Topika: Integrating Collaborative Sharing with Email

Jalal Mahmud, Tara Matthews, Steve Whittaker*, Thomas P. Moran, Tessa Lau
IBM Research - Almaden, 650 Harry Road, San Jose, CA 95120
{jumahmud, tlmatthe, tpmoran, tessalau}@us.ibm.com
* University of California Santa Cruz, 1156 High Street, Santa Cruz, CA 95064
swhittak@ucsc.edu

ABSTRACT
New enterprise tools (wikis, team spaces, social tags) offer potential benefits for enterprise collaboration, providing shared resources to organize work. However, a vast amount of collaboration still takes place by email. But email is problematic for collaboration because information may be distributed across multiple messages in an overloaded inbox. Email also increases workload as each individual has to manage their own versions of collaborative materials. We present a novel system, Topika that integrates email with collaboration tools. It allows users to continue to use email while also enjoying the benefits of these dedicated tools. When a user composes an email Topika analyzes the message and suggests relevant shared spaces (e.g., wiki pages) within the user's collaboration tools. This allows her to post the email to those spaces. An evaluation of Topika's suggestion algorithm shows that it performs well at accurately suggesting shared spaces.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Algorithms, Design, Human Factors.

Keywords: CSCW, social computing, office, workplace.

INTRODUCTION
New enterprise tools (wikis, team spaces, social tags) offer potential benefits for enterprise collaboration, providing shared resources to organize work. However, attempts to introduce team wikis or similar tools into organizations have sometimes proved unsuccessful [7,9], especially when teams work on focused projects. Considerable effort is required to shift to these tools since teams have to all agree on a tool they want to use and make changes to their collaborative practices to use it. As a result, teams tend to revert to email. In our own studies of collaboration tool adoption in the workplace, reverting to email represented the main reason for not adopting collaboration software [3,13,15]. People 'live in' their email because it serves as task manager, file system, contact list, and alerting mechanism as well as supporting asynchronous communication [4,14]. However, email studies show many disadvantages of using email for collaboration, e.g., related messages get distributed throughout the inbox making it hard to collate, track, and monitor [14]. Also, every involved participant has to organize and monitor their own versions of collaborative materials. And because messages are distributed to explicitly named email recipients, senders may forget to include relevant people [13].

Solving this problem could make transformative improvements to online collaboration. Participants in our studies [3,15] indicated that if contributing to collaboration tools was part of their email practice, they would do so more often. They believed that if fellow group members could see the value of collaboration tools, which would require that enough content is posted, they would put more effort into using them.

This paper presents a new tool, *Topika*, which addresses these problems by bridging between email and enterprise collaboration tools. Specifically, it makes posting to these tools a seamless part of writing and responding to email, and helps groups seed their spaces with enough content to see the value of collecting materials in a shared space. To

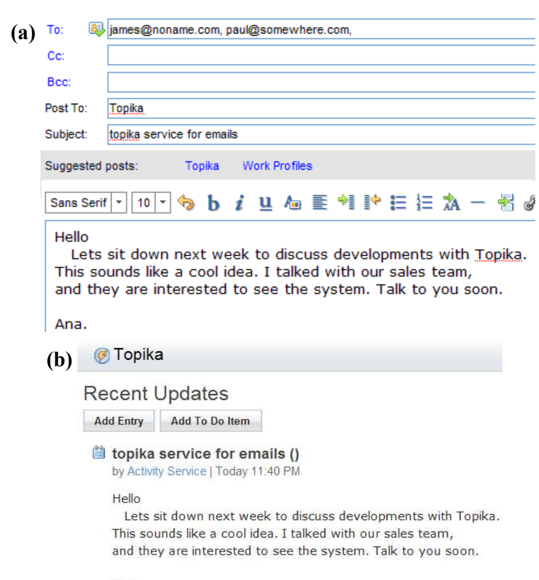

Figure 1. (a) The Topika email interface suggests collaborative spaces where this email can be shared ("Suggested posts"), and posts the email to spaces the user selects ("Post to"). (b) Email post to a collaboration tool (Activities [15]).

do this, Topika enhances existing email clients to provide *suggestions* about relevant shared spaces within collaboration tools. Users can forward their emails to one or more suggested spaces. The suggestion mechanism reminds users about the existence of the relevant shared spaces, simplifying the process of contributing to them. Figure 1(a) shows the Topika-enhanced email interface and Figure 1(b) shows an email post in a collaboration tool.

As a concrete example, consider Ana, a technical sales rep who participates in multiple collaborative groups to solve customer problems. Groups in her department organize their customer-related work in the Activities tool [15] in Lotus Connections [1]. But Ana has a hard time keeping track of all her Activities, and she finds it difficult to maintain her use of Activities when emailing is so ready-at-hand. Ana just received a call from a customer, and she wants to tell the other tech reps about the customer's request. She starts an email to the tech reps. At this point Topika suggests (based on the people in the To field and the message content) a particular Activity to post this to. 'Ah yes,' she thinks, this should be posted to the Activity so other team members can find it later. She accepts Topika's suggestion. When she hits Send, it is posted to the Activity as well as being sent directly to the tech reps.

RELATED WORK

Embedding task management in email

Embedding task management within email clients is described in [4]. In contrast, our approach allows users to continue to use email in the standard way while also having their work posted to appropriate collaborative tools. In semantic email [8], the sender inserts semantic tags to automate email-mediated applications. To participate, users must use a dedicated client that can interpret the tags and applying tags is a labor intensive process for the sender. In our approach, the burden on the sender is minimal.

Tool addressing

Several collaboration tools provide special email addresses so that people can direct an email to them (e.g., Yahoo Groups). A variant of this is that users can tag emails and these tags are used to post emails on a webpage [9]. However, tool addressing imposes a significant memory load: users must remember both that there are appropriate tools to send to, as well as the *exact tool address*. In contrast, Topika both reminds users about relevant shared spaces and allows them to post email to those spaces.

Automatic foldering

Researchers have developed techniques for automatically classifying emails into folders [10,12]. Topika is different from this prior work because it classifies and routes emails to actual shared spaces. Dredze et. al. classified emails as part of an ongoing activity [5]. However activity management in an email interface is not our goal; rather, Topika enables users to transition management of their collaborative activities to appropriate collaboration tools.

THE TOPIKA SYSTEM

Given email's centrality in the work ecology, we wanted people to continue with their standard email habits, but to also exploit the benefits of collaborative tools. We therefore generated two key design requirements that were intended to achieve these goals.

Minimal modification to the email client while providing dedicated collaboration support. This precludes complex thread visualization or embedded task management, which have both been proposed to integrate email and collaboration tools. Both require substantial modifications to the email client. While simpler threading approaches such as Gmail introduce only small client modifications, they do not provide any direct support for collaboration, and all users have to independently manage messages and resources associated with the collaboration. Furthermore, simple threading cannot cope with complex collaborations that may distribute across multiple conversational threads.

Minimal additional cognitive overhead. Users often forget to add a document attachment even when sending the document is the main point of their message [6]. This forgetting precludes similar 'tool attachment' solutions, e.g. requiring users to remember to append a collaboration tool address while they are composing the message. Furthermore, our analysis of the collaborative tool use of 12 users revealed complex usage patterns. People were using as many as 42 different shared spaces within collaborative tools. The sheer number of shared spaces used precludes a simple solution listing all of them. Again to reduce cognitive overhead, we rejected approaches relying on users remembering addresses or tags to route the message. To minimize this overhead, the system should both actively remind the user about the existence of relevant shared spaces and suggest relevant addresses.

Figure 3 shows Topika system architecture. Note that the system requires minimal client modification, and uses standard techniques to represent and mine collaborative applications. The core component is the *work profile* which is built by mining and indexing the user's *shared spaces* within different collaboration tools. The *work profile builder* takes the user's authentication information (e.g., username and password in collaboration tools) and builds that user's work profile by extracting his/her shared spaces from the collaboration tools. It uses the standard ATOM API [2] exposed by the collaboration tools to retrieve an ATOM feed for each shared space and creates a *work unit* for each of them (1 work unit per shared space). A work unit contains a list of features, such as the *title* of the shared space, the *type* of the collaboration tool, *people* who are members of the shared space, e*mail* addresses of these members, *keywords* collected from the shared space contents, a *unique identifier* for the shared space, and *weights* indicating the relevance of the title, people, and keywords in the work unit. The TF-IDF [11] algorithm is used to compute weights of keyword, people, and title words. The work profile for each user is stored as a separate file on a server.

Our user interface is embedded in an email client (shown as *emailer* in Figure 3). As the user begins to compose an email (see Figure 1(a)), the *emailer* calls a *suggestion agent.* This invokes the *matcher*, which uses a similarity function to compute the similarity value of each work unit with the email and then picks the top-scoring *K* work units as relevant. A simple definition of this similarity function could be the cosine similarity [11] between the bag-of-words from the email and the work unit. However, such a simplified approach ignores the structure of the email (recipients, subject, and body) and the work unit (names, title, and keywords) and could yield poor similarity scores for an email that does not have enough word-matches with the work unit. To incorporate this structured data, we define our similarity function as a weighted combination of similarity values (where weights are determined experimentally). These similarity values use the cosine similarity values of two vectors and are defined as follows:

- *people:* Email recipients and work unit people similarity.
- *subject-title:* Email subject and work unit title similarity.
- *subject-keywords:* Email subject and work unit keywords similarity.
- *body-title:* Email body words and work unit title similarity.
- *body-keywords:* Email body words and work unit keywords similarity.

The titles of the matched work units are returned by the matcher and displayed by the suggestion agent as links in the “suggested posts” email interface field (see Figure 1a). The user can click on one or more suggestions to select them. Once selected, they appear in the “post to” field. When the user hits send, the *router* posts the email message to the shared space using the ATOM API exposed by the corresponding collaboration tool. Our *router* posts an email as a new entry when the selected shared space is an Activity and as a new wiki page when it is a Wiki.

Knowledge of specific clients is only needed to display suggestions in the email interface and to post to specific collaboration tools (such as Lotus Connections [1]). The algorithm to build the work profile and compute and display suggestions is general.

SUGGESTION ALGORITHM EVALUATION

Collecting Labeled Email Data

To train and evaluate the suggestion algorithm we needed email messages labeled with their relevant work units. We collected 1237 emails from 12 users (approximately 100 emails per user). We automatically extracted people’s work units from the collaboration tools they used (Activities and Wikis), using Topika’s *work profile builder* utility. We also built a utility program that extracted the

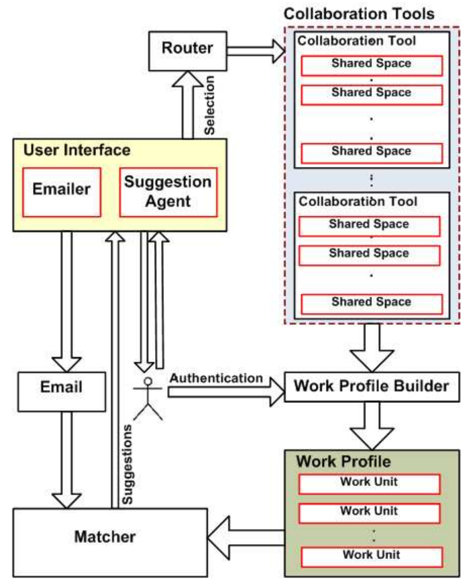

Figure 3. Topika system architecture.

emails from their inbox and put them in a spreadsheet. The participants were shown that spreadsheet and a list of work units extracted from their collaboration tools in the enterprise network. If an email was relevant to any of the work units, they made an entry in the spreadsheet containing the title of the email and a list of the work units. If there were multiple relevant work units, participants were asked to label the email with all relevant work units. If no work units were relevant, participants did not label the email. In total, 65% of the emails were unlabelled, 24% had 1 label, 5% had 2 labels, 1.4% had 3 labels, and 4.6% had 4 to 10 labels.

Experimental setup

We used this labeled user data to determine the weights of the similarity metrics and to evaluate the quality of our suggestions using standard recall (*r*) and precision (*p*) metrics**.** We divided the dataset into two parts: training (50% of the randomly selected emails) and validation (rest of the emails). We used the training set to learn the weights as follows: for each similarity metric, we compared the relevancy label computed by the algorithm (when only that similarity was used) for each *(email, work unit)* pair with the relevancy label supplied by the user (an email is either relevant with a work unit or not) and counted the number of matches. We call this the agreement score of that similarity metric. Weight of the similarity metric is computed by normalizing its agreement score by the total agreement score (i.e., the sum of all agreement scores for all similarity metrics). The resulting weights were [*people-weight*=0.17, *subject-title-weight*=0.28, *subject-keywords-weight*=0.22, *body-title-weight*=0.19, *body-keywords-weight*=0.14].

For our experiments, the size of the maximum number of suggestions (*K*) was varied from 1 to 10. Here, K represents the maximum, so if K = 1, the system will show 0 or 1 suggestions, if K = 2, the system will show 0, 1, 2 suggestions. We did not consider emails that had more labels than *K*.

Recall results: Were the right work units suggested?

Recall assessed how many suggestions matched a user's email work unit labels. As expected, the more suggestions given (i.e., higher values of *K*), the better the algorithm's recall performance across all similarity metrics. But combining all of the similarity metrics led to significantly better suggestions than any of the individual similarity metrics (two-tailed, paired t-test, all $p<0.0001$). On average across different values of *K*, combining similarity metrics generated 73% of the correct suggestions.

Precision results: How noisy were the suggestions?

