory and experiences. Event flow serves as a structure hint that helps users to organize massive email archives, while each single email message serves as detail description which enriches the event flow.

4. EMAIL PROCESSING

In this section, we describe the algorithms used to process the email archives for visualization.

4.1 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 message. Therefore, we first group email messages into email threads, and then apply similarity analysis technique to group these threads into a hierarchical email structure.

Although general document content similarity analysis 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 document content similarity algorithms would loss the similarity of contact information. Therefore, we developed a similarity measurement approach that integrates the similarity of content and participated-contacts of the emails.

Content similarity. To calculate content similarity $S_{content}$, we use the widely adopted Salton's TFIDF algorithm [25], which determines the weights of feature words of a document by its relative frequency in the corpus.

Participated-contact similarity. Participated-contact similarity measures the level of overlapping contacts between two email mes-

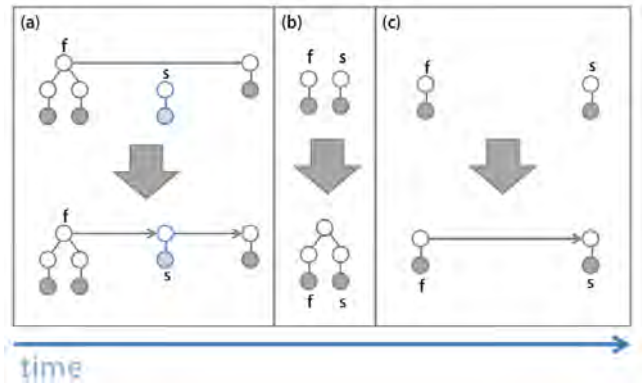

Figure 4: The three clustering conditions, (a) temporally overlap, (b) non-overlap, temporally close to each other, and (c) non-overlap, temporally far away from each other.

sages. The number of contacts shown in the two messages are used as denominator, whereas the number of appearance of the contacts shown in both the two messages are used as the molecular. For example, if contacts A, C, and E are shown in message 1, and contacts A, and B are shown in message 2, the participated-contact similarity will be 2/5. The formula is described as:

$$S_{contact}(d_1,d_2) = \frac{|P(d_1) \cap P(d_2)|}{|P(d_1)| + |P(d_2)|}, \quad (1)$$

where $P(d_i)$ is the number of participated contacts of message d_i.

Email message similarity is calculated by the weighting combination of the two similarity measures as:

$$S_{email} = (1-w)S_{content} + wS_{contact}. \quad (2)$$

In our implementation, we use $w = 0.5$.

Hierarchical email structure. Then, we describe the grouping technique that incorporate the email message similarity. The event flow is designed to enable users to browse several similar conversations as the components of a high-level event concept. Adopting flat clustering would eliminate the different levels of similarity, which could provide useful information in a local view. Therefore, we adopted hierarchical clustering as our basic concept, which enables the dynamic control of final grouping numbers. This flexibility could be used to adapt email archives with different attributes (with many long or short conversations), providing a cluttering-minimized and meaning-preserved 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 data of emails, which is a prominent feature that should be preserved and well-considered. Considering the time factor, we categorize grouping two similar email threads into three conditions: (1) temporally overlap, (2) non-overlap, temporally close to each other, and (3) non-overlap, temporally far away from each other. Fig. 4 depicts the 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 subevents 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.

4.2 Keyword Extraction

In order to generate the keyword hints, a set of keywords are extracted from the email archives. As email subjects usually summarize and give good hints about what the message is about, keywords are extracted from the message subjects. As the event flow grows, keyword lists are updated to its ancestor nodes. It ensures that the keyword list contains in each node describes its descendant properly.

4.3 Contact Processing

The major issues when processing the contacts are that people might have multiple email addresses. As people could have different displaying names of these email addresses (e.g., "Jeremy Lin" and "Shu Hao Lin"), and different people could also have the exactly displaying name, to precisely identify email addresses that belong to the same person is almost impossible.

In EmailMap, we choose to regard each distinct email address as an independent contact flow. The advantages of this design are as follows. First, users are enabled to track how a contact shift from different mail addresses over time. Second, users can still track all the email addresses belong to one contact by clicking on the contact control panel. Third, users will not get confused or misguided by wrongly aggregated email addresses.

5. 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:

- **Root node.** The root of the event flow with no parent node.

(a) (b) (c)

Figure 5: 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 $w_{\mathbf{v}'_c}$.

- **Email node.** The node which stands for an email message, which has time stamps to indicate when the email was sent.
- **Branch node.** The node which has multiple children.
- **Subdivision node.** The node which has one parent node and one child node, but does not stand for 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.

5.1 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 as smoothness cost, occlusion cost, time stamp cost, and flow direction cost.

Smoothness cost. To make the event flow as smooth 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:

$$\Omega_s = \sum_{\mathbf{v}'_i \in \mathbf{V}'_\varepsilon} \sum_{\mathbf{v}'_c \in C(\mathbf{v}'_i)} w_{\mathbf{v}'_c} |s_{ip}(\mathbf{v}'_i - \mathbf{v}'_{P(\mathbf{v}'_i)}) - s_{ic}(\mathbf{v}'_c - \mathbf{v}'_i)|^2, \quad (3)$$

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. Fig. 5 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 (Fig. 5(b)).

Occlusion cost. The email nodes should be prevented from occlusion 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:

$$\Omega_o = \sum_{\mathbf{v}'_i \in \mathbf{V}'} \sum_{\mathbf{v}'_j \in \mathbf{V}', \mathbf{v}'_i \neq \mathbf{v}'_j} \Omega_o(\mathbf{v}'_i, \mathbf{v}'_j), \quad (4)$$

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:

$$\Omega_t = \sum_{\mathbf{v}'_i \in \mathbf{V}'_e} |\mathbf{v}'_{i,x} - X_{\mathbf{v}_i}|^2, \quad (5)$$

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

$$\Omega_f = \sum_{\mathbf{v}'_i \in \{\mathbf{V}'_e, \mathbf{V}'_b, \mathbf{V}'_s\}} |\mathbf{v}'_{i,x} - P(\mathbf{v}'_i)_{,x}|^2, \quad (6)$$

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:

$$\Omega = w_s\Omega_s + w_o\Omega_o + w_t\Omega_t + w_f\Omega_f. \quad (7)$$

We weight the time stamp cost and flow direction cost 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.

5.2 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, and are summarized 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. The more weight it has, the more y-axis space it will be assign. The y-axis space will be past down from parent node to child nodes. Second, as email threads are regarded as the basic component of 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

Figure 6: The visualization of EmailMap project.

closeness make the treads easier to be tracked, as well as avoid visual clustering. 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-axis 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. Specifically, denote by $\mathbf{T}_0$ the input hierarchical email structure, we first apply the optimization to compute $\mathbf{T}'_0$. Then, we subdivide $\mathbf{T}'_0$ by inserting one subdivision node at the center of each edge whose length is longer than the predefined length threshold l_ε, and get a finer structure $\mathbf{T}_1$. We then optimize the finer structure to get $\mathbf{T}'_1$ and iterate the process until no subdivision node is inserted.

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 note), we compute the energy only for the nodes locate 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. In our experiment, the coarse-to-fine strategy works well together with the sliding window strategy for most input data although it is not guaranteed to get a globally optimal solution.

6. RESULTS AND DISCUSSION

6.1 Case Study

Fig. 6 shows the visualization of all the mails related to EmailMap project. As can be seen from it, this project started around June, 2011 and lasted until the end of March, 2012 (the deadline). The

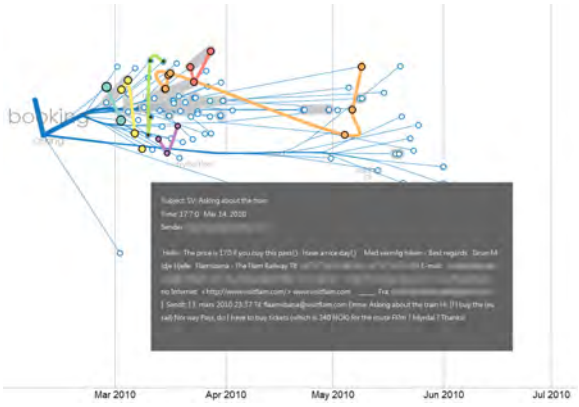

Figure 7: The visualization of planning a trip to Norway.

frequency of message exchanging 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 discussion of some noticible references. For example, the hovered on message was when we were trying out Xoboni [4].

Several participants were highlighted. The blue contact track indicates that this person was involved for a relative longer time compared to the orange one. Both of them were professors who gave us advises from time to time, but not continuously participated in the work. The pink, cyan, and purple contacts were other researchers we consulted or who provided us some related information.

Fig. 7 depicts a Norway travel plan. The most busy time of message exchanging was in March, 2010, when the traveller started to organize the itinerary and book tickets and hostels. Keyword hints such as "booking" and "asking" can be seen from it.

Five short contact tracks represents the booking confirmation email as well as the inquires about transportation, available rooms, and prices. These relatively straight tracks portray the quick responses from the hostels and transportation companies.

It also reveals the traveller's pattern of planing a trip: he/she booked and confirmed most of the things within one month (March, 2010). A smaller email traffic came right before going to Norway (at the end of June, 2010).

The message content is displayed after double clicking on an email node.

Fig. 8 illustrates a massive email archives related to organizing PacificVis. It spans for more than two years, with over 2,215 messages. As can be seen from the visualization, this event was composed of four major flows. The top one was probably related to communication between the coordinators and the invited speaker, and some announcements. The other three were most likely to be the interactions within the coordinators.

The red and the purple contact tracks were only involved in the first

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

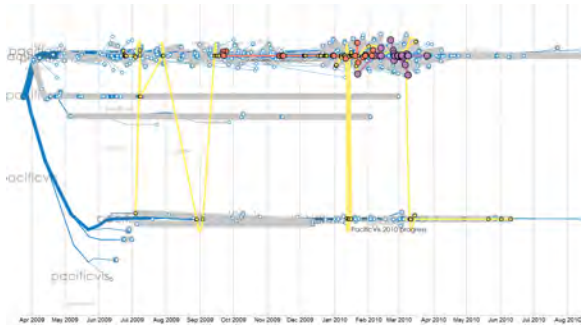

Figure 8: The visualization of massive email archives with over 2,215 messages.

flow, while the yellow one participated in three out of the four main flows, and for longer time.

The messages exchanged between January, 2010 and March, 2009 were more diversified and with various subjects. However, the messages exchanged afterwards shifted to few subjects related to "timeline" or "reminder".

6.2 Evaluation

We conducted a small-scale user study to test and verify our assumptions. As a proof-of-concept system, the users identified some problems, however, they also verified some functionalities of the system. We now report on the study below.

Methodology

In this study, we recruited six participants who were frequent email users and kept a relative long email archives. As a matter of privacy issue, we provided them a Java email parser program, and required them to download their email messages at their own computer. We reminded them to checked if the result file contained any privacy content before giving it to us. In general, the users provided us a total of 200 to 500 email messages that were without privacy concerns. Two of the participants are female, and the participants had an average age of 25. After giving a basic instructions, users were asked to use the system freely for five to ten minutes. We required them to share any comments and feeling during using the system. The users also filled out a simple questionnaire evaluating EmailMap afterwards.

Qualitative results

Generally, the users were pleasure and interested when exploring their email messages in EmailMap with an average rating of 4.33 out of a 5-point Likert scale. We report on more qualitative results below.

Reminiscence. Our users gave EmailMap quite positive feedbacks in terms of reminiscence. Specifically, three aspects of benefits were mentioned: to remind, to reconfirm, and to recollect.

As a tool help people to remind, P3 mentioned: "Actually, I don't need much detail when reminiscencing. However, EmailMap serves as an overview structure of the memories email have recorded. It is nice to have a high-level understanding." P4 also talked about this aspect: "This overview brings me up some memories that I don't come to think of normally." P6 described how EmailMap saved his time to reminisce from his email archives: "I don't think anybody would want to traverse their email one after another. So it's good to have EmailMap to summarize for me."

P3 also talked about how EmailMap can stand as an "evidence" for reconfirming people's memories about past: "Oh, the mails have proved that he asked me to do so much work back them."

EmailMap also helps people remembering some events that had been forgotten. P1: "I thought he didn't give me the answer of that homework. But actually he did!" P5:"I came to remember that we used to have a study group when exploring my email archives with the Contact Tracks. And there was a guy blamed by the others as he didn't contribute enough.")

