

A Design Study of the Integration of Email and Role Management for University Students

H. Ross Baker, Nicolas B. Duarte, Aydin Haririnia, Dawn E. Klinessmith, Hannah Lee, Leonid A. Velikovich, Alfred O. Wanga, Matthew J. Westhoff, Catherine Plaisant*

University of Maryland, College Park, MD 20742

* Human-Computer Interaction Laboratory

plaisant@cs.umd.edu

ABSTRACT

In order to accommodate the increasing diversity of email users, applications have evolved in both functionality and user interface. In this study, we attempt to determine whether email user interfaces can be improved to serve a specific target population: college students. We present our results from college campus surveys that examine email usage patterns and subjective experiences among college students. From our survey feedback and related research, we conclude that email overload and feature intimidation are the greatest hindrances to email communication on campus. To address these problems, we propose employing role management to organize messages calendar and contacts in an email program for students, using school, work and family roles. We describe a prototype and user reactions. Our conclusion is that role management, integrated into email software, may help college students manage their email more effectively.

1. INTRODUCTION

Email use is becoming increasingly widespread among the general population [1], and particularly among college students. However, surveys suggest that college students are not adequately served by current email software. The problems

discovered include difficulty organizing a large number of email messages and contacts, as well as failure to use email application features due to poor exposure and perceived complexity.

We propose to address these problems by exploiting the categorical nature of college students' email correspondence. The contacts and messages involved in intra-college communication often belong to well-defined groups (e.g. students and professors in specific classes), which are known ahead of time and can be communicated automatically. This knowledge permits an email program to automatically organize many of the messages and contacts by grouping them visually for the user. Furthermore, additional functionalities such as integrated class directory listings and event calendars become possible with the requisite back-end support.

The functionality described above is designed to aid a student by making use of a "school role". When dealing with other messages, it may be useful to introduce a "work role" or "family role". Role management in this paper refers to the organization of email-related information by the main roles assumed by users. Its potential benefits include filtering of relevant information by role/subrole and customization within a given role.

Much recent work has been dedicated to facilitating *task* (or activity) management in email by grouping together *temporally* and *semantically* related messages and/or resources (e.g. a conversation thread or documents related to a specific task). While this method of organization appears promising, it currently targets mostly users in a business environment. Business users typically engage in a variety of ephemeral *tasks* related almost exclusively to their work role [2], but college students often assume a multitude of *roles* (school, work, family, etc.) and subroles (different classes) which are relatively permanent. We propose the alternative strategy of *role* management in a college environment to reflect the nature of such correspondence.

2. PREVIOUS WORK

The US Department of Commerce surveys show email use among the general US population at 45.2% in 2002, up from 35.4% in 2000 [3]. College students represent a continuation of this trend. A study by the Pew Internet and American Life Project in 2002 indicates that “college students are heavy users of the Internet compared to the general population... in part because they have grown up with computers. [The Internet] is integrated into their daily communication habits and has become a technology as ordinary as the telephone or television” [4].

Along with its increasing number of users, email has been used in an increasing number of ways. In their 1996 study of email overload, Whittaker and Sidner observed that people were using email for task management and personal archiving [5]. They describe the goals of task management as “[ensuring] that information relating to current tasks is readily available”. The researchers conclude from a study of Lotus

NotesMail users that keeping email organized presents a major problem for some email users, resulting in backlogs of unread and unanswered mail.

In 2001, Duchenaut and Bellotti concluded further that email is being widely used as a personal information manager (PIM) [2]. Through interviews, they examined how people sort their email messages and deal with clutter in a business environment. The researchers suggest, “to better support the use of email as a PIM tool, organization of folders should be more flexible... the management of to-dos and reminders within email should be supported”. The interview results indicated that available software *did not adequately expose such features*. They raise the following question based on their research: “Would it be possible to leverage a model of users’ roles and organizational environment in the design of email clients? One possible way is to present a different interface, with different email management options, depending on a user’s role”.

In 2003, Duchenaut and Bellotti introduced a prototype of a task management-centric email client, and received mostly enthusiastic feedback from business users who tested it [6]. Two other recent papers [15, 16] discuss email organized by task or activity. The researchers’ choice of *task* management over *role* management appears to suit observed business usage patterns, where employees often juggle many short-lived tasks—all within the single role of their job [3]. However, in the case of college email use, our surveys suggest that students assume a large number of relatively permanent roles. Hence, a *role* management approach may be fitting.

The amount of available research on role management in software is presently limited.

In 1994, Plaisant and Shneiderman introduced role management in application to software user interface design [7]. Their proposal is rooted in earlier research from social psychology and organizational design [8, 9, 10]. The researchers quote Singh and Rein: “Role theory views individuals as occupying positions in organizations... Roles are the building blocks of organizational phenomena as division of labor and specialization” [11].

In 1995, Plaisant and Shneiderman created a software prototype of a computer desktop that uses role management [12]. Their prototype illustrates how role management can ease coordination between World Bank employees in their everyday duties, particularly those working on multiple projects. The presented design uses multiple views corresponding to the personal, workgroup, and institutional roles of an employee. When using a specific role, its relevant section of the view can be zoomed to fill the entire screen.

An example of role management design for college professors was prototyped in 1997 by Kandogan and Shneiderman [13, 14]. This research focused on the window management aspects of the interface and introduced a hierarchical windowing scheme that employs elastic windows to fill the screen area without overlapping. Top-level container windows correspond to roles, such as Research, Teaching, and Industry in the case of a professor; each can be maximized to fill the screen. These windows contain child windows representing specific subroles, such as different classes that a professor teaches. User testing with scenarios yielded a statistically significant performance advantage over a conventional windowing system. In another domain, Barreau and Nardi [16, 17] have argued for the importance of location-based saving and

searching, and have shown that the user's perception of their information space and the location of information within that space serve as a reminding function. This is in contrast with Fertig, Freeman, and Gelernter [19] who suggest that users only need better tools to find the value in archiving without specific organization.