As expected, the more suggestions given (i.e., higher values of *K*), the more false positives (i.e., the lower the precision) across all similarity metrics. Both email subject and work unit title similarity (*subject-title*) and combining all of the similarity metrics (*combined*) led to significantly better precision than all other similarity metrics (two-tailed, paired t-test, both $p<0.0001$), but were not significantly different from each other. On average, *subject-title* led to 31% false positives (i.e., 69% precision) and *combined* led to 37% false positives (i.e., 63% precision). We also ran a separate experiment omitting the "no label" emails and observed 15-20% increase in precision.

Effect of user collaboration pattern

To see whether an individual user's collaboration pattern could affect the similarity metrics that lead to the best suggestions, we compared our algorithm's recall and precision performance for two users: User 1, who had significant variation in the people she collaborated with across work units, and User 2, who worked with many of the same people across work units. We found that *people* led to more accurate suggestions than any other individual similarity metrics for the User 1, whereas *subject-title* led to more accurate suggestions for User 2, and *combined* outperformed in both cases. We leave it as future work to update the weights of similarity metrics depending on each user's collaboration pattern (which may change over time).

How many suggestions are best for both accuracy & noise?

Our experiments show that there is a trade-off between recall and precision. However because users' normal problem is that they forget the existence of collaboration tools, we thought it better to optimize recall by over-generating suggestions and allowing user to reject suggestions. But too much noise in the suggestions might also lead to user annoyance. One design question is how many suggestions to present to users to be accurate without annoying them. Our results show that maximum F_1 (harmonic mean of precision and recall) occurs when 6 suggestions are shown (for $K = 6$, F_1 becomes 0.68).

CONCLUSIONS AND FUTURE WORK

Despite providing better support for collaborative task management than email, enterprise Web 2.0 tools are currently under-utilized. Topika bridges the gap between email and collaboration tools, allowing users to exploit dedicated collaboration tools while still using email. Prior studies suggest that such integration may enhance groups' ability to adopt new collaboration tools. Future work will involve extending our system to build users' profiles from more collaboration tools, e.g., blogs and discussion forums. Also, we plan to deploy Topika with users and explore additional user issues, such as determining more sophisticated strategies for posting content.

REFERENCES

1. http://www.ibm.com/software/lotus/connections/
2. http://en.wikipedia.org/wiki/Atom_(standard)
3. Balakrishnan, A., Matthews, T., Moran, T.P. Fitting an activity-centric system into an ecology of workplace tools. In *Proc. of CHI 2010*, pages 787-790.
4. Bellotti, V., Ducheneaut, N., Howard, M., Smith, I. Taking email to task: The design and evaluation of a task management centered email tool. *Proc of CHI '03*. 345-352.
5. Dredze, M., Lau, T., Kushmerick, N. Automatically classifying emails into activities. *Proc of IUI '06*, 70-77.
6. Dredze, M., Brooks, T., Carroll, J., Magarick, J., Blitzer, J., Pereira, F, Intelligent email: reply and attachment predictions. *Proc of IUI 2008, 70-77.*
7. Holtzblatt, L.J., Damianos, L.E., Weiss, D. Factors impeding wiki use in the enterprise: A case study. CSCW Workshop "Collective Intelligence in Organizations," 2010.
8. McDowell, L., Etzioni, O., Halevy, A., Levy, H. Semantic email. *Proc. of WWW '04*, 244–254.
9. Nelson, L., Nairn, R., Chi E.H. Mail2Tag: Efficient targeting of news in an organization. CSCW Workshop "Collective Intelligence in Organizations," 2010.
10. Pazzani, M. J, Representation of electronic mail filtering profiles: A user study. *Proc. of IUI '00*, 202-206.
11. Salton, G., Wong, A., Yang, C.S. A vector space model for automatic indexing. *Comm. of the ACM, 18*(11): 613–620, 1975.
12. Segal, R.B., Kephart, J.O. Mailcat: An intelligent assistant for organizing e-mail. *Proc. of AAAI '99*, 276-282.
13. Tang, John C., James Lin, Jeffrey S. Pierce, Steve Whittaker, and Clemens Drews, "Recent Shortcuts: Using Recent Interactions to Support Shared Activities", *CHI 2007*, San Jose, CA, pp. 1263-1272.
14. Whittaker, S., Sidner, C. Email overload: Exploring personal information management of email. *Proc. of CHI '96*, 276-283.
15. Yarosh, S., Matthews, T., Moran, T., Smith, B. What is an activity? Appropriating an activity-centric system. In *Proc of INTERACT 2009*, pages 582-595.