Facemail:
Showing Faces of Recipients to Prevent Misdirected Email

Eric Lieberman and Robert C. Miller
MIT Computer Science and Artificial Intelligence Lab
32 Vassar Street, Cambridge MA 02139
{ericl, rcm}@csail.mit.edu

ABSTRACT
Users occasionally send email to the wrong recipients – clicking Reply To All instead of Reply, mistyping an email address, or guessing an email address and getting it wrong – and suffer violations of security or privacy as a result. Facemail is an extension to a webmail system that aims to alleviate this problem by automatically displaying pictures of the selected recipients in a peripheral display, while the user is composing an email message. We describe techniques for obtaining faces from email addresses, and discovering mailing list memberships from existing web data sources, and a user interface design that keeps important faces recognizable while scaling up to hundreds or thousands of recipients. Preliminary experiments suggest that faces significantly improve users' ability to detect misdirected emails with only a brief glance.

Categories and Subject Descriptors
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

General Terms
Algorithms, Experimentation, Security, Human Factors.

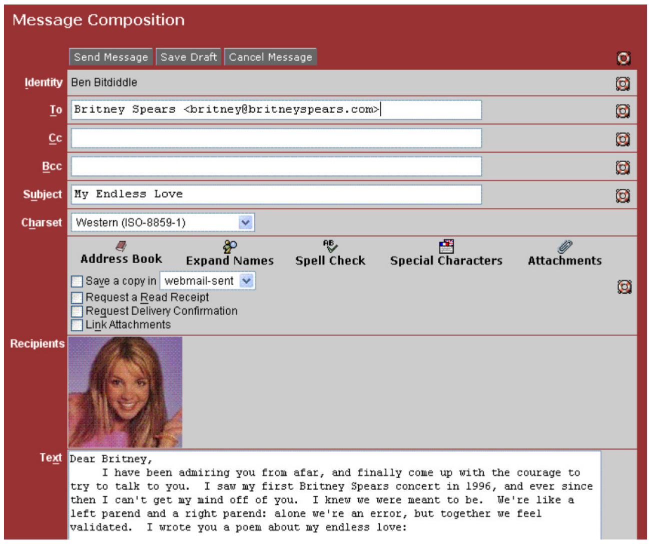
Keywords
Email, errors, Reply To All, security, privacy.

1. INTRODUCTION
Email has become a critically important medium, both for business and social use. One area that has not been closely studied, however, concerns the kinds of errors that users make when using email. For example, users often forget to add an attachment, forcing them to send another message with the missing attachment. This problem is common enough that several extensions to popular mail clients try to detect missing attachments by looking for keywords in the message body, so that the user can be warned before sending it.

Forgetting an attachment may be an annoyance, but sending the message to the *wrong people* may be a serious privacy or security problem. As email is used for more and more sensitive information, it becomes increasingly important that emails reach their correct recipients. We have collected a number of anecdotes that show some of the risks of misdirected email:

- A company employee mistakenly clicked Reply-To-All and sent very personal medical information to the entire company. Another employee was so embarrassed for her that she said, "I couldn't even make eye contact with her." [9]
- A large company has many employees with the same name. The email directory always displays them in the same order, and the first person on the list often seems to get email intended for one of the others.
- A teaching assistant for a course at our university had just written an exam. He intended to send it to the professor for checking, but instead he accidentally sent it to the list for the entire class. The exam had to be postponed while the TA wrote an entirely new one.
- The undergraduate admissions process at our school involves alumni called "Educational Counselors," who interview prospective students and send confidential reports about them to the admissions office. Educational counselors call themselves "ECs", and send their reports to educational-counselors@mit.edu. Our school also has an East Campus

Symposium On Usable Privacy and Security (SOUPS) 2007, July 18-20, 2007, Pittsburgh, PA, USA.



Identity: Ben Bitdiddle
To: Britney Spears <britney@britneyspears.com>
Subject: My Endless Love
Charset: Western (ISO-8859-1)
☐ Save a copy in webmail-sent
☐ Request a Read Receipt
☐ Request Delivery Confirmation
☐ Link Attachments

```
Dear Britney,
     I have been admiring you from afar, and finally come up with the courage to
try to talk to you.  I saw my first Britney Spears concert in 1996, and ever since
then I can't get my mind off of you.  I knew we were meant to be.  We're like a
left parend and a right parend: alone we're an error, but together we feel
validated.  I wrote you a poem about my endless love:
```

Figure 1: Facemail composition window.

dormitory, which is commonly called "EC," and which has a mailing list for all dorm residents called ec@mit.edu. About once or twice a year, the dorm list receives highly personal information about students from educational counselors who don't know the correct address for the Educational Counselors' office.

The frequency of stories like these suggests that misdirected email errors are all too easy to make. Probably the most common reason is invoking Reply-To-All instead of Reply, causing a message to be sent to an entire list of recipients rather than just one. (Using Reply instead of Reply-to-All is another possible error, but like missing attachments, having *too few* recipients is more an annoyance than a security problem.) Worse, some mailing lists automatically add a reply-to header pointing back to the list, so invoking *either* Reply or Reply-To-All will send the user's reply to the entire list, which may be surprising. Other reasons for misdirected email include similar email addresses, such as ec@mit.edu and educational-counselors@mit.edu in the example above, and a misspelled email address that inadvertently matches a different recipient.

To detect misdirected email errors, we propose augmenting the mail composition window with the actual *faces* of the recipients of the message. The faces appear automatically as the user is composing the message, acting as a peripheral interface that informs the user without demanding direct attention. Figure 1 shows our interface, called Facemail, implemented as an extension to our university's web-based email client.

Facemail's information display has two competing design goals: (1) make the recipients of the message easy to recognize at a glance, and (2) minimize the screen space it uses in the composition window. With few recipients, the design problem is straightforward, but for many recipients, a naïve approach would either use too much screen space or make the faces too small to be recognized. Facemail addresses the tradeoff by keeping as many faces recognizable as possible, while still conveying a sense of the overall *size* of the recipient list, degrading gracefully as the recipient list grows.

The Facemail prototype is implemented as a Firefox web browser extension, currently designed for one webmail system (IMP) but readily extensible to others. The database that maps between email addresses and faces is stored locally at the user's web browser, rather than at the server. This database is populated automatically by plugins that search various web services for faces, such as Google Images and Facebook. Facemail also tries to unpack mailing lists to find their subscribers, so that its display can show the *people* who might eventually read the message, not just the email addresses to which it was sent. Trying to solve these problems in general, for the entire Internet, is probably impossible without new standards and careful controls to protect users' privacy. Within a corporate email system, however, resolving faces and unpacking mailing lists should be much easier.

The contributions of this paper include: (1) the idea of reducing email composition errors by displaying recipients' faces in an automatic, peripheral display; (2) a display design that keeps faces recognizable while scaling up to hundreds or thousands of recipients; (3) techniques for discovering faces from existing web services, and an evaluation of their performance; and (4) a controlled experiment showing that, when allowed only a brief glance at the mail composition window, users are far more likely to detect errors with faces displayed than without.

2. RELATED WORK

Email security has received a great deal of attention [4,5,6,13], since email spoofing and confidentiality are growing areas of concern. Even if more email messages were cryptographically secure, however, there would still be the risk that the user would inadvertently send it, securely, to the wrong people. Facemail aims to fill a gap in security by reducing the likelihood of this error.

Many email clients offer auto-completion capabilities that help prevent errors when the user is typing a familiar email address. Apple Mail's auto-complete, in particular, highlights known addresses at a specific domain in a different color. For example, if emailing within Apple, all known addresses ending in "@apple.com" are highlighted with a specific color to show that they are known good addresses.

Google has recently introduced a new feature called Gmail pictures, where a Gmail user can select a picture to represent them, and other Gmail users can see the picture by hovering over a name or email address. Since this approach requires the user to mouse over to see the images, the user can't verify the recipients of their email without a conscious effort, which makes it less useful for catching errors than a continuously visible display. Other mail clients, such as Apple Mail, allow the user to associate photographs with email addresses in the address book, so that received messages display the sender's face, but do not use the faces in the mail composition window. Many instant messaging clients and desktop login windows for Windows and Mac OS also display a picture associated with a user name, but these displays are not targeted at error correction, nor do they encounter the scaling problem that email composition frequently does.

X-Face is a long-standing convention for including a 48x48 pixel black-and-white image as a header in an email message. Although never formally standardized, X-Face is nevertheless supported by many Unix mail clients, and by plugin extensions on other mail clients, such as Thunderbird's MessageFaces. Unlike X-Face, which displays the face of the *author* of a *received* message (if an X-Face header was included on that message), Facemail displays the faces of the recipients of a message being composed, a substantially different problem. If they were more widely used, X-Face headers would nevertheless be a good source of face images for Facemail, especially more recent extensions that support high-resolution color images.

ContactMap [12] is a *social desktop*, which displays the faces of the user's contacts on the desktop as proxies for their physical presence. The goals of ContactMap are to support, in a virtual workplace, social practices that are common in physical workplaces, such as social reminding (e.g. honoring commitments and keeping in touch with contacts) and social network mining (e.g. finding someone who might know an answer). Faces on the desktop can be used for sending email or instant messages, and as reminders for to-do items or unread email. Unlike ContactMap, Facemail has a different goal (preventing communication errors), which leads to different design characteristics (a peripheral display embedded in the email composition window, rather than a primary display on the desktop). ContactMap's display is also manually constructed by the user, including the selection and arrangement of the faces. ContactMap is also less concerned with scalability, first because it uses the entire desktop, and second because prior work had shown that most users have fewer than 150 direct contacts. Facemail, however, is a peripheral display, so less screen real estate is available to it, and it must scale to larger collections of recipients, because even users with few direct contacts may work in organizations with large mailing lists.

EmpathyBuddy [8] augments an email composer with *Chernov faces* displaying stylized emotions inferred from the content of the message, such as happiness, sadness, anger, fear, and surprise. The face is meant to represent the software's affective response to the user's words, not an actual human recipient. Nevertheless, it would be interesting to measure how viewing human faces during composition affects the tone of the email message.

Human faces have also been used for other purposes in computer security. For example, Passfaces uses faces for authentication, in which the user's password is represented as a sequence of faces chosen from arrays. Passfaces has been found to be less error-prone than conventional textual passwords [2], presumably because cued recall is easier than unassisted recall, but also because humans are very good at recognizing faces (as opposed to other kinds of symbols or images). Facemail seeks to derive similar benefits from displaying faces, though its goals are different.

3. USER INTERFACE

Facemail integrates the individual's faces directly into the email composition window, rather than with a popup window or mouseover. This approach does not require the user's direct attention; instead it acts as a peripheral interface which the user is likely to notice but does not have to focus on.

Faces appear in a reserved area between the email headers (To, CC, BCC) and the body of the message. When a blank message is first opened, this area is minimized and shows no faces. As the user enters recipients into the To, CC, or BCC fields, Facemail automatically expands the face display area and begins populating it with faces, running in the background if it needs to search for them from web sources. If the compose window was opened by clicking Reply or Reply-To-All, then the faces of the initial recipients are displayed automatically. Facemail keeps the face display synchronized with the To, CC, and BCC headers, so that no user interaction is required to make the correct faces appear.

A key principle guiding this design was to place the faces as close as possible to the message body, which presumably is the user's focus of attention while composing an email, so that unexpected recipients would be easier to notice. We also experimented with placing the faces in the message body itself, as translucent watermarks behind the text, but found that either the body text would be too hard to read or the faces too hard to recognize, with no happy medium between.

The faces for all recipients are included in the same area, regardless of whether they are found in the To, CC, or BCC fields. Since blind-carbon-copy (BCC) recipients are treated differently by the email system – other recipients aren't aware of their presence, and they won't be automatically copied on replies – it may make sense to distinguish the faces of BCC recipients in some way. BCC recipients are relatively rare, however, and we have found no convincing cases of errors involving BCC, so Facemail currently makes no distinction.

3.1. Displaying Mailing Lists

Since mailing lists are common, it is important for Facemail to provide a good mechanism for displaying the membership of those lists. Because lists can potentially have thousands of members, this display must be compact but still representative of its members.

There are several errors that users make with incorrectly sending emails to lists. First, they might send an email to a list that contains a person who should not receive it. For example, a surprise party invitation to a group of friends might accidentally include the friend who's meant to be surprised. Second, users might send an email not realizing how many people are on the list. Users trying to remove themselves from a list often send this request to the entire list, which may be far too many people. Third, users might mistake the intent of a list and send email to it incorrectly. For example, users could send an email about selling a refrigerator to a list that is only about offering free items and not about selling them.

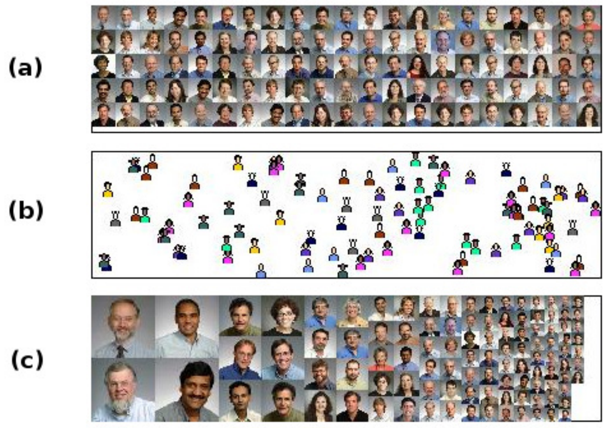


Figure 2: Alternative displays for mailing lists. (a) Fixed-size grid; (b) Random crowd of icons; (c) Progressive scaling grid.

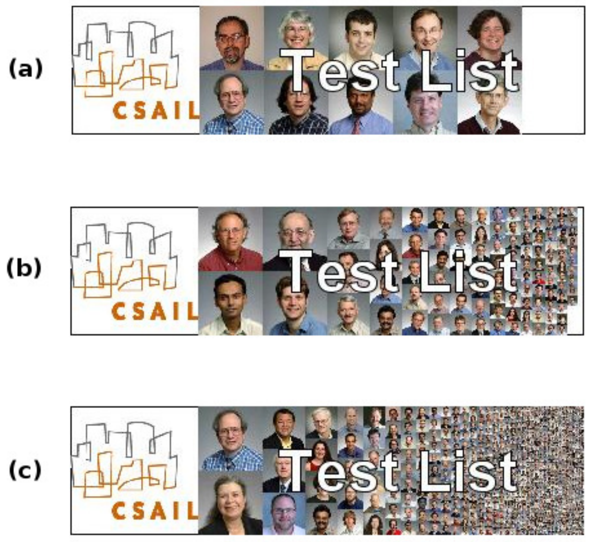


Figure 3: Progressive scaling grid showing (a) 10 faces, (b) 100 faces, and (c) 1000 faces.

To prevent these errors, Facemail's list display tries to answer three questions: (1) Who, specifically, is on the list? (2) How many people are on the list? (3) What is the purpose of the list?

In order to keep the display compact, we set an upper limit on the size of Facemail's display. Figure 2 shows some of the design possibilities considered for squeezing a large set of faces into a limited screen area. A simple grid of faces (Figure 2a) effectively answers the second question (*how many*) at the cost of the first question (*who*), since the faces become unrecognizable at small scales. A randomly arranged crowd (Figure 2b), of either icons or faces, has a similar problem. Facemail's current design trades off between *how many* and *who* by displaying the faces in a grid, whose cell size progressively shrinks from left to right (Figure 2c). As a result, faces on the left of the display are easy to recognize, while faces on the right might appear merely as a crowd of blobs.

To satisfy the third question (*list purpose*), Facemail overlays the list's name (which by default is its email address) on the face grid, and allows a list to have a logo for visual recognition. Figure 3 shows how the final design looks for three different scales (10, 100, and 1000 list members). Larger list sizes look much like 1000. Since a 1000-member list is already large as far as violations of privacy or security are concerned, it doesn't seem useful to visually distinguish even larger lists.

An important question in the progressive scaling grid is the *order* of the faces, since faces on the left are recognizable even in a large list, while faces on the right may not be. Currently, since Facemail collects faces automatically from unreliable sources, we simply display the *best* faces on the left – i.e., the images most likely to be the face of an actual recipient, as measured by the *a priori* quality of the source, relevance ranking if available, and validation by a face detection algorithm.

Assuming, however, a reliable source of faces, such as a corporate directory, there are better principles to guide the ordering. One principle is *familiarity*: recipients that the user knows better should be more recognizable. Familiarity could be measured by mining the user's email patterns to discover a social network [3]. Another principle is *list distinctiveness*: if two of the user's mailing lists have overlapping membership, it would be better to display people who are members of one list but not the other, so that the lists are visually distinctive.

A third principle is *social or organizational relationship*: since a major risk of misdirected email is security or privacy violation, it would be better to display people who represent the greatest risks, or at least represent the diversity of risk. For example, in corporate email, the user's immediate supervisor and direct reports should be recognizable if they're on the list; so, probably, should the CEO. In academia, faces from different roles (faculty, teaching assistants, undergraduates, administrators) should be recognizable, because they may have different access rights to the information embodied in the email. Some of this information is straightforward to obtain, since many corporate databases include org charts. In general, inferring relevant social relationships is a hard problem, in which the content of the message should also come into play.

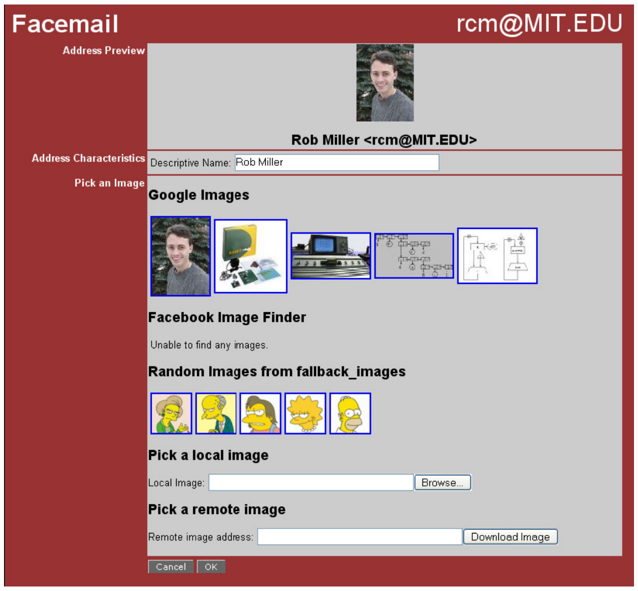
3.2. Editing Faces

Because the images that Facemail finds automatically are not necessarily correct, it is important that the user has a mechanism for easily editing Facemail's automatic choices. To do this, the user can simply click on an image in the compose window, and Facemail will pop up an editor for that address. The user can edit both individual addresses and mailing lists.

Figure 4 shows the editing interface for a single address. The user is presented with several choices of images that the system automatically finds, or they are allowed to supply their own. At the top, there is a preview of the image they have picked, and when they pick an image below the preview is dynamically updated.

In addition to an image itself, the user can enter or modify the recipient's full name (e.g., Bill Gates for billg@microsoft.com). When finding images, the system uses as much information as possible to come up with suggestions, and therefore providing a full name would allow it to make better image suggestions.

Figure 5 shows the list editing interface. There are many different list characteristics that the user can edit in this interface. First, they can modify the descriptive name and list logo in a very similar way to the single address editor. The system will suggest several images, but the user can also provide their own.

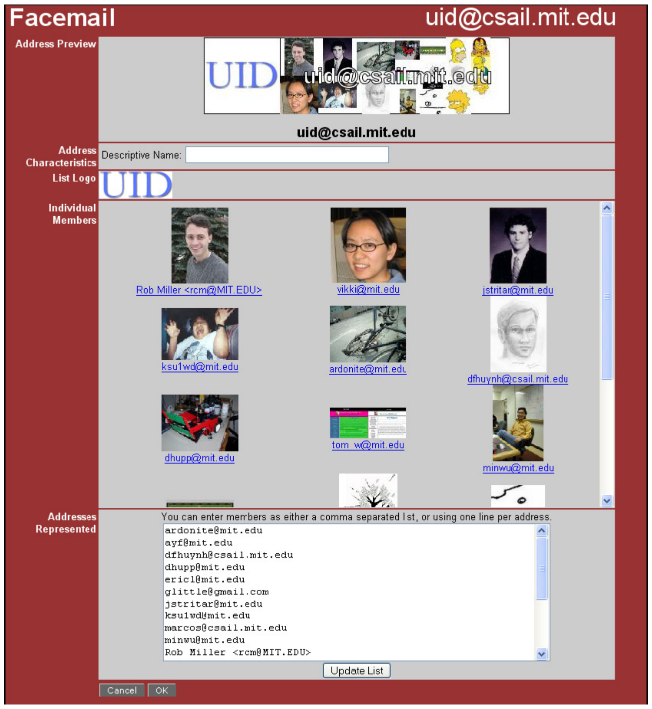


Rob Miller <rcm@MIT.EDU>

Descriptive Name: Rob Miller

Unable to find any images.

Figure 4: Interface for changing the face associated with an email addresss. Faces found automatically by Facemail are displayed as options to choose from; the user can also load an image from a file or URL.



You can enter members as either a comma separated list, or using one line per address.

```
ardonite@mit.edu
ayf@mit.edu
dfhuynh@csail.mit.edu
dhupp@mit.edu
ericl@mit.edu
glittle@gmail.com
jstritar@mit.edu
ksu1wd@mit.edu
marcos@csail.mit.edu
minwu@mit.edu
Rob Miller <rcm@MIT.EDU>
```

Figure 5: Editing interface for the display of a mailing list. Users can choose a list logo, reorder the faces, and change the list membership.

For the list itself, the user can edit the individual images of each of the members by clicking on them. The order of the list can be rearranged by dragging the images into the spot in the list where he wants them, and the preview at the top updates to show his changes. We considered supporting direct manipulation on the progressive-scaling grid itself, but for large lists, the images on the right of the grid would be nearly impossible to recognize, select, or edit. The current design is a reasonable compromise.

A user can also edit the membership of the list by editing the textbox in the bottom of the display. This is particularly useful if the system is unable to find the membership for the list automatically; the user can simply copy the list membership into the textbox and then click "update list" to create a preview of the display of that list.

Because the textbox and membership display both show different representations of the same information, the selections are synchronized. For example, if the user is typing or selects text in the textbox below, that selection is reflected in the display above. If the user clicks on the display of an individual member in the top display, the corresponding text is highlighted in the textbox below.

Lists may contain other lists, and therefore it is important that the user can see exactly which list a particular image is coming from. The individual members display shows the faces of every person on the list and its sublists, while the textbox displays the exact representation of the list, including references to its sublists. Therefore, if the user selects a face that came from a list, or highlights a list in the textbox, the selection shows which individual members make up that list.

4. IMPLEMENTATION

The Facemail prototype adds faces to an existing webmail system, IMP. Rather than changing the webmail server, however – which is hard to persuade system administrators to experiment with, since email has become so vital to organizations and people – Facemail runs entirely on the client side, as an extension to the user's web browser.

Facemail consists of a user interface component that runs in the web browser, and a backend written in Java that manages faces and mailing list information.

4.1. User Interface

Facemail's user interface is implemented using Chickenfoot [1], a Firefox extension toolkit designed for customizing web sites in the user's web browser. The main user interface component is Compose Trigger, a Chickenfoot "trigger script" that runs whenever the web browser displays a message composition window from the webmail system. This script connects to the Java backend to request faces corresponding to each email address, and inserts the faces into the compose window.

4.2. Address Resolving

One important task for the Facemail backend is determining whether an email address represents a single person or a list of people. If the address is a list, then Facemail needs to find the members of that list as well. Since there is no single service that Facemail can query to discover list membership for every list, address resolving is implemented using a plugin architecture. Each address resolver plugin implements a simple interface that takes an email address (as a string) and returns the membership of the list if the email address corresponds to a mailing list known to that plugin. Given a new email address, Facemail passes it to all resolver plugins to give them an opportunity to resolve it into a list. If no plugin succeeds, Facemail assumes the email address belongs to a single person.

The Facemail prototype currently has plugins for Blanche, a list management system specific to our university, and Mailman, a list maintenance system in widespread use on the Internet. Both list systems generally require a password to access list membership information. For Blanche, the user's university account is sufficient to view any list, and the list information is retrieved from a single server. For Mailman, however, the user may have a different password for each mailing list they subscribe to, and Mailman servers are scattered throughout the Internet. Currently, Facemail must be manually configured with the locations of Mailman servers to check, and when it must look up the membership of a list, it requests a password from the user.

Since Facemail has access to the user's email account, however, it could do a better job of discovering where Mailman servers are located and collect passwords. Like many list maintenance systems, Mailman adds List-* headers to messages distributed by the list, from which the location of the Mailman web server can be discovered. When the user subscribes to a list (and sometimes on a monthly basis), Mailman also sends a stylized email message containing the user's password. By scanning the user's email archive and monitoring incoming email, a future version of Facemail could configure the Mailman plugin automatically.

