Mail2Wiki: Low-cost Sharing and Early Curation from Email to Wikis

Ben Hanrahan*, Guillaume Bouchard†, Gregorio Convertino*, Thiebaud Weksteen†, Nicholas Kong*, Cedric Archambeau†, Ed H. Chi*[1]

*Palo Alto Research Center (PARC)
Palo Alto, California
{firstname.lastname}@parc.com

†Xerox Research Centre Europe
Grenoble, France
{firstname.lastname}@xrce.xerox.com

ABSTRACT

In this design paper we motivate and describe the Mail2Wiki system, which enables low-cost sharing and early curation from email to wikis by knowledge workers. We aim to aid adoption of enterprise wikis and enable more efficient knowledge sharing and reuse. We present a design rationale grounded in prior empirical work, the design of the system, and the evaluation of the user interface. The system includes two alternative front-ends to enable incremental adoption by workers who are currently using email to share with their communities.

Keywords

Corporate wikis, organization, design

INTRODUCTION

In our knowledge economy the ability to *efficiently share and reuse* knowledge among workers is a key advantage. Enterprise wikis offer collaborative authoring mechanisms as a way to develop and organize such knowledge. Using wikis, workers can share content such as project updates, organized lists of tools or links, calls for participation, reports, publications, and frequently asked questions.

However, much of the current sharing among knowledge workers continues to occur via *email*, which remains the central content management tool [12]. This leads to two problems. First, email provides an inefficient channel to share, as it does not enable reuse across the organization. In fact, useful knowledge is trapped in personal inboxes, making it unavailable to coworkers and making reuse difficult. This problem is exacerbated when a worker leaves an organization, rendering the information lost. Second, as more media are piped into email (e.g., updates from social networks) workers increasingly experience email overload, in the absence of suitable tools for offloading content from email [37].

On the other hand, *enterprise wikis*, in spite of being flexible and ubiquitous tools for sharing, struggle to reach the adoption rates required to make them sustainable and valuable [1, 7, 8, 17, 19]. Similar adoption problems have been observed for other Enterprise 2.0 tools [23] such as corporate blogs, Q&A sites, microblogs, and social networking sites (e.g. [6, 13, 15, 14]).

Researchers have started to investigate this *adoption problem* through examination of specific deployments of enterprise wiki and Web 2.0 tools [29]. These studies point to both the intrinsic limitations within an enterprise and the deficiencies in the design and deployment of these new tools. One such limitation is a very small population willing to edit and maintain enterprise wiki pages (see power law distributions in [6]), which results in less and sparser shared content in comparison to the same tools deployed outside enterprises (e.g., Wikipedia, [36]).

The results of these studies directly motivate the Mail2Wiki system proposed in this paper. Mail2Wiki focuses on two key limitations of current enterprise collaboration tools (such as wikis) that seem particular hindrances to adoption [8, 17, 19]. First, the *high interaction cost* (i.e., too many steps or context switches) for contributing and organizing content. For example, researchers have observed that it takes significantly less effort to email information than create a wiki entry [19]. A second key limitation of current Enterprise 2.0 tools is their *poor integration* with existing content-management tools and practices. Some researchers have observed that knowledge workers tend to fall back on email when there are too many non-integrated channels for sharing [2]. This suggests that integrating such tools with email may be an unexploited opportunity. For example, while reviewing successful adoptions of early enterprise systems, Palen and Grudin [30] argued that the success of the electronic calendar was due, in part, to the peer pressure that was elicited by integrating the calendar with email. Using email, non-users are reminded of others' use and the benefits that they might be missing.

We believe that the design of a corporate wiki should be grounded on the understanding of the knowledge sharing ecology in organizations (see Design Rationale section) and should provide a realistic story for adoption (see User Interfaces section). In line with this vision, drawing on our observations and prior literature, we built the *Mail2Wiki system* to reduce interaction costs and integrate wikis with email. We designed the system to enable incremental adoption, where using the simplest functionalities requires the simplest interactions at the lowest cost. The system

C&T'11, 29 June – 2 July 2011, QUT, Brisbane, Australia.
ACM 978-1-4503-0824-3

includes two front-ends to give access to incremental levels of system functionality. This *built-in adoption story* accentuates the amelioration of interaction costs and tool integration. Note that while these pro-adoption solutions are embodied here with enterprise wikis, they might be later re-applied to other common Enterprise 2.0 tools.

We have already presented a brief demonstration of some of the functionalities in the Mail2Wiki system in [18]. In this paper, our contributions include the design rationale, a detailed description of the architecture and system design, and our evaluation of the most novel of the front-ends.

RELATED WORK

The adoption problem

Recent studies of Enterprise 2.0 tools [23] deployed in corporate intranets, including wikis and blogs, point to low contribution rates. These rates range between about 2-15%. The reported rates of contribution reported for *corporate blogs* in large companies such as IBM [13] and Microsoft [14] are about 3%. About 2-4% of the IBM workforce visited the Beehive social networking site monthly, while about 15% had registered over two years [15]. Similar participation rates for social media were observed at HP, with large variability across tools, countries (10% in UK vs. 1.9% in Japan and Mexico), job functions, groups, and managerial activity level. Blogs and wikis were used 10 and 28 times less, respectively, than the discussion forums [6].

A few studies have focused on *enterprise wikis*. The studies, conducted at MITRE [19], IBM [1], and Microsoft [17], suggest that while many workers report viewing the wikis, very few actually contribute. Danis and Singer [10] studied an enterprise wiki in a 900-member research organization and found that viewers of the wiki were reluctant to edit and modify others' content. Similarly,

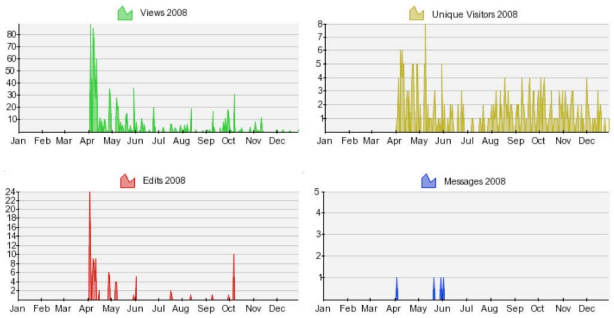

Figure 1. Community wiki logs. The top chart shows the decreasing number of views and visitors from the launch of this wiki in April 2008 (about 80 members). The bottom chart shows the very few edits and messages. We observed multiple community wikis with similar trends [9].

Grudin and Poole [17], who studied wikis at Microsoft, found that numerous wikis had been created (mainly for small groups) but most were quickly abandoned. They identified three challenges for adoption and sustainability of enterprise wikis: (1) positioning the wiki in an existing information ecology and corporate culture, (2) the high costs for content organization, and (3) aligning expectations between managers and individual contributors. In this paper, we address the first two challenges.

Related Tools

Several *tools for contributing to shared repositories* have focused on making contributions to a repository easier. Posterous (http://posterous.com) is a consumer tool that makes blogging easier by allowing users to email contributions to a server that publishes the content. Mail2Tag [27] is a corporate blog with tagging, browsing, and search functions. The users can email their content to tags and the system enables folksonomy-style organization. Mail2Wiki differentiates from these systems by integrating directly with the email client, enabling early organization of contributions in existing pages and sections, page and section generation, and recommendations.

Previous *tools extending email clients* have supported collaborative activities around e-mail. Systems such as XOBNI (http://www.xobni.com), Meshin (http://www.meshin.com) and Salsa [35] achieve this by providing insights about the worker's inbox from outside sources. These tools are all centripetal to email, as they pull in relevant information from various sources and past activity while relating it to emails in an inbox. In contrast, Mail2Wiki, by integrating a shared space, is *centrifugal to email*. Mail2Wiki is intended to embed sharing functions directly in email, offloading content to the shared space so it may be crystallized and reused collaboratively later on.

There are various recommenders and intelligent agents to support activities such as searching and browsing. Montaner et al. gave a detailed survey of Internet systems [24]. In Montaner's taxonomy, our system is a *document recommender* based on an *adaptive profile-item matching* technique. We base our algorithm on the idea that any ranking algorithm can be obtained by applying a discriminative binary classifier to a dataset [4, 26]. In Mail2Wiki, logistic regression was used as the base discriminative classifier with stochastic gradient descent learning [5], enabling real time parameter updates. To match pages with emails we used the Okapi BM25 similarity measure [16].

Several advanced email *summarization techniques* have been developed over the last few years. For example, Rambow et al. [33] described a method to select interesting email parts to summarize email threads. Another example, given by Dredze et al. [11] is a technique to summarize a list of emails using a set of carefully selected keywords. These techniques focus on very specific types of pages and may lose a significant amount of. In our approach, the emails are not summarized, but simply grouped into section and/or subsection, but displayed entirely, after removal of verbose redundant text, such as email replies.

DESIGN RATIONALE

While Web 2.0 tools have moved from the consumer [29] to enterprise space [23], their design has not yet adapted to this new domain. Specific factors that impact enterprise requirements include centrality of email, limited time of workers, issues of credit and motivation [9], as well as addressing, high interaction costs [8, 17, 19] and the poor integration [2]. Specifically, we believe that the design rationale of an enterprise wiki should be grounded on understanding the enterprise knowledge sharing ecology and provide a realistic story for adoption. This rationale is the focus of this section.

Understanding Sharing Communities

Based on an analysis of online communities supported by public Web 2.0 tools, Preece and Shneiderman [31] proposed the Reader-to-Leader framework. This framework describes how community members can be roughly categorized in classes that reflect levels of participation: readers, contributors, collaborators, and leaders. They further use their framework to identify design changes that may encourage members to move to higher levels. For example, designers can promote greater participation by lowering the threshold for making small contributions (e.g., no login) or giving visibility to contributors', collaborators', or leaders' work via a reputation system.

We adapt the Reader-to-Leader framework to characterize the community of workers sharing knowledge via enterprise wikis and email. We refer to the different levels of participation observed in terms of *informal roles* that the workers can take in relation to wiki-based sharing.

Understanding Sharing Communities in the Enterprise

In a prior field study of communities of professionals [8], we analyzed 15 enterprise wikis in two organizations: a large business organization and a research center. We surveyed the type of shared information and collected various usage statistics, such as views and edits over time (Figure 1). These observations confirmed low adoption rates of wikis. Drawing on these observations and prior studies, we identify four main roles of workers in regards to enterprise wikis: curators, simple contributors, lurkers (or readers), and unengaged workers (or non-users).

Only a few workers take the role of *curator* in an organization. These individuals aggregate and package information on wikis. For example, we interviewed a manager who regularly updated and maintained the wiki for Bid Managers [8]. This role is analogous to the curator role observed in enterprise file-sharing services [25].

We observed various instances of the role of *contributor*. In this role, users make isolated contributions to the wiki, but do not make changes to the structure of the content. This role was especially visible when a *curator* had created a skeletal page for others to populate.

The *lurker* refers to individuals who consume the content of the enterprise wiki for individual benefit but make no contributions. The emergence of this role is analogous to the lurker within enterprise social networks [15].

Finally, due to the small number of active participants, we concerned ourselves with the *unengaged* worker. These workers often see no utility or benefit in enterprise wikis. Many of the interactions of this role are indirect, such as consuming benefits of already curated content or indirectly contributing content through out of band communications with more involved coworkers.

Overall, these four roles provide a more complete view of the enterprise wiki lifecycle in which knowledge is contributed, curated, and consumed. In Figure 2, we display the roles: the unengaged and lurkers are at the bottom producing raw information outside of the wiki; the middle layer is occupied by the simple contributors who sift through this raw information and transform it into semi-structured information; and the top layer contains the curators, who create the final document from the semi-structured data. As the level in the diagram increases, the number of people occupying that level decreases (generally, exponentially, see power law distributions in [6]).

Designing for Sharing Communities in the Enterprise

A key aspect that affects the sustainability and efficiency of this process is the effort and reward lifecycle for the various roles. *Readers* gain rewards in the form of new knowledge by consuming the curated information (and face interaction costs for consuming). *Curators and Contributors* gain rewards in the form of credit and attention for curating and contributing information (and face costs for producing).