The preceding research makes a case for exploring role management for email clients on campuses. This paper describes information gathered about the use of email by the student population, and presents a prototype interface illustrating how a “student role” might be implemented in a role-centric email program. This research was conducted by an interdisciplinary “Gemstone team”¹ of undergraduate students.

3. UNDERSTANDING STUDENTS NEEDS

In order to learn about the concerns, preferences, attitudes, and needs of students, two surveys were conducted on campus. By studying representatives of the college student population, information about email use was gathered and conclusions were drawn about the target population. The first survey was distributed in November 2001 to

¹ The Gemstone Program at the University of Maryland focuses on the development of the students outside the standard classroom environment, and challenges the students in the development of their research, teamwork, communication and leadership skills. Our team including students from Civil Engineering, Biochemistry, Electrical Engineering, Physiology and Neurobiology, German and Computer Science. Working under the guidance of a mentor we met once a week for 3 years. We conducted our research mostly independently, wrote a final thesis about our work, which is summarized here.

students at the University of Maryland, College Park.

The survey revealed that college students use email to communicate on a daily basis. Of 35 students surveyed, 86% check their email several times a day and 100% check their email at least once a day. In addition, 89% of students use more than one email address to send and receive email messages. The extent to which students use email as a communication tool is evident in the frequency with which students check their email and the number of different email addresses that students have.

To assess the adequacy of current email software in meeting the needs of college students, students were asked to identify the email functions that they use regularly. Some functions (e.g. send attachment, forward message and delete messages) are used by nearly all students, while other functions (e.g. send signature file and send autoreply message) are used by only a few students. Figure 1 identifies email functions and the frequency with which they are used by students. While some of the features are simply not relevant to the everyday student, other features are not used because of the complexity that surrounds the feature and the lack of exposure in the email program. For example, 100% of respondents receive junk email; however, only 43% use filters to block the unwanted messages. 6% of students were uncertain of what filters are, and 40% believe filtering should be improved, particularly its ease of use.

The topic of email organization was also addressed in the survey. Students were asked if they use folders to sort and store email messages. 80% of students surveyed use folders. Three quarters of these students have less than 10 folders. The rest of the students surveyed have between 10 and 30

folders. Email organization is relevant to college students as evidenced by 48% of students who save more than half of all the emails that they receive. Only 21% of students save less than one tenth of all the email that they receive.

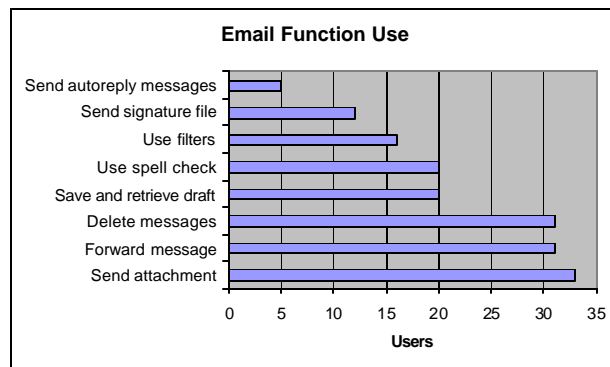


Figure 1. Use of various email functions as described by University of Maryland students in 2001 survey

To better understand student use of email, students were asked to identify the people that they email regularly. As expected, students use email to communicate with friends and family members. 63% of students also use email to communicate with coworkers (Figure 2).

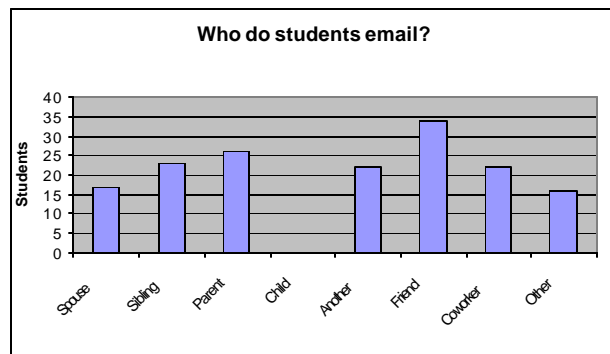


Figure 2 . Person(s) to whom University of Maryland students sent emails as found in 2001 survey

A section of the survey was devoted to the evaluation of current email software by students. Students commented on email features that they like and dislike. Students named the following positive features frequently:

- speed
- simplicity
- email notification
- address book
- folders
- support for multiple email addresses

Students also identified problems with current email programs. The following issues were acknowledged by

- students:
- difficulty changing how features work
- difficulty setting up
- suboptimal default settings
- lack of spell checking
- feature overload

Despite a small sample size, the first survey provides a good deal of insight into student email use. The survey revealed that current email software does not meet the needs of students fully and it can be changed to better accommodate students. Disregarding the technical shortcomings of specific email programs, the following issues appear to present problems for students: insufficient exposure and presentation of complex features and limited tools for email organization.

In April 2002, a second survey was distributed to 47 individuals, most of whom were students at the University of Maryland, College Park. Like the first survey, the second survey addressed problems that students encounter with current email software. In addition, the second survey considered students' attitudes towards potential user interface changes.

Students were asked to identify any email features that they avoid using and to explain why they avoid using them. Three students responded that they avoid sending and receