

Patience for Agility: Designing a Web-based Email Conversation Viewer

Eric Wilcox, Stefan Nusser, Julian Cerruti, Jerald Schoudt, Hernan Badenes

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

{eric_wilcox@us.ibm.com, nusser@us.ibm.com, jcerruti@ar.ibm.com, jschoudt@us.ibm.com, hbadenes@ar.ibm.com}

Abstract

We present a case study describing the process and resulting design for the bluemail conversation viewer: a web-based mashup between social networking information and email threads within a corporate setting. In particular, we look closely at the role design played when adjoined to an agile development process. We discuss the benefits and trade-offs from working in highly iterative and modestly scoped development cycles. In our experience, establishing an equally significant design role served as a valuable means to maintain strategic focus across many parallel development activities. Additionally, we talk about how the nature of agile development and the broad reach of email enabled frequent user input to continuously ignite design direction.

Introduction

Email has been well established as the leading collaborative tool for the modern workplace [4,8]. Web-based email clients in the corporate setting have long taken a backseat to more traditional, platform specific rich clients. Until recently, the level of responsiveness and interactivity demanded in the workplace has been challenging to achieve in a web-based client. The bluemail project explores the feasibility of delivering an enterprise-level email experience in a web browser. On the technical side, we set out to illustrate that common user-based tasks in rich clients such as sorting and filtering large (10,000+) email message collections could attain adequate responsiveness in a web-based client without requiring page refreshes. In terms of user interface design, we aimed to illustrate analogous levels of user interactivity and capability that people have come to expect in richer clients. Reproducing rich client behavior alone was seen as an essential but not sufficient goal for the project. The team also realized that running on a web-application platform opened the door to new possibilities and methods for development that could drive enterprise email into new areas. In this paper we focus on a data and user interface (UI) *mashup*¹ that integrates corporate directory and social networking information into email thread conversations.

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹ Wikipedia defines a mashup as, “a website or application that combines content from more than one source into an integrated experience.”
[http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))

Much research has been conducted on visualizing email threads [1,6,7]. A common theme among these approaches is streamlining thread presentations to elide duplicated content from reply with history functionality. Systems like Gmail™ accomplish this by performing text diffs between messages in a thread and hiding the redundant overlap. This method is extremely effective at minimizing overall text volume and presenting a thread as a single conversation. Our solution builds upon these examples by infusing corporate information such as employee photos and social networking information to further delineate contributions. By doing so, we aimed to provide a more enriched user experience than existing email clients.

It is also worth noting that the often unspoken and latent goal for the project was to provide evidence that ultimately supplied insights and areas for potential business impact to IBM’s product organization. The level and degree of influence we hoped to have could be realized at many levels – from purely strategic product positioning (e.g., should our company be marketing a rich email client application?) to more concrete and tactical deliverables (e.g., can the product organization reuse one particular component or feature(s) from the bluemail prototype?). Apart from directly partnering with the product teams, we set out to grow a significantly sized internal user population that would allow us to validate our solution through usage data and continuous end-user feedback.

Background

The bluemail research project started in Aug 2006. At the time of this publication, usage has been limited to internal IBM employee participants.

Bluemail has been developed by a globally distributed development team. While the project is centrally managed and designed from the IBM Almaden Research Lab in San Jose, California, both lead research developers work remotely from two different time zones – Philadelphia, Pennsylvania, USA and Buenos Aires, Argentina. A three-person contract development team also works from Buenos Aires and San Martin de los Andes, Argentina.

Challenge

The original design brief for the conversation viewer was to create a useful and meaningful mashup between bluemail, corporate directory information, and a concurrently developed social network service in IBM research. Based on trends in commercially available clients such as Gmail™ and Mac OS X Mail, we decided to focus on threaded email conversations as the primary context where we would surface integration. The realization for this brief was constrained by the following factors:

Agile development process: Although we found great flexibility and efficiency in employing such a process, there were drawbacks in terms of constantly limiting the scope for developing the conversation view when balanced against prioritizing other features. This led to a slower design, build, and validate cycle than originally intended.

Time to develop basic threading support: A complete threading architecture where all associated messages in an email thread can be gathered for operations in the UI was not going to be ready in a shortened timeframe to support the idealized design. As a consequence, we based the design on the available content in a single message and later scaled upward to incorporate complete threading content. The team saw this early trade-off as acceptable since many email clients by convention automatically append conversation history when using reply options.