Other factors characterizing the community (e.g., geographic distribution) or the organization (e.g., corporate culture) can also facilitate or hinder the lifecycle. Our observations together with prior studies suggest there are a few evident breakdowns in this lifecycle in current tools.

First, the cost threshold for lurkers and simple contributors to easily contribute content is too high. Many steps and context switches are involved for posting content on a wiki. This is an obstacle both for readers to become contributors,

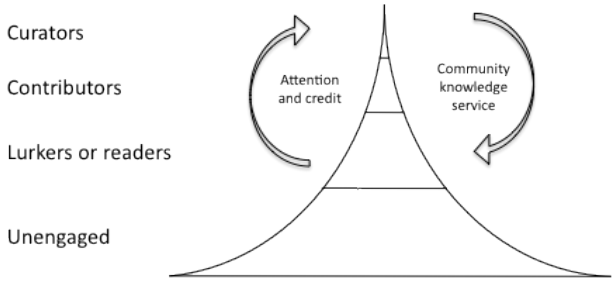

Figure 2. Lifecycle. Attention and rewards are given to individuals with higher involvement in return for their community knowledge service. The population decreases (exponentially) as the level of participation increases.

and for contributors to increase the frequency of their

contributions. Moreover, the work of curators is made difficult by the lack of tools that allow easy reorganization of the content, see our related system, VisualWikiCurator [20, 21] for details on how we combat this problem.

Second, the current tools for sharing are not properly integrated with core work tools. Email, which is central to many workflows [12], keeps knowledge trapped in inboxes. On the other hand, wikis support sharing and reuse but are disconnected from the majority of current workflows. Integration is further hindered as current workers may have multiple unrelated wikis, all of which require them to login every time they wish to make an edit.

Both breakdowns create disincentives not only for those who might want to contribute or curate, but also for those who might want to simply consume content. This, in turn, reduces both the potential pool of contributors, as well as, the incentives for those currently engaged.

Finally, we speculate that the many observed failures of wiki adoption [8, 17, 19] suggest that the design of the current corporate wikis also lack a realistic adoption story. In contrast with the simplicity and success of email, wikis have failed to scale the effort of the different tasks and interactions. That is, making simple contributions should require only simple interactions and minimal effort from the user. Performing more powerful data manipulation may require more complex interactions and increased effort.

To mitigate these breakdowns we identified key requirements, which are incorporated into the Mail2Wiki design. Specifically, we focused on reducing the costs for sharing and early curation, making contributing easier. Simultaneously, we focused on increasing the level of integration between wikis and email, making both contribution and consumption easier. Moreover, we designed for incremental adoption: simple functionalities require simple interactions.

MAIL2WIKI ARCHITECTURE

The Mail2Wiki system is designed to sit between two repositories, personal email and shared wikis. In our architecture we treat both of these repositories as resources. Figure 3 shows the overview of our architecture. The different layers represent logical separations of our system. The transactions between the layers are also provided.

In framing the overall architecture of our system we liken it to a Model-View-Controller design pattern. The two resources (email and wikis) provide the layer that is analogous to the model. The communication layer and language processing engine function as controllers, handling requests from our views and serving the necessary information about the model. Finally, our two front-ends are analogous to views. This design allows us to use the same algorithms and concepts for different repositories. For instance, the personal repository could be web pages being viewed in a browser, and the shared repository could be a discussion thread or additional wiki engines.

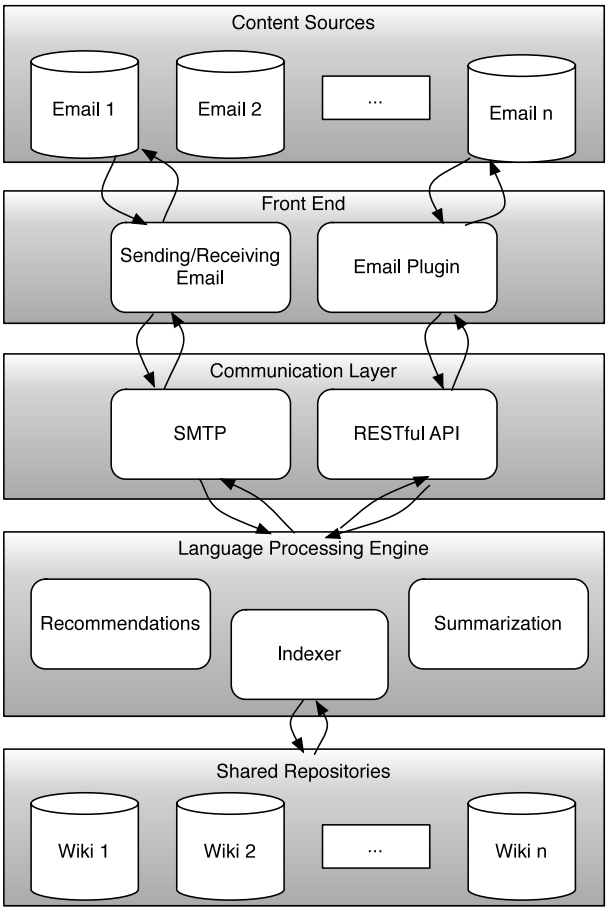

Figure 3. Mail2Wiki software architecture.

Language Processing Engine Architecture

The Language Processing Engine is a dedicated server for the natural language processing required to support our two interfaces. The engine comprises three modules: recommendation, indexing, and page generation.

Recommendation module

The recommendation module assists the user in finding the correct target for contributing information. This functionality is used by both interfaces, either by sending an email to the system requesting recommendations or via interactions with the Outlook Plugin. The engine performs a matching between the content being considered for contribution and the potential targets in the wiki. The recommendation algorithm was implemented as a binary classifier applied on all *the* possible *(content, target)* pairs, where *content* denotes the set of emails or text being submitted and *target* loops over all the possible wiki pages and sections. When the number of possible targets is too big to be processed in real time, the system applies an initial pruning step based on the BM25 score to reduce the number of targets to a fixed number. The binary classifier takes into account several features such as: Similarity between content and target (using standard similarity scores between text, including BM25 [34]); authorship of the content; and authorship of already published content in the target.

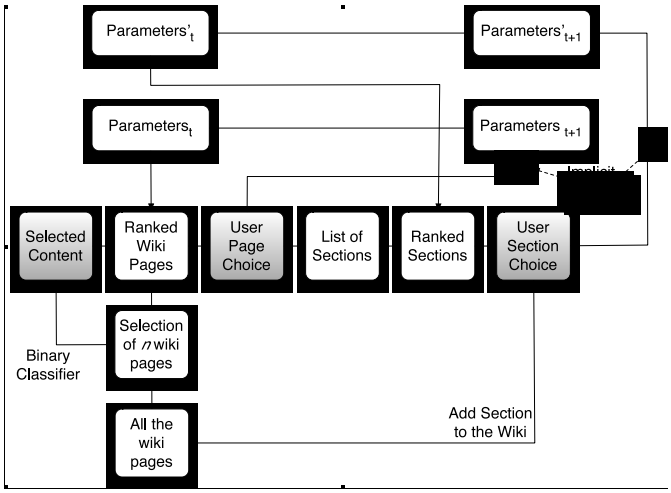

Figure 4. Adaptive recommendation module. Every time the user interacts with a recommendation, the parameters are updated using a stochastic gradient step in the logistic regression classification model.

Indexing

Since recommendations must be made in real time, the system maintains a persistent snapshot of the wiki and a bag-of-word representation of every page and section of these pages. When deployed on an already existing wiki, the system first creates a global index. To keep this index up-to-date with the wiki, the system uses a number of triggers on the wiki. These triggers refresh the snapshot through interactions with the RESTful API. Scheduled indexes are also utilized to capture any edits committed without the knowledge of our system.

Page generation (Summarization of emails)

When an automated update to a wiki page is requested, a batch of emails for instance, the page generation module is activated. To generate the page or section, the submitted data must be merged with existing content. To accomplish this a filtering step is first performed, where the relevant content is clustered into sections that may share the same subject. Every identified cluster is represented by one subsection of the generated wiki page. The content of these subsections is a simple concatenation of the filtered email content that was assigned to each cluster, ordered by date.

After this initial merging step, extra sections are added on the wiki page. Each extra section is defined in a Python subroutine that takes the old content and new emails as input and outputs the new content of the section. Currently, we have implemented the following subroutines:

- **References**: A list relevant metadata from the emails, such as authors, recipients, and date.
- **People**: A list of people retrieved from email metadata (e.g., author) and extracted from the email text.
- **Social Graph**: The social graph constructed from the People subroutine is graphed. This requires the graphviz plugin of MediaWiki. The number of exchanges between the two persons weights every link in the graph.
- **Related Pages**: A list of wiki pages is populated by using the results of the recommendation module.
- **External Links**: A list of the external links (e.g. http://something.com) is collected from the emails.
- **Attachments**: A list of email attachments is also provided.

Front-end Architecture

We have two front-ends: a thin and thick front-end. Both have access to similar functionality. The thin front-end only involves email interaction with our system. This interface requires no installation by the user and is based on the interface of Mail2Tag [27].

The thick front-end, an Outlook plugin, provides quicker feedback by establishing a more direct connection. Instead of email communication, the plugin uses RESTful APIs to directly communicate with the back-end.

Integration with other sharing tools

Our system is also able to integrate with other tools. As one example, it currently supports a wiki plugin that aims to reduce the cost of organizing wiki pages for curators. This plugin embeds relevant external content in each page using our back-end's recommendation feature, and displays extracted entities to give curators multiple views [20].

USER INTERFACES

We developed two interfaces to improve the enterprise wiki adoption story. First, our thin front-end requires no end user configuration or installation, lowering the barrier for *unengaged* users. This interface focuses on simple interactions and supports the addition of sections and pages. Our thick front-end, the Outlook plugin, requires minimal end user installation and configuration. This interface focuses on making simple contributions easy, such as contributions from email content. While more complex actions, like the addition of pages and editing of existing content, require more deliberate, complex interactions. These two interfaces are discussed in more detail below.

Thin front-end: Without-Plugin Interface

In order to engage non-users, we need to leverage their existing practices. For inspiration we looked to the Mail2Tag system [27], where workers email content using a naming convention that tags the content and places it in a searchable, shared repository. Mail2Tag has seen success in its current deployment within our organization and also fit our criteria for leveraging an existing practice (in this case, email). Enabling the editing of content by extending the metaphor to wikis seemed a natural progression. An additional benefit of this front-end is to be email-client agnostic, making our system at least minimally accessible to all of the workers within an organization.

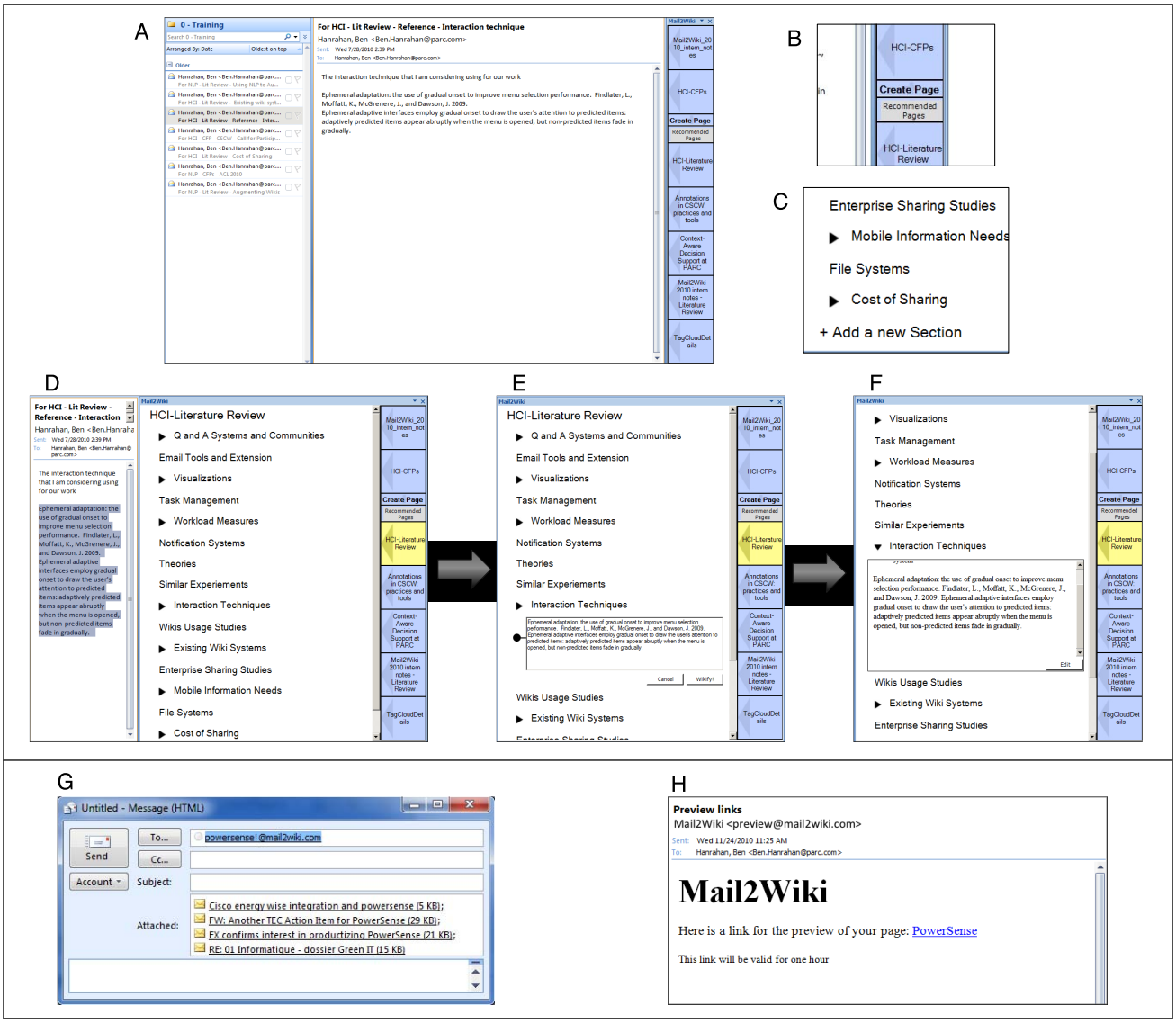

Figure 5. The two front-ends of Mail2Wiki (A-F Screenshots of Thick front-end, G-H Screenshots of Thin front-end)

We extend the Mail2Tag formalism used to interact through emails in a few specific ways. First, we use the wiki page name as the email recipient, instead of a tag (e.g., example@mail2wiki.com). Furthermore, users can attach batches of emails to system-bound mail. Once the server receives the mail, it sends back a confirmation that provides a preview of the user's additions to the page, or new page that they can decide to publish or edit. Users can skip the confirmation step by appending an exclamation mark to the page name (e.g., example!@mail2wiki.com). The user can also request page recommendations by sending a question mark in place of the page name (e.g., ?@mail2wiki.com). The system will respond with several page recommendations. An example generated page is provided in Figure 6.

Thick front-end: Outlook Plugin Interface

We chose to develop a plugin for Microsoft Outlook, as it is the most commonly used email client amongst knowledge workers. This aim of this interface (Figure 5A-F) is to make simple contributions simple to execute, achieved via drag and drop of text snippets from email (Figure 5D-F).

To assist with obtaining the appropriate target for a contribution, the tool provides a view with relevant wiki pages (Figure 5A Right), including user-selected favorites and recommended wiki pages. The worker can glance over the list of targets to decide whether to contribute.

Additionally, the tool assists the worker in finding the appropriate section (Figure 5E) within the page by providing two features. First, it gives an overview that the worker can quickly scan. Second, the tool suggests sections on the overview based on the current content.

To ease the contribution and editing process, the costs for transferring content are reduced by supporting direct manipulation (drag & drop) and editing text directly within the page outline. The user can select text from an email

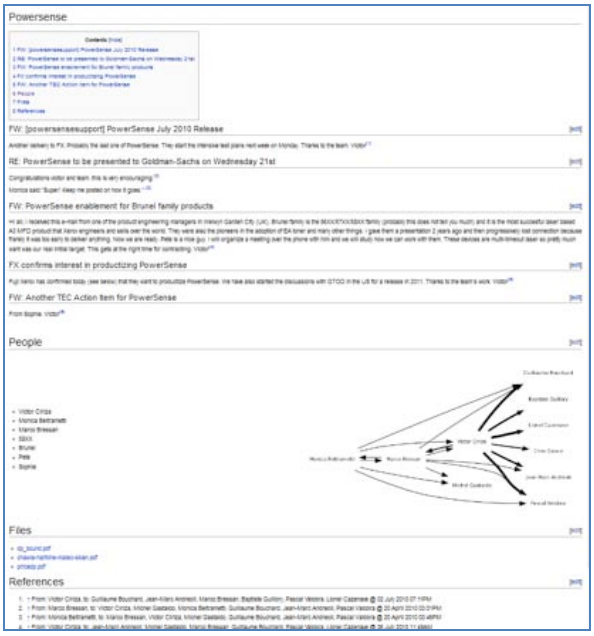

Figure 6. An example of a generated page.

(Figure 5D) and drag it to the appropriate section header (Figure 5E). The tool appends the content to the section (or page) and immediately displays the result within the outline (Figure 5F). The user can also briefly edit or curate the contribution and the existing text. In this manner, workers are able to put as much effort into contributions as desired.

Through this usage metaphor user can also create pages (Figure 5B) and sections (Figure 5C), either manually by clicking or automatically by dragging a batch of emails onto the correct target. In this manner we make more complex interactions only marginally more expensive.

Design development process

As Mail2Wiki focuses on making wikis more inclusive and reducing the cost of making contributions, we felt that the usability and intuitiveness of the tool was of paramount importance. To accomplish this we performed multiple iterations in a participatory design process.

First, we had four users evaluate interaction storyboards derived from insights gained from our previous fieldwork. This evaluation validated some of our basic decisions, such as providing drag and drop mechanisms for sharing.

Second, we performed an interview and usability study with an early prototype, where nine potential users were given the opportunity to use the prototype and request changes or additions. This stage further informed our design through key findings about wiki usage and desired functionality. Based on our observations, we added recommendations for pages and sections, viewing and editing existing content, and the ability to hide the outline. Figure 5 depicts the current iteration.

Lab evaluation

To further validate our design we performed a lab study to specifically test two hypotheses: 1) The tool will reduce total interaction costs of contributing and initial organization of content. 2) The tool will specifically reduce the interaction costs of transferring content.

Method

We used a 2 (tool) x 2 (page complexity) within-subject experimental design. Participants made contributions both with and without the tool. We also tested both simple and complex pages. A fully crossed design with 2 contributions per condition resulted in 2x2x2=8 contributions per subject.

In the with-tool condition, the four pages were immediately available in the interface. We disabled the recommendation feature to measure the impact of our interaction techniques only. In the without-tool condition, we configured Mozilla Firefox to have the homepage set to the main page of the wiki. From here, the participants navigated to a group page, project page, and finally the target page. This navigational stage was based both on our interviews about current wiki usage and the work of Phuwanartnurak [32], who reported on the existence of many project pages. In our analysis, we considered the without-tool condition both with and without this navigational stage.

Subjects: We ran our study on a Tobii X120 desk mounted eye-tracker and recruited 14 knowledge workers in a research organization (7 male 7 female, age: M=43, SD=11.3).

Material: A simple page consisted of three sections, whereas a complex page consisted of fifteen sections. The appropriate section for the contribution on complex pages was below-the-fold of the initial view. We created two simple and two complex pages and associated each with one of two *projects*: Human Computer Interaction (HCI) and Natural Language Processing (NLP). Each project included a Call for Participation page and a Literature Review page.

Tasks and Procedure: In both tool conditions, participants were presented with e-mails and asked to make a contribution to the appropriate page as quickly as possible, without worrying about the quality of formatting. A contribution consisted of two or three lines of text at the top of the email. Each email gave information about the project, page, and section where a contribution should be made. We measured task completion time, number of clicks, and number of eye fixations. We measured mental workload using the NASA TLX after each task set (or tool condition).

We accounted for learning effects by incorporating a training period at the beginning of the evaluation and alternating the tool conditions (with and without) between subjects. We gave an advantage to complex pages by presenting two simple pages then two complex pages as the targets of the contribution.

Analysis: Each performance measure was further divided into three stages: reading, navigating, and editing stages. We coded the eye-tracking screen captures to retrieve each

stage's performance measurements. The reading stage was defined by the beginning of the task and the disengaging from the email. The navigation stage existed only in the without-tool condition and was defined by the end of the reading stage and reaching the target page. The editing stage was defined by reaching the target page and the verbal indication of task completion.

Results

We performed a Repeated Measures ANOVA with Tool and Page Complexity as factors. We used Project as a control variable because participants could have prior knowledge in one of the areas and to validate that the effect of the tool generalizes beyond specific content sets.

For completion time, eye fixations, and mouse clicks we compared the total time required to 1) complete all stages, 2) complete all stages excluding the navigation stage, and 3) complete the editing stage. The NASA TLX measure was collected after each task group and measures the effect of tool only. Figure 7 summarizes our results.

The tool had a clear and strong effect on Completion Time (Figure 7, top). Average time for all stages indicated that the with-tool condition was significantly faster (M = 34.6, SD = 1.9) than without (M = 61.5, SD = 4.2), $F(1, 13)$ = 14.4, $p < .001$. Even when we excluded the navigation stage, the average time spent was lower (M = 34.6, SD = 1.9) than without (M = 44.9, SD = 3.7), $F(1,13)$ = 6.4, $p <$ .05. The average time spent was also lower when considering the editing stage alone (M = 21.3, SD = 1.1) than without (M = 31.1, SD = 2.9), $F(1,13)$ = 9.2, $p < .01$.

Similar to the effects on time, the measure of the number of clicks (Figure 7, middle) indicated that with the tool the users performed fewer clicks across all the stages (M = 7.1, SD = 0.5) than without (M = 14.3, SD = 0.7), $F(1,13)$ = 59.9, $p < .001$, all the stages minus the navigation stage (M = 7.1, SD = 0.5) than without (M = 9.6, SD = 0.6), $F(1,13)$ = 8.8, $p < .01$, and during the editing stage alone (M = 4.1, SD =0.6) than without the tool (M=6.3, SD = 0.6), $F(1,13)$ = 5.5, p < .05.

The average number of eye fixations over all stages (Figure 7, bottom) was significantly lower with the tool (M = 85.9, SD = 6.1) than without (M = 128.8, SD = 9.5) , $F(1,13)$ = 18.4, $p < .001$). In the other cases, while the number of fixations trended consistently, the measured effect did not reach significance. Finally, the Perceived Mental Demand from the NASA TLX instrument was lower with the tool (M = 3.2, SD = 0.5) than without (M = 4.3, SD = 0.5), $F(1,13)$ = 8, $p < .01$. Other measures did not reach significance.

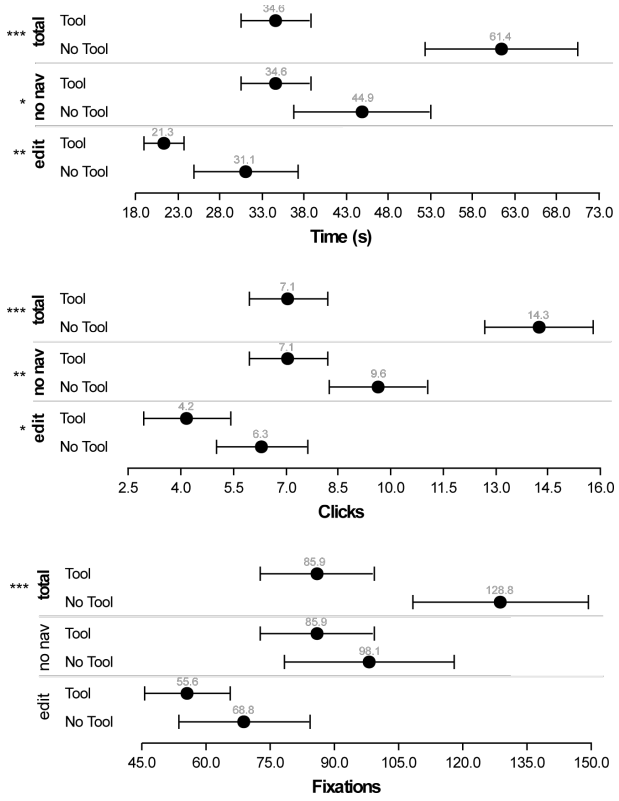

Figure 7. The results of our evaluation. The significance for $p < .05$, $p < .01$, and $p < .001$ is indicated by a *, **, and * respectively. "Total" includes all three stages, "no nav" excludes navigation, and "edit" includes only editing stage.**

DISCUSSION AND CONCLUSION

Novel aspects of the front-ends.