Most Mailman mailing lists make their membership list available only to list members, so if the user is sending mail to a list they don't subscribe to, then it may be impossible to display any faces for its members. When Facemail can determine that an email address corresponds to a list (e.g. by finding it on a known Mailman server) but cannot discover its membership, it generates a default membership of 100 placeholder addresses, so that it can at least give the sense that many people may receive the user's message.

4.3. Face Finding

Finding an image of a person's face given their email address is a hard problem. Many different websites or databases could contain these images, and some of those might be accessible only to authenticated users. Some people may not even have a suitable image accessible in any way to the system. Because of these problems, Facemail attempts to find a good default automatically, but it allows the user to change the image later if desired.

Like address resolving, Facemail uses a plugin architecture to find face images. The main difference is that in address resolving, the address resolver that succeeds is assumed to be correct because each address resolving plugin is precise, meaning that if it has an answer the answer must be correct. Face finding, on the other hand is often not precise, and therefore Facemail provides a mechanism to choose between images returned by different finders. Each face finder plugin implements an interface that takes an email address representing a single person and returns a list of rated images, where ratings range from 0.0 to 1.0. For the default image, Facemail simply picks the highest rated image.

Face finders that are more trusted or more precise than others should return higher ratings. For example, faces drawn from a corporate ID database should return a rating of 1.0, while faces found in Google Images may have a maximum rating of 0.5.

For face finders that draw from an image source that contains images other than faces, it is useful to rate resulting images using Intel's OpenCV face detector. When the face detector finds a face in the image, a rating is calculated based on the relative area of the face in the picture to an ideal area ratio. The ideal ratio was calculated by taking a database of 92 head-and-shoulders portraits of faculty and calculating the average ratio in those images. The average ratio of the face's area to the full image's area was 23%, and the Face detector rates images close to that ideal close to 1.0. If the image has a smaller ratio, then its rating decreases linearly to 0.0 as the area ratio decreases to 0. If the face ratio is too high, then the rating also decreases, but only down to 0.5 for a ratio of 100%, because an image that is purely a face is still a pretty good portrait for Facemail to use. Figure 6 shows an example of ratings that Facemail's face detector gives for various images.

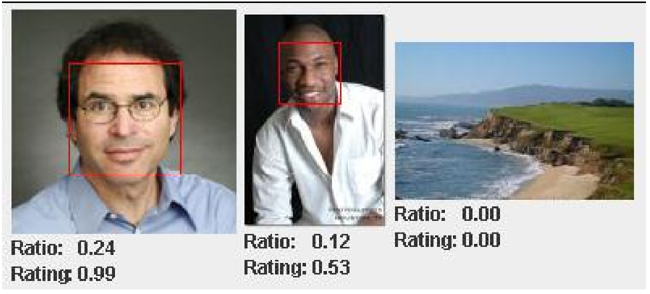


Figure 6: Ratings assigned by the face detector.

Facemail currently has two face finder plugins. The Google Images plugin uses Google Image Search (images.google.com) to search for images related to an email address. Email addresses are translated into a query simply by removing the '@' symbol, so billg@microsoft.com becomes "billg microsoft.com." Querying directly for "billg@microsoft.com" doesn't work well, because many users avoid publishing their email address in full in order to prevent spam.

Conveniently, the image thumbnails that Google returns in its summary page are just about the size of image that we want, so the plugin returns those thumbnails directly rather than downloading full-sized images.

The second plugin is for Facebook (www.facebook.com), a social networking site used largely by students at select high schools and colleges (although since late 2006, registration for the site has been open to anyone). The premise of Facebook is to promote social networking and allow users to post pictures of themselves and their friends, so it is a far higher-quality source of face images, although less ubiquitous than Google Images. In order to use the Facebook plugin, the user must be a Facebook member, and the plugin must be configured with the user's Facebook password.

Facebook does not let users search directly by email address, despite the fact that email addresses are stored in their database. As a result, the plugin queries Facebook with the full name from the email address, such as "Bill Gates." Since Facebook localizes itself to the user's campus, results that were at the same campus appear first, this rating helps us to choose between images. Since Facebook images are generally faces, the plugin does not run the face detector to rate the images; instead, it uses Facebook's search rank as the basis for the image rating.

Other social networking sites, such as Friendster and MySpace, could be queried for images in the same way as Facebook. In corporate settings, many companies have photo IDs for all employees and a centralized database containing those pictures, so a Facemail plugin could connect to this database. These face finders could simply return a rating of 1.0 for any faces they find, since they have a high probability of correctness, so that their results would take precedence over any of the heuristic searches.

4.4. Caching

In order to make Facemail faster and reduce load on the servers that Facemail searches, Facemail keeps a cache of face images and list information locally. Caching happens automatically during address resolving and face finding. In both cases, the cache is represented as another plugin which is consulted before other plugins.

Facemail currently does not expire or refresh the cache automatically, since using consistent faces in the interface has usability benefits. Users can manually request a new search for faces or refresh a list's membership using the editing interfaces (Figures 4 and 5).

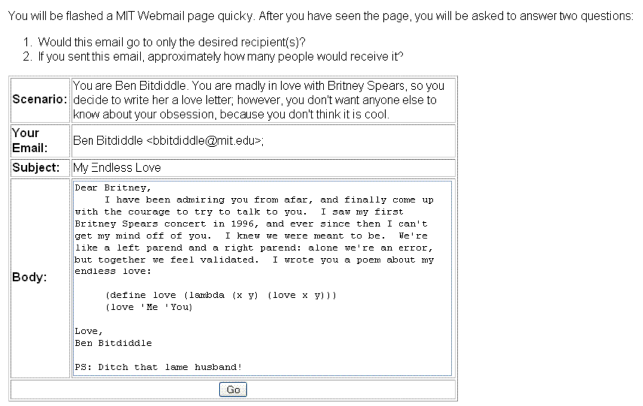
5. EVALUATION

This section describes two experiments conducted on Facemail. The first experiment concerned the user interface, and tested whether adding faces to a compose window makes misdirected emails easier to detect. The second experiment concerned the backend, specifically the ability of the heuristic face finder plugins to find correct faces for email addresses.

5.1. User Study

As a peripheral interface, Facemail must provide enough information in a single glance for the user to judge whether the recipients are correct. In order to evaluate how well Facemail achieves this goal, we conducted a web-based study in which users viewed an image of a message composition window, either with faces or without, and determined whether or not the recipients of the email were correct. To simulate a glance, the message window was shown for only a brief interval. We had two hypotheses: (1) users would detect incorrect recipients more often with faces than with plain email addresses; and (2) users would be better at estimating the *number* of recipients with our face display than with plain email addresses.

Each trial in the test had three parts. First, a page was displayed describing the scenario motivating the email message, and asking the user to assume the role of the email sender. Each scenario contained a short description of the situation, along with a description of the intended recipients (but not their actual faces or email addresses). Recipients were chosen to be familiar figures, either celebrities or well-known people at our university. Figure 7 shows an example of a scenario.



You will be flashed a MIT Webmail page quicky. After you have seen the page, you will be asked to answer two questions:

1. Would this email go to only the desired recipient(s)?
2. If you sent this email, approximately how many people would receive it?

Scenario:	You are Ben Bitdiddle. You are madly in love with Britney Spears, so you decide to write her a love letter; however, you don't want anyone else to know about your obsession, because you don't think it is cool.
Your Email:	Ben Bitdiddle <bbitdiddle@mit.edu>;
Subject:	My Endless Love
Body:	(see below)

```
Dear Britney,
     I have been admiring you from afar, and finally come up
with the courage to try to talk to you.  I saw my first
Britney Spears concert in 1996, and ever since then I can't
get my mind off of you.  I knew we were meant to be.  We're
like a left parend and a right parend: alone we're an error,
but together we feel validated.  I wrote you a poem about my
endless love:

     (define love (lambda (x y) (love x y)))
     (love 'Me 'You)

Love,
Ben Bitdiddle

PS: Ditch that lame husband!
```

Figure 7: Example scenario from the user study.

Once the user understood the scenario, pressing the Go button briefly displayed a picture of a message composition window showing a completed email message ready to be sent. Sometimes this window included faces, sometimes not. Sometimes the recipients on the message were correct, and sometimes not. After the brief display, the user was asked two multiple-choice questions. The first question was "Would this email go only to the desired recipient(s)?", and the choices were Yes, No, and Not Sure. The second question was "If you sent this email, approximately how many people would receive it?", with the choices Less than 10, More than 10, and Not Sure.

The test started with 3 warmup trials in which the message display interval was 10 seconds. These warmup scenarios were designed to get the user acquainted with the task, and with the format of the message composition window, in case it was unfamiliar. The warmup trials included one trial without faces, one trial with faces, and one trial with faces and the wrong recipients.

After the warmup, the user saw 30 trials, based on 5 basic scenarios, each with 3 possible recipient lists (only one of which was correct), and 2 possible interface conditions (with faces or without). The order of the trials was randomized. The display time grew shorter as the study proceeded, so that the first 10 trials were displayed for only 1 second, the second 10 trials for 0.5 seconds, and the last 10 trials for 0.1 seconds. In each block of 10 trials, the test presented 5 scenarios with faces and 5 scenarios without.

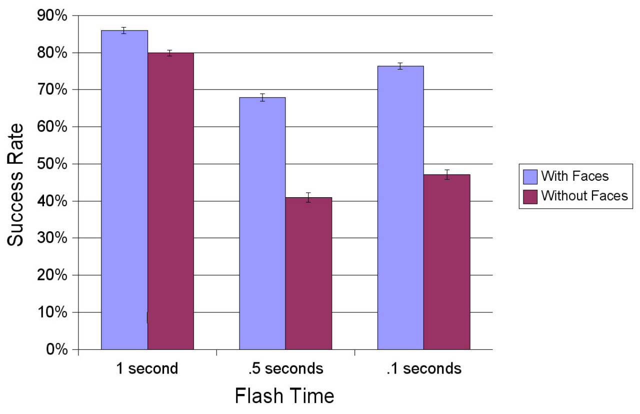


Figure 8: Success rate for the question "Would this email go only to the desired recipient(s)?" Not Sure responses are treated as wrong answers. Error bars show standard error.

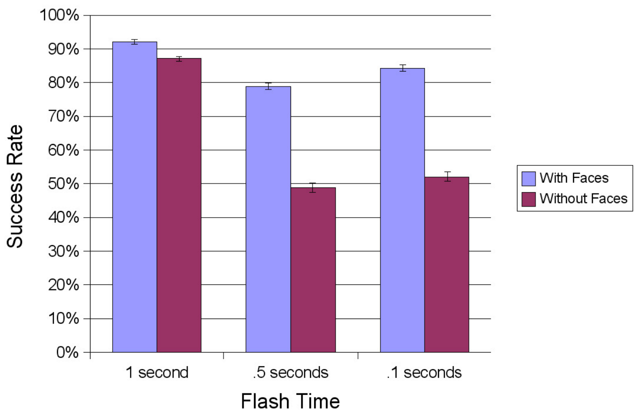


Figure 9: Success rate for the question "How many people would receive this email?" Not Sure responses are treated as wrong answers. Error bars show standard error.

We recruited 84 users to the web study by advertising to on-campus mailing lists. The web study collected information on their demographics and email use, but unfortunately this data was lost due to a programming bug. We believe that the subjects are representative of college students on our campus, and have found no reason to believe otherwise.

5.2. Results

Figures 8 and 9 show the fraction of correct answers given to each question, as a function of interface condition (with faces or without) and window display time. Any answer of Not Sure was treated as incorrect. When the user had time for a long glance at the recipients (1 second), faces made very little difference in the answers. This shows that our scenarios were designed so that the faces weren't *necessary* to answer the question, which was desirable. The information needed to decide whether the recipients were correct was available in the email addresses as well, assuming the user actually took the time to study them.

		Google Images				Facebook			
	Size	First	Top 5	Wrong	None	First	Top 5	Wrong	None
Students	63	6 (10%)	8 (10%)	1 (2%)	34 (54%)	17 (27%)	17 (27%)	10 (16%)	17 (27%)
Faculty	93	42 (45%)	47 (51%)	11 (12%)	15 (16%)	1 (1%)	1 (1%)	27 (29%)	60 (64%)

Table 10: Performance of face finders. *First* means the top rated image was correct; *Top 5* means the top 5 rated images included a correct image; *Wrong* means the chosen image was the wrong face; *None* means no faces were found.

When the user had little time for a glance, however (0.5 seconds or less), the presence of faces made a significant improvement in the user's ability to answer the questions correctly, increasing the success rate from around 50% (not much different from random guessing) to 70-80%. A two-factor ANOVA showed that both faces and flash times had statistically significant effects on both questions ($p < .001$). These results confirm both our hypotheses, and clearly indicate that faces are a better way to obtain information about recipients at a glance, even packed into a message composition window.

Faces also affected the user's confidence in their answer. For the desired-recipients question, 36% of the answers were Not Sure when no faces were shown, while only 14% were Not Sure when faces were shown The number-of-recipients question showed a similar effect: 35% Not Sure without faces, 14% with faces.

An interesting point to note is that users generally did better at the 0.1-second glance time than at the 0.5-second time, in both the face and no-face conditions. This can be explained by learning effects: display intervals were always presented in decreasing order, in order to acclimate the user to faster and faster speeds. We surmise that the 1-second glance was abundant time for most users, while the 0.5-second glance crossed a threshold that forced the user to develop a new strategy, so that their performance was temporarily worse in the 0.5-second block. Note, however, that the gap between faces and no-faces remains the same in spite of the learning effect.

5.3. Face Finding Experiment

Two different mailing lists of people with known faces were run through Facemail's face finders in order to test their effectiveness. The first mailing list consisted of 63 current and former undergraduate residents of a dormitory on our campus, and the second was the list of 93 computer science faculty at our university. Both Google Images and Facebook were able to find some images for both lists, though neither had a particularly high success rate. The success rates for both plugins are shown in Table 10.

The Facebook plugin was roughly 27% accurate for students, but since faculty are very unlikely to subscribe to Facebook, the success rate for faculty was only 1%. Google Images, on the other hand, performed much better for computer science faculty who were more likely to have their own home pages than students. For faculty, the Google Images plugin's first choice was correct 45% of the time, while for students it was correct only 10% of the time. Incidentally, all the faculty had a face image in a gallery on the school's website, but that site blocks Google from indexing the images. Therefore, all of the results were from faculty members' individual web pages, although many of them used the image copied from the official school website.

Another interesting statistic to note is how accurate an image source is at finding the correct image and placing it first, not just in the Top 5. For Facebook, if it found a correct image at all, then that image was invariably first. On the other hand, Google Images sometimes placed the correct image farther down in the rankings. Using the face detector considerably increased the success rate of

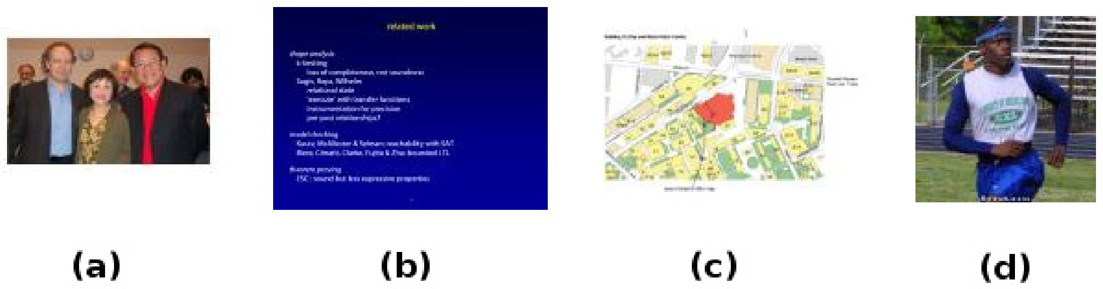


Figure 11: Incorrect images found by face finder plugins. Image (a) contains the correct person standing with two others; (b) is a slide written by the person; (c) is a map to the person's office; and (d) is a different person with the same name.

ranking images first, and it proved that it was useful for automatically finding good images.

This evaluation also showed us how valuable the face detection could be in some cases. For six addresses on the faculty list, and one address on the student list, face detection improved the rating such that the system automatically picked a more suitable image than Google's top-ranked result. Clearly, including face detection helped the system to find better results.

Figure 11 shows some of the ways that the face finders failed. Sometimes, face finders find the correct person, but the photo contains extra people that make the image confusing. Automatic cropping may help [7,10]; so might using the face detector to prefer images with just one face over those with more than one. Often, if the user has a webpage, there are graphics on that webpage such as a slide presentations or maps. The face detector helps to filter out these distractors and pick only images containing faces. Lastly, since some names are very common, a face finder relying solely on the user's real name might find a completely different person sharing the same name. This suggests that Facemail should attempt to search for email addresses (which are unique) before using real names.

6. DISCUSSION

This section addresses some of the limitations of Facemail and concerns that might be raised about it.

First, what if Facemail's display is wrong? The techniques used in the prototype implementation to resolve mailing lists and discover faces suffer from partial or imperfect information, as the face-finding experiment demonstrated. Despite being conducted with a nearly ideal corpus – computer science faculty members, who seem most likely of anyone to have a homepage containing a photo, and undergraduates, who seem most likely to have a Facebook page – the best success rate for our face-finding plugins was still below 50%. Mailing list resolvers suffer from similar problems when faced with resolving an arbitrary email address from somewhere on the Internet.

One answer is that Facemail may be better suited to corporate intranets and "walled garden" email systems, such as Google Mail, in which reliable information about faces and mailing lists is more readily available.

Facemail may still provide error-checking value, even if its guesses are sometimes wrong. If it determines that an address is a mailing list, it displays an array of placeholder faces, even if it can't correctly resolve any of them – so the user can notice that the email is going to a crowd, not just to one person. Pressing Reply To All instead of Reply will always display at least *two* faces, rather than one, even if both faces are wrong. Of course, the user can always select faces manually for their most frequent (or most important) correspondents, but an interface designed primarily for error prevention should not count on users to actively contribute to it.

A related issue is the *quality* of the faces displayed by Facemail, even if they are correct. Group photos and full-body shots are not likely to be effective, since Facemail can display only a thumbnail. Automatic cropping might help here. Similarly, the "faces" chosen by users for instant messaging and discussion forums are frequently not photographs at all, but rather avatars or cute icons. We doubt that displaying an icon would be as effective at preventing misdirected email as a real human face.

Facemail has some risks of its own. There is a serious risk, for example, if the user tries to guess a recipient's email address, and Facemail appears to confirm an incorrect guess by displaying the right face. The current user interface makes no distinction between faces obtained from a corporate database (rated 1.0) and faces found out on the Web (which might be rated much lower). One might hope that this would happen very rarely by accident, but one can imagine an attacker trying to subvert the email-address-to-face mapping in order to trick users into sending private information to the wrong place.

Should faces be displayed on received messages as well, as in mail clients that support X-Face? A serious concern here (which did not exist when X-Face first appeared) is *phishing*, fake emails that try to persuade the user to give away sensitive personal information. It's already easy to create a fake message that appears to come from any email address. Adding human faces that apparently legitimize these messages will only make the problem worse.