Structure insight. User have verified our design purpose of providing structure insights of email archives. It helps to understand how the user interact differently with his/her contacts with events as a contextual information. (P1:"I have many emails of this project at that time. Then I can see C joint this event at a point, which reflects that fact that we were team members working on the final project." P4: "It is quite interesting that I can see not only how I interact with one of my contacts, but several of them at the same time! I can compare and see how they involve in different events." P5:"I noticed that there were lots of emails about the stock market, which continued for almost a year. During that time, there were always emails about it on every Mondays. But then we had some argument and stopped contact since.")

P6 had also mentioned that EmailMap helped to identified group messages without remembering the

Contacts with multiple email accounts are display separately also reveal interesting patterns for the users. (P1: I can see that this person has shifted from using hotmail account to gmail account.)

The overall structure also provided new insight that the users had not been aware of before. (P1: "Very interesting! A and the girl he likes only intersected at those email related to going out for fun. He was into her, and that's why he asked her to join whenever other friends were planning to gather and have fun.)

Orientation. Even though EmailMap was not designed for searching, one user had mention it helps to locate the messages being looked for. P4: For those contacts I frequently contact with, it is difficult to locate a certain one only by giving the contact's name as a search key. As I have many email messages going on with these persons, the resulting lists are always too long too look into. By using EmailMap, it is much easier to find what I was looking for with the help of context information including other contacts, other events, and the temporal patterns.

Other feedback. One user mentioned about the clustering was not good enough to help him make sense of his email archives. (P3: "I can't tell how my emails related to each other only from the event flows. It's not well-classified enough.) As the data of P3 were mostly composed of single message without longer threads or major events detected, the relations between emails were not obvious and thus made the result visualization not as helpful.

Our users had also comment on future possibilities of the system. For example, it might classified the contacts in terms of appear-

ance, which would enable the user to identify what are the people that they have only contacted once. Besides, users were also hoping for a multi-combination views which allows them to choose from "contact vs. time", "keywords vs. time", and other possible aspects of email archives. The experience of using EmailMap had provoked our users to imagine about much more dimensions email archives could provide.

To conclude, the users agreed on that EmailMaps offers insight that were inaccessible with general email clients (with an average rating of 4.5 out of a 5-point Likert scale). They also found it helpful for reminiscing of past life, especially the cooperation with others, and how the cooperation changed and evoluted (with an average rating of 4.16 out of a 5-point Likert scale). The average rating of whether EmailMaps is useful is 3.33. However, users mentioned EmailMap is not "useful" as they might not use it everyday, or it might not served as an efficient searching tool. However, they affirmed the usefulness in terms reminisce.

7. LIMITATION AND FUTURE WORK

7.1 Limitation

While much work has been investigated on email classification [15] [3] [17] [1], 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.

Our similarity measure was proved being able to model people's understanding of events at a certain level. However, it might be not as precise in some cases. For example, some people might not classify their email messages purely from the email content and the people involved, but also their subjective judging of importance. An encourage message in need could be much more meaningful than if it was sent on another time. The subjective judging of proper timing, and the personal connection in real life is difficult to measure from only email messages. When the clustering algorithm lost its preciseness, the visual representation (i.e., event flow and contact tracks) comes after would be limited on depicting a meaningful evolution of life events.

The second limitation of this work is the detail presentation of the interaction between the contacts and ego. As we focused on and moved room for portraying an overview structure of how different people related to various life events, the detailed one-one information, such as the most frequent-talked words over time between each contact and ego, was not shown. The current optimization-based layout algorithm is affected by the initial layout. For some extreme cases, the optimization did not give satisfying results.

7.2 Future Work

We identified the following future research directions. First, a more sophisticated email clustering algorithm is needed. There is still much need to be done on understanding how users mentally clustering their email archives into not only a flat classification, but also, an evolutional 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.

8. CONCLUSION

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 on presenting email archives have limited on only one of the two aspects lain in email messages, restricting 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. Our preliminary user study also suggested that this integral visualization has the potential to facilitate email searching when keyword-based filtering method failed to give a smaller-enough set to look over. 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.

9. 致謝

本論文分別感謝國科會與國立臺灣大學經費補助，計畫編號分別為：NSC100-2628-E-002-036-MY3與NTU10R70725。

10. REFERENCES

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

[2] R. P. Biuk-Aghai. Visualization of interactions in an online collaboration environment. In *Proceedings of the 2005 International Conference on Collaborative Technologies and Systems*, pages 228–235, 2005.