Dependencies on outside services: The social networking service we planned to leverage was being re-factored at the time we were exploring the conversation view in bluemail. Depending heavily on the availability of an outside service, hence, introduced a level of risk since we were forced to balance our design to regard the other project's changing trajectory. Fortunately, the development for this service was folded together with bluemail. This alleviated concerns about aligning the two systems, but ultimately led to more time spent re-factoring the service. Many benefits such as improved UI component reuse were realized as a

result. However, extending the team in such a way shifted focus towards more general architectural issues and away from focusing on the conversation viewer development.

Core email versus new territory: Shifting priorities to illustrate performance and core email functionality to product groups slowed down the development and deployment for the conversation viewer. For the overall project success these trade-offs were vital, however, in terms of really focusing and refining this one capability, the time from design to deployment was longer than other major features passing through the same agile process.

Process

Our development process borrows freely from the Agile family of methodologies [2]. Since bluemail was being developed as a web application, we further tailored our practices to take advantage of how hosted applications are deployed and used. We focused on continuous deployment and rapid exploration of ideas. As illustrated in Figure 1, newly implemented features become robust by moving from sandboxes to more stable instances of our application. Along the way, the internal team and our users exercise emerging features. More stable instances at the end of the development pipeline serve larger audiences. Introducing a bug later in the chain has more severe implications. Thus, update cycles become longer as a change migrates from a sandbox through the development instance and the prerelease instance to our production server.

The agile process has benefited our development team by providing a high degree of transparency. The team and our user community have, at any point in time, complete insight into the status of all development activities.

Project Dynamics

One consequence of the geographically distributed, iterative, and parallel development style we've described is that we found maintaining a coherent direction for the project to become increasingly difficult as more concepts matured. *Strategic design* played a central role in addressing this issue by frequently cutting across sandboxes and by anticipating the integration of concurrently ongoing feature development efforts with the help of storyboards and design mockups [5].

Bi-weekly team meetings further harmonized ongoing team progress by engaging the greater team in discussion around major feature development efforts in corresponding sandboxes. Typically, there was a lead developer for each feature who acted as the owner of the sandbox and leader for discussion. Design partnered with each sandbox leader to set the user experience direction. A resulting, often very loosely defined specification was maintained collectively in a wiki. Both email and instant messaging were used

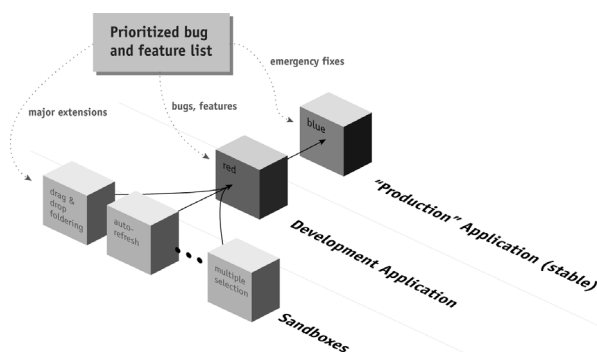


Figure 1: The bluemail agile development process. Major extensions are developed independently in parallel sandboxes. As features mature, they are folded into more widely deployed application instances.