In the thin front-end, we extend the design previously proposed in Mail2Tag [27], which builds on existing user practices (i.e., emailing useful content to colleagues). As users contribute to tags in Mail2Tag, Mail2Wiki users contribute to wiki pages. In addition, Mail2Wiki introduces a preview step for the user to validate the new content. This is important to mitigate the distribution of sensitive content. The user can also ask the system for recommended wiki pages.

In the thick front-end, the same functionalities mentioned above (contributing to a page, validating, and recommended wiki pages) are supported in the context of real-time interaction and direct manipulation. The email plugin introduces a novel interaction technique that extends an email client into a shared wiki space. The same technique could extend other tools such as a chat client, web browser, RSS feed aggregator, or word processor. Using this technique the user can view the structure of a chosen shared space (wiki page), without switching to another tool and transfer content to the correct portion in the structure.

Both front-ends allow selecting a batch of emails and contributing these to a new or existing wiki page (or section), for example generating a project page based on all the emails exchanged about that project.

All these new functionalities are aimed at lowering the cost of sharing and curation on the wiki and increasing email-to-wiki integration. Both front-ends enable direct interaction with the information in the wiki, beyond just previewing

related wiki documents (as Google Ads preview related sites).

Observed and expected impact of Mail2Wiki

The designs of both of our interfaces have a basis on empirical user data. The thin front-end extends the Mail2Tags interface, which has shown success as a deployment in an organization. The system has shown a good level of adoption in relation to similar Enterprise 2.0 systems [28]. The thick front-end (Outlook plugin) was based on multiple iterations of user-centric design. Furthermore, the results of our lab evaluation provide evidence that our system enables contributions at a significantly lower cost, even excluding the time spent for navigation. We expect that this reduction in interaction costs will result in higher contribution rates to wikis and more efficient knowledge reuse.

The page recommendation function is accessible via both front-ends. A key benefit expected for this function is an increase in the workers' awareness of the content and an increase in participation with additional groups in their organization. We have not evaluated the recommendation functionality yet; however, we believe that prior work such as BlogMuse [13] and Topika [22] and the details included about our implementation in this paper provides enough preliminary evidence about the expected utility of the recommendation function in Mail2Wiki.

Limitation of the evaluation and generalizability

The evaluation presented has only focused on the thick front-end. Focusing on a part of the system was a necessary first step, given the broad range of functionalities. In the lab study we focused on the reduction of interaction costs in the simplest case of contributions to known wiki pages, which is comparable to the current "without tool" conditions. The evaluation of the recommendations requires a separate study design with different conditions, which we are currently planning. An additional factor not accounted for in the evaluation is contribution quality. We plan to evaluate this in a future, longitudinal evaluation. The primary aim of this paper was presenting the design rationale, grounded in prior work, and the implementation of Mail2Wiki.

We acknowledge that in order to increase the ecological validity of our results the lab study needs to be complemented with future observations of users in the field. However, given that email is such a critical tool in the workplace [12], we had to first develop and validate our tools iteratively in a controlled setting. We incorporated multiple rounds of feedback from knowledge workers, after which we validated our design with a focused lab evaluation. We learned (based on our fieldwork and prior studies) that the high interaction costs and the poor integration with core work tools such as email are significant impediments to wiki adoption. Therefore, our study aimed at proving that we could mitigate a primary deterrent in contributing to a wiki. Even when we excluded the navigation time in our evaluation, we observed a gain in efficiency with out thick front-end. The results of the multiple measures used in this evaluation were informative and non-obvious.

FUTURE WORK

For the *future evaluation* of Mail2Wiki, we plan to study both front-ends in use by a pilot community of professionals to measure the impact on wiki adoption. This field evaluation will also examine the recommendation and page generation functionalities.

For *future development* of functionalities, a line of research is to improve the page generation module by learning how to generate a page from the history of edits, which is already being stored by most wiki systems. Our objective is to modify the Mail2Wiki page generation to use an adaptive system [3]. In this system, Wikipedia pages are automatically generated based on search results associated to the title of the page. We would like to use the same technique to transform emails instead of search results into a wiki page.

Current corporate wikis tend to have a top-down structure: they are often organized in categories and sub-categories. However, if many of the pages are created using Mail2Wiki the wiki would have a flat-hierarchy. We plan to develop a system that identifies these categories and generates clusters of pages. These clusters will be used to make meta-pages, lists of pages that are associated with the categories. Additionally, these pages will have a summary of the content in common (e.g. keywords, list of recurrent authors, etc.). These generated meta-pages are expected to improve discovery and re-finding of content on the wiki by automatically linking pages together.

REFERENCES

1. Arazy, O., Gellatly, I., Soobaek J., Patterson, R., (2010). Wiki deployment in corporate settings. In Technology and Society Magazine, IEEE, 28(2) (2009), 57-64.
2. Balakrishnan, A. D., Matthews, T., and Moran, T. P. Fitting an activity-centric system into an ecology of workplace tools. In Proc. CHI 2010. ACM 787–790.
3. Barzilay R., Sauper C. (2009). Automatically Generating Wikipedia Articles: A Structure-Aware Approach. In *Proc. of ACL '09*.
4. Billsus, D. & Pazzani, M. J. (1998). Learning Collaborative Information Filters. *In Proceedings of the ICML*. Madison, WI: Morgan Kaufmann Publishers.
5. Bottou, L. (2003) Stochastic Learning. *Advanced Lectures on Machine Learning*: 146-168
6. Brzozowski, M. J., Sandholm, T., and Hogg, T. (2009). Effects of feedback and peer pressure on contributions to enterprise social media. Proc. GROUP '09. 61-70.
7. Buffa M., Intranet Wikis, in Proc. of the Intraweb Workshop, Proceedings of WWW 2006, ACM Press.