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

[4] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. TextFlow: Towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, 2011.

[5] L. A. Dabbish, R. E. Kraut, S. Fussell, and S. Kiesler. Understanding email use: predicting action on a message. In *ACM CHI 2005 Conference Proceedings*, pages 691–700, 2005.

[6] J. S. Donath. Visual who: animating the affinities and activities of an electronic community. In *ACM Multimedia 1995 Conference Proceedings*, pages 99–107, 1995.

[7] M. Dork, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1129–1138, 2010.

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

[9] D. Fisher and P. Dourish. Social and temporal structures in everyday collaboration. In *ACM CHI 2004 Conference Proceedings*, pages 551–558, 2004.

[10] S. Frau, J. C. Roberts, and N. Boukhelifa. Dynamic coordinated email visualization. In *Proceedings of the 13th International Conference on Computer Graphics, Visualization and Computer Vision*, pages 187–193, 2005.

[11] S. Hangal, M. S. Lam, and J. Heer. MUSE: reviving memories using email archives. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 75–84, 2011.

[12] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[13] H. Kang, C. Plaisant, T. Elsayed, and D. W. Oard. Making sense of archived e-mail: Exploring the Enron collection with NetLens. *Journal of the American Society for Information Science and Technology*, 61(4):723–744, 2010.

[14] B. Kerr. Thread arcs: an email thread visualization. In *Proceedings of the 2003 IEEE Symposium on Information Visualization*, pages 211–218, 2003.

[15] A. Leuski. Email is a stage: discovering people roles from email archives. In *ACM SIGIR 2004 Conference Proceedings*, pages 502–503, 2004.

[16] M. Mandic and A. Kerne. Using intimacy, chronology and zooming to visualize rhythms in email experience. In *ACM CHI 2005 Extended Abstracts*, pages 1617–1620, 2005.

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

[18] B. A. Nardi, S. Whittaker, E. Isaacs, M. Creech, J. Johnson, and J. Hainsworth. Integrating communication and information through contactMap. *Communications of the ACM*, 45(4):89–95, 2002.

[19] C. Neustaedter, A. Brush, M. Smith, and D. Fisher. The social network and relationship finder: Social sorting for email triage. In *Proceedings of the 2nd Conference on E-mail and Anti-Spam*, 2005.

[20] A. Perer, B. Shneiderman, and D. W. Oard. Using rhythms of relationships to understand e-mail archives. *Journal of the American Society for Information Science and Technology*, 57(14):1936–1948, 2006.

[21] O. A. Puade and T. G. Wyeld. Visualising collaboration via email: Finding the key players. In *Proceedings of the 10th International Conference on Information Visualization*, pages 124–129, 2006.

[22] S. Rohall, D. Gruen, P. Moody, and S. Kellerman. Email visualizations to aid communications. In *Posters of the 2001 IEEE Symposium on Information Visualization*, 2001.

[23] S. Rose, S. Butner, W. Cowley, M. Gregory, and J. Walker. Describing story evolution from dynamic information streams. In *Proceedings of the 2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 99–106, 2009.

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

[25] G. Salton, editor. *Automatic text processing*. Addison-Wesley, 1988.

[26] M. A. Smith and A. T. Fiore. Visualization components for persistent conversations. In *ACM CHI 2001 Conference Proceedings*, CHI '01, pages 136–143, 2001.

[27] S. Sudarsky and R. Hjelsvold. Visualizing electronic mail. In *Proceedings of the 6th International Conference on Information Visualisation*, pages 3–9, 2002.

[28] J. R. Tyler and J. C. Tang. When can i expect an email response? a study of rhythms in email usage. In *Proceedings of the 8th Conference on European Conference on Computer Supported Cooperative Work*, pages 239–258, 2003.

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

[30] 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 *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, volume 4, pages 40109.1–40109.10, 2004.

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

[32] S. Whittaker and C. Sidner. Email overload: exploring personal information management of email. In *ACM CHI 1996 Conference Proceedings*, pages 276–283, 1996.

[33] B. Zhu and H. Chen. Communication-garden system: Visualizing a computer-mediated communication process. *Decision Support Systems*, 45(4):778–794, 2008.