One solution to these security problems would be to combine faces with digital signatures, so that Facemail only displays recipient faces that have been digitally signed, either by the recipients or by a trusted third party. Unsigned faces could be omitted entirely or visually marked as uncertain, until the user explicitly validates the connection between the email address and the face.

Facemail may also threaten the user's privacy, because it enables easier shoulder-surfing. Displaying faces not only helps the user notice and recognize the recipients of an email, but also helps other people who happen to glance at the user's screen recognize the user's correspondents. This may be an inevitable consequence of designing a glanceable interface: information that is easy for the user to perceive at a glance is liable to be easy for *anyone* to perceive. Privacy-augmented display techniques [11], such as blinding or secret visual codes, are one proposed approach to shoulder-surfing, but these techniques would seem to work against the error-prevention goals of Facemail, since they require much greater attention from a user: intentionally moving aside a blinder, for example, or mentally mapping a code. A better compromise might be to make Facemail *transient*. The faces might appear briefly at important points of an email composition, such as when the user is choosing recipients, or when the user is about to send the message. At other times, the faces would be invisible, or very small.

Facemail also makes the membership of mailing lists far more visible than normally, which may threaten the privacy of other list members. Although many mailing lists technically make the membership of the list available to the list's subscribers, many users are probably unaware of it, and few would make the effort to look at it regularly, anyway. Facemail, on the other hand, puts the list members right in your face. For people who *send* to the mailing list, this is on balance a good thing, if it saves them from embarrassment or privacy violation. For other people, however, Facemail may change the social dynamics of mailing list subscription, making it less anonymous in practice, and harder to *lurk* on a mailing list. Currently, if you never send a message to the small list *stinky-cheese-lovers@yahoo.com*, then few people are likely to notice that you're a stinky cheese lover. But with Facemail, your face will show up in every other subscriber's compose window. Mailing list subscription would thus become more like a chat room, or a physical group meeting, in that your presence is easy to observe.

Security issues aside, adding faces to the composition window may have secondary benefits beyond error prevention. It may make writing an email feel more personal and enjoyable. It may also help to moderate the tone of email, by forcing the writer to look their victim in the eye, so to speak, before flaming.

Facemail may not go far enough in preventing unintentional information disclosure. For example, many mailing lists have public archives posted on the Web and indexed by search engines. How should Facemail display a list like this, whose recipients may include not just the immediate subscribers but in fact everybody in the world with web access, now and in the future? Many users aren't aware of the traces they leave on the Web that may return later to bite them. How can Facemail's visualization convey the message that a future employer or potential spouse might find and read this email too?

Finally, we have not yet conducted a full-scale field evaluation of Facemail, to determine whether it actually prevents errors in the wild. We have, however, been using it ourselves for several months. We found the faces distracting at first, partly because they were interesting (Facemail's automatic suggestions were often amusing), and partly because the face display kept changing as the system downloaded and added new faces. Over time, however, as we grew more accustomed to the faces, and the system built up a database of faces for common recipients, we paid less attention to the display. Facemail even caught an error: one of us accidentally mistyped a mailing list address, and the fact that Facemail didn't display the correct faces clued him in to his error.

7. CONCLUSION AND FUTURE WORK

This paper has described Facemail, a novel approach to preventing information disclosure errors caused by misdirected email. Facemail automatically collects faces for email addresses and renders them in a peripheral display while the user is composing an email message. Our experiments showed that faces significantly improved users' ability to detect misdirected emails at a glance, and that automatic face collection is possible, if not as good as might be desired.

Future work on the implementation side includes extending Facemail to support other webmail systems, such as Yahoo Mail, HotMail, and Google Mail, and adding plugins that increase the scope of Facemail's automatic information collection, such as X-Face headers, Google Mail contact images, other social networking sites, and other list maintenance systems. On the evaluation side, we are planning a controlled study in which users compose messages in the lab, with artificially injected errors like Reply To All instead of Reply. We also hope to deploy Facemail in a corporate setting, in order to do a field study where high-quality face information is available.

8. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under award number IIS-0447800. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

1. Bolin, M., Webber, M., Rha, P., Wilson, T., and Miller, R.C. Automation and customization of rendered web pages. In *Proc. UIST '05*, 163–172.
2. Brostoff, S. and Sasse, M.A. Are Passfaces more usable than passwords? A field trial investigation. In *Proc. HCI 2000*, 405-424.
3. Culotta, A., Bekkerman, R., and McCallum, A. Extracting social networks and contact information from email and the web. In *Proc. CEAS 2004.*
4. Garfinkel, S.L., Margrave, D., Schiller, J.I., Nordlander, E., and Miller, R.C. How to make secure email easier to use. In *Proc. CHI '05*, 701–710.
5. Garfinkel, S.L. and Miller, R.C. Johnny 2: a user test of key continuity management with S/MIME and Outlook Express. In *Proc. SOUPS '05*, 13–24.
6. Gaw, S., Felten, E.W., and Fernandez-Kelly, P. Secrecy, flagging, and paranoia: adoption criteria in encrypted email. In *Proc. CHI '06*, 591-600.
7. Liu, F. and Gleicher, M. Automatic image retargeting with fisheye-view warping. In *Proc. UIST '05*, 153-162.
8. Liu, H., Lieberman, H., and Selker, T. *Automatic Affective Feedback in an Email Browser.* MIT Media Laboratory Software Agents Group Technical Report, November 2002.
9. Sandberg, J. Never a safe feature, 'reply to all' falls into the wrong hands. *Wall Street Journal*, Oct. 25, 2005.
10. Suh, B., Ling, H., Bederson, B.B., and Jacobs, D.W. Automatic thumbnail cropping and its effectiveness. In *Proc. UIST 2003*, 95-104.
11. Tarasewich, P. and Campbell, C. What are you looking at? In *Proc. SOUPS 2005.*
12. Whittaker, S., Jones, Q., Nardi, B., Creech, M., Terveen, L., Isaacs, E., and Hainsworth, J. ContactMap: organizing communication in a social desktop. *ACM Transactions on Human-Computer Interaction*, 11(4), December 2004, 445-471,
13. Whitten, A. and Tygar, J.D. Why Johnny can't encrypt: a usability evaluation of PGP 5.0. In *Proc. USENIX Security 1999*, 169–184.