8. Convertino, G. Kairam, S. Chi, E.H., Grasso, A, Pirolli, P. Stricker, T. and Bascaran, E. (2010) Designing for Learning Communities in a Large Enterprise. CSCW 2010.
9. Convertino, G. Grasso, A. De Michelis, G. Millen, D. Chi, E.H. (2010) Clorg: collective intelligence in organizations. In *Proceedings of* GROUP '10.
10. Danis, C. & Singer, D. (2008). A wiki instance in the enterprise: Opportunities, concerns, and reality. Proc. CSCW'08, 495-504.
11. Dredze, M., Wallach, H. M., Puller, D. and Pereira, F. J. (2008). Generating summary keywords for emails using topics. *IUI 2008*. pages 199–206
12. Ducheneaut, N., Bellotti. V. (2001). E-mail as habitat: an exploration of embedded personal information management. Interactions, ACM. 8(5): 30–38.
13. Dugan, C., Geyer, W., and Millen, D. R. (2010). Lessons learned from blog muse: audience-based inspiration for bloggers. In Proc. of CHI 2010. ACM, New York, NY, 1965-1974.
14. Efimova, L. and Grudin, J. 2007. Crossing Boundaries: A Case Study of Employee Blogging. In Proc. of HICSS. IEEE Computer Society, Washington, DC, 86.
15. Farzan R., DiMicco JM, Brownholtz B. (2010) Mobilizing Lurkers with a Targeted Task. Proc. of ICWSM '10. May 2010.
16. Jones K. S., Walker, S. and Stephen, E. (2000). A Probabilistic Model of Information Retrieval: Development and Comparative Experiments. *Informat. Processing and Management*, 36(6): 779-840. 2000.
17. Grudin, J. and Poole, E. S. (2010). Wikis at work: success factors and challenges for sustainability of enterprise Wikis. Proc. of Wikisyms 2009, ACM, NY.
18. Hanrahan, B. Weksteen, T. Kong, N. Convertino, G. Bouchard, G. Archambeau, C. Chi, E. H. Mail2Wiki: Posting and Curating Wiki Content from Email. IUI2011. Demo.
19. Holtzblatt, L. J., Damianos, L. E., and Weiss, D. Factors impeding Wiki use in the enterprise: a case study. In Proceedings of CHI '10, ACM, 4661-4676.
20. Kong, N. Hanrahan, B. Weksteen, T. Convertino, G. Chi, E. H. VisualWikiCurator: Human and Machine Intelligence for Organizing Wiki Content. IUI2011.
21. Kong, N.; Convertino, G.; Hanrahan, B.; Chi, E. H. (2011) VisualWikiCurator: Mixed Initiative Interaction for Content Curation. To appear as CHI 2011 Work-In-Progress, extended abstract and poster. May 7-12 2011, Vancouver, Canada.
22. Mahmud J., Matthews T., Whittaker S., Moran T.P., Lau T., (2011) Topika: Integrating Collaborative Sharing with Email. Proceedings of CHI 2011. ACM Press.
23. McAfee, A. Enterprise 2.0: New collaborative tools for your organization's toughest challenges. Harvard Business Press. Boston, MA. 2009
24. Montaner, M.; Lopez, B.; de la Rosa, J. L. (2003), A Taxonomy of Recommender Agents on the Internet, Artificial Intelligence Review 19 (4): 285–330
25. Muller, M., Millen, D.R., & Feinberg, J., Information curators in an enterprise file-sharing service. Proc. ECSCW 2009.
26. Nallapati, R. (2004). Discriminative Models for Information Retrieval. In *proceedings of ACM-SIGIR*.
27. Nelson,L. Nairn, R. Convertino, G. Chi, E. H. (2011)_Mail2Tag: implicit organizing in organizational sharing. To appear in the proceedings of the International Symposium on Collaborative Technologies and Systems 2011 (CTS 2011).
28. Nelson L., Convertino G. (Under review) Measuring the Adoption of a Lightweight Enterprise2.0 Tool for Sharing. *Also* to appear at CHI 2011 workshop W20: Appropriation and Creative Use: Linking User Studies & Design. www.hiit.fi/~asalovaa/chi2011-appropriation (last accessed on March 18 2011), May 7-12 2011, Vancouver, Canada.
29. O'reilly, T., (2007). What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. Communications & Strategies, 1, p. 17.
30. Palen, L. and Grudin, J. 2003. Discretionary adoption of group support software: lessons from calendar applications. In Implementing Collaboration Technologies in industry: Case Examples and Lessons Learned, Munkvold, Ed. Springer., London, 159-179.
31. Preece, J. and B. Shneiderman. 2009. The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation, AIS Transac. on HCI, (1) 13-32
32. Phuwanartnurak, A. J. 2009. Did you put it on the wiki?: information sharing through wikis in interdisciplinary design collaboration. SIGDOC 2009, 273–280.
33. Rambow, O., Shrestha, O, L., Chen, L, J., Lauridsen, J and C. (2004). Summarizing email threads. *Proc. of HLT-NAACL 2004: Short Papers on XX*. pages 105-108.
34. Robertson S., Zaragoza H., Taylor M.(2004) Simple BM25 Extension to Multiple Weighted Fields.CIKM'04.
35. Stecher, K., Counts, Cheng, S., L., Williams, S., and Turski, A. Salsa: Leveraging Email to Create a Social Network for the Enterprise. ICWSM 2009, 323–326.
36. Wilkinson D. M. Strong regularities in online peer production. In Proc. of the 9th ACM conf. on Electronic commerce (EC'08), NY, USA, 2008. ACM. 302–309.
37. Whittaker, S. and Sidner, C. Email overload: exploring personal information management of email. In Proc. CHI '96, ACM Press (1996), 276–283.