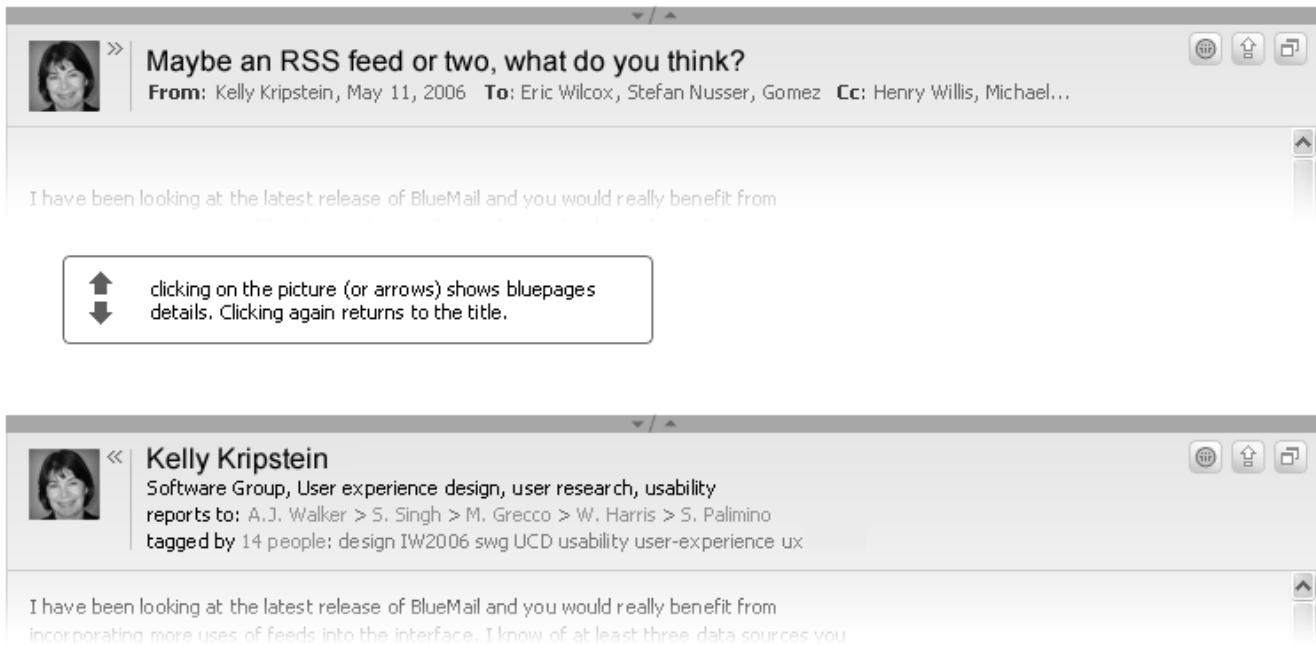


Figure 2: the original mashup design was limited to the message header. Clicking on the small arrows near the image would toggle to view more information on the person. It was determined that this control was too subtle.

heavily for resolving design issues and the wiki was updated frequently to reflect any changes.

Scoping and Development

In the early stages of development many of the team's resources were still working on developing basic email and calendar functionality such as folder support, message composition, and a basic calendar day view. As a result, the beginnings for the conversation viewer mashup started out as a quick means to illustrate viewing of senders' corporately stored photos in the mail message header. We subsequently added a small control that toggled the message header to replace mail information with the sender's profile information (see Figure 2). This feature was added rather quickly to meet the deadline for an upcoming demonstration and the original design proposal was focused on facilitating an easy integration. One downside to the initial solution was that information was only presented for the sender of a message. We found that on many messages we wished to view information for other participants as well.

Once we deployed this functionality to our production site, we started tracking how our greater user population utilized this new capability. Surprisingly, the usage was extremely low with most of the events we captured being caused by our own developers or other people who were close to the project. This gave rise to the realization that the people were simply not aware of the functionality in its current form. People were not aware of the capability to

click on a small icon that toggled the mail header to show additional profile information.

To address these concerns, we decided to completely redesign the user interface for the conversation viewer, with the objective of providing access to profile information for all the main actors in a mail thread. Therefore, we focused on a view that would target and take advantage of full threading support. Following the process outlined in the previous section, a lead developer was assigned and a separate sandbox created for this effort.

As development got underway and a working prototype took form, we interviewed 8 users as they experienced their own email from within the fully functional sandbox. During this study we asked participants to open their inboxes and navigate to a number of recent email conversations to solicit direct feedback on the design. Additionally, we discussed their familiarity with tagging and asked them to envision how they might best use people tagging in the context of email. We inquired how incorporating such functionality might change their email behavior. All interviews were conducted in an informal manner with half taking place over phone and web conference. It's worth noting that the choice in participants was not random. Half of participants we questioned were thought leaders in the product and company's CIO organizations. In addition to gaining the insights from our users, we also aimed to use the interviews as a secondary way to both inform and gather business strategy insights from the participants who represented potential

stakeholders. These conversations often started out by focusing on the task and discrete usage of the bluemail conversation viewer but quickly grew to encompass more strategic concerns. These included discussions about parallel efforts in the company and seeking insights for potential areas impact for the bluemail project in general.

During the interviews, we started to understand that the integration of people tagging with email was a promising direction to explore. Several of our interviewees were active people taggers who were able to conceptually devise ways to leverage people tags within bluemail. However, keeping with our iterative development goal, we concluded that it would be best to build and deploy the basic conversation view first, taking into account their suggestions in a separate round of development. Nevertheless we decided to prominently feature people tags in the user interface in anticipation of future extensions that would leverage this capability.

The core features of the conversation viewer included:

- The view consists of a parsed and flattened reply-chain based on a single-message body. This functionality was a source of uncertainty since we needed to first understand performance and scalability implications of performing this operation, which involves parsing each message in HTML and devising a set of rules and expressions to detect message boundaries.
- People information was extracted from message headers and shown in the form of person tiles that show a picture and provide a limited actions menu. Again, the implications of such a user interface with this complexity were not well understood. We decided to make this feature part of the initial set of changes to face head-on whether we would be approaching such a solution would be feasible.
- Finally, we provided a way for users to readily toggle back to the original unparsed text. The importance of this feature lies primarily in that it provided us a means to statistically judge the adoption of the conversation viewer and allow users to opt to a more traditional view if desired.

As discussed earlier, the actual design envisioned additional steps in the evolution of the interface. Namely, being able to tag people in place using our social network service and being able to filter and view mail by clicking on a person or a tag. Clicking on a tag would aggregate all mail from people associated with that tag. However, these two additional extensions were seen as relying on other core functionality for dynamically filtering the inbox that was not yet under development. We therefore decided to defer the implementation to a later phase.

Solution

Figures 3 and 4 show the conversation view interface in the context of a bluemail message preview. The reply and/or forwarded structure in an email message is flattened and each contribution in the chain is shown as an individual section. Sections are displayed in alternating white/grey background colors (see Figure 3). A global toggle is available to view the unparsed mail message.

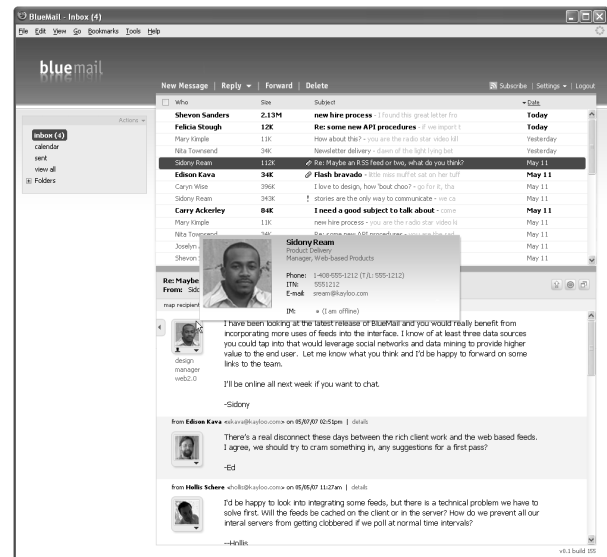


Figure 3: the conversation view. Each response in a reply chain is displayed in sections with alternating background colors. Hovering over a person tile displays a business card containing additional contact and job role information.

The left column for each section contains a person tile (see Figure 4). Each tile represents a grounding point to gain more insight on an individual. If the sender has uploaded a photo to the corporate directory, it is shown in the body of the tile (see Figure 4A). People without photos on file (such as external contacts to the company) are represented by an empty pattern image. Clicking on the photo opens a new browser window showing the sender's profile information if it exists. Hovering the mouse over a tile from an internal mail sender reveals a small business card of corporate directory information including job title, division, phone, email address, and online instant messaging status.

In addition to standard directory information, the tiles also incorporate data from a corporate social networking service named Fringe [3]. If two people have agreed to connect to each other, a small silhouette icon is displayed on the tile (see Figure 4B). One of the key capabilities this service provides is the ability to tag anyone in the company with one or more keywords. In Fringe, these tags are aggregated across all contributors and displayed as an augmented profile page. For example, Steve can tag Sandra as 'blogger' and that tag will be reflected on Sandra's

profile. If more people tag Sandra in a similar manner, a cloud of entries is socially built up on her profile. As a first pass, we chose to limit the conversation view to only surface tags personally applied to a sender. Entries are listed directly beneath the tile (see Figure 4D). Clicking on a tag will open a new browser window into the social networking application that lists all the people tagged with the same keyword across the company. In a future release we plan to change this behavior to filter a user's mail by people matching the clicked on tag.

Each tile also has a dropdown menu (see Figure 4C) that provides an action list. The list includes actions relating to each service in the mashup. One menu item allows a user to start a new. Another navigates to the person's profile. In a later iteration, the user can also add and modify person tags directly from the tile.

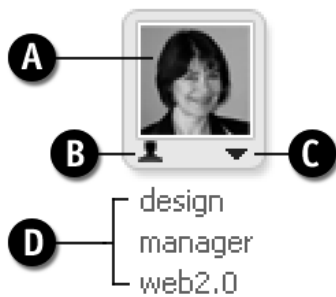


Figure 4: The person tile incorporates information from both the corporate directory and social networking service. A) the photo from a user's corporate profile, B) an icon to represent a mutual connection from this person to the user, C) a dropdown menu to take actions associated to this person, and D) tags personally attributed to this person by the user.

Results

Feedback from the early round of 8 interviews on the conversation viewer was positive. Utilizing photos to quickly scan, parse, and visually weight contributions to the discussion was seen as a valuable addition. One participant noted that our company had reached a point where seeing pictures is becoming second-nature, making an extension like this in mail felt natural. Another participant noted concerns that in certain professional situations, the use of imagery might lead to decreased perception of credibility. He felt that users in authority roles with abstract images (e.g., a picture of their pet, or a cartoon image) might result in employees taking a written message less seriously. Another participant noted that having photos and hover information helped her realize the geographic location for someone she had only interacted with via email – something she saw as increasingly more common in global work environments. This participant recently joined the organization and found the views to be helpful in putting names to faces.

Feedback from the early interviews also saw integrating people tagging as a generally useful addition to the interface. Participants ranged greatly in their experience

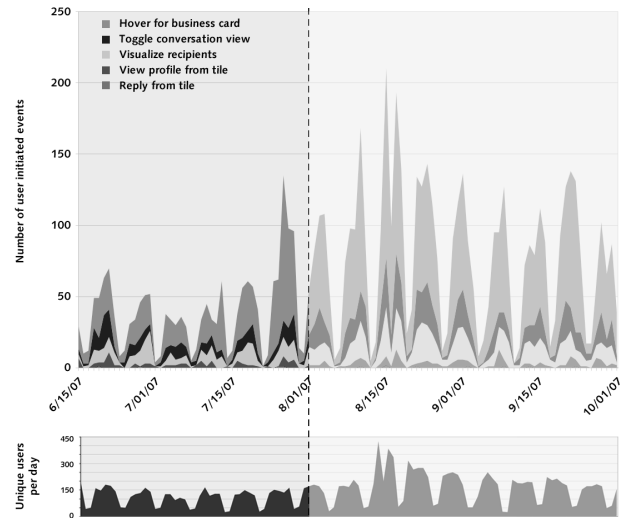


Figure 5: usage data by feature. Each colored segment corresponds to number of clicks recorded across all users during a 3 ½ month period. To the bottom of the graph is a graph of counts corresponding to the number of unique people logging into bluemail each day. The valleys in the data correspond to reduced usage during weekend days.

and expectations for tagging other people. One user was an avid tagger, and used tags in the social network application as a way to group co-workers for later communication. She saw bringing this functionality closer to email as a positive way to cut down on the steps she must now take to cut and paste from the service to compose email. Other participants, however, expressed concerns about publicly sharing tags entered from their private email. They pointed to a need for clear distinctions and capabilities to allow tags to remain private. Even given these concerns, these users felt tagging in this manner would begin to offer more direct personal value as opposed to tagging for purely altruistic purposes in the existing system.

Before deploying the conversation viewer to our production level server, we added an event logging mechanism that allowed us to track user actions taken on the people tiles and associated tags. The graph in Figure 5 summarizes user activity over a 3 ½ month period. A dashed line indicates an initial 3-week period over which we originally intended to evaluate the usage patterns. After this initial period we modified the design to take into account the behaviors noted below. We have included the subsequent usage to show how the initial trends we investigate continued in the subsequent 15 weeks. Taking a closer look at this usage and contrasting against the earlier interviews, we have come to the following realizations:

Take better advantage of the widely used hover – the most used function was the hover-over business card display from the people tiles. In this iteration, we displayed simple

contact and job role information. Moving forward, we feel there is opportunity to design a fully rounded business card that integrates more of the dynamic social networking information such as community attributed people tags.

Users will need explicit onramps to tagging - During the data collection period, only four clicks ever occurred on a tag beneath a person tile. It turned out upon later reflexing that problems with authenticating to the social networking service prevented the majority of users from seeing people tags in the interface. Even once this problem was addressed, people tagging adopting from email has yet to meet our users needs or desires. Future plans include incorporating address book functionality where we hope to revisit the role people tagging can play.

Consider better visualization integration - The link we added that navigates the user to a social network graph visualization in the Fringe application got more usage than we originally anticipated. This had led us to consider lighter-weight ways to surface such visualization directly into bluemail.

Toggle activity: good or bad? - We were unsure how to best interpret the levels of toggle activity between the conversation view and the original message format. Depending on the origin of the message, bluemail is sometimes unable to display the message body, which may have resulted in users toggling to see the original format. To gain a better understanding of this behavior, we have decided to add more data points to our logging capabilities to investigate if there is a common characteristic among the messages toggled.

Discussion

The agile development process we chose to follow provided many benefits to the team and served in a positive way to keep progress moving forward across many complex and intertwined features. The implications for design in comparison to a more traditional waterfall development models were experienced as positive. The lightweight manner and naturally limited scope inherit from working in sandboxes allowed designs to quickly transfer into development hands. Shortened cycles allowed designs to be built and more quickly adjusted based on usage. The freeform flow of a wiki was sufficient for the designer to anchor a solution for development and then work through frequent communication to iterate on details. The main drawback we experience when developing the conversation viewer was that all sandboxes were not necessarily created equally. In the end, the actual solution for the conversation viewer spanned a number of different sandboxes. The process we followed had no formal way to encode this type of progression. Design in this regard served as an overarching integration reminder. There were cases where individual benefits could be seen (e.g., waiting to integrate with more general filtering capabilities) that

ultimately resulted in a more grounded and stable development. Design therefore had the tacit role of strategically tying together parallel work streams. This turned out to require a delicate balance that allowed the team to properly scope and leverage the contracting developers skills without necessarily weighting them down with the more political and strategic decisions.

Our general conclusion is that design and development become more tightly coupled as result from using the agile process. The smaller scale iterations minimized many tensions brought on by more traditional up-front design methods. The one caveat that we offer is that the team must remain patient, even in such a highly iterative setting. Fine-grained functionality was able to steadily evolve, but this does not remove the responsibility of design to maintain a strategic stance and comprehensive project awareness. In hindsight, it's now apparent that the conversation viewer development grew beyond the scope of a single sandbox. As a result, the feature by itself suffered in terms of the amount of time and attention between inception and release, however, the overall grounding and integration with other work streams have led to a more sound architecture. This provides a more scalable solution to build and iterate upon moving forward.

References

- [1] Bellotti, V., Ducheneaut, N., Howard, M., and Smith, I. 2003. Taking email to task: the design and evaluation of a task management centered email tool. Proceedings of CHI 2003 (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003).
- [2] Cockburn, A., Agile Software Development, Addison Wesley, 2002.
- [3] Farrell, S., Lau, T., Wilcox, E., Nusser, S., and Muller, M., Socially augmenting employee profiles with people tagging. UIST 2007 (Newport, Rhode Island, USA, October 7-10, 2007).
- [4] Fisher, D and Moody, P. Studies of Automated Collection of Email Records. University of Irvine, Technical Report, UCI-ISR-02-4, 2001.
- [5] Gabrielli, S. and Zoels, J. Creating imaginable futures: using human-centered design strategies as a foresight tool. Proceedings of the 2003 Conference on Designing For User Experiences (San Francisco, California, June 06 - 07, 2003).
- [6] Gruen, D., Rohall, S. L., Minassian, S., Kerr, B., Moody, P., Stachel, B., Wattenberg, M., and Wilcox, E. Lessons from the reMail prototypes. In Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (Chicago, Illinois, USA, November 06 - 10, 2004).
- [7] Smith, M., Cadiz, J. J., and Burkhalter, B. Conversation trees and threaded chats. In Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (Philadelphia, Pennsylvania, United States).
- [8] Whittaker, S. and Sidner C., Email Overload: Exploring Personal Information Management of Email. Proceedings of CHI 1996 (Vancouver, April 1996), ACM, pp. 276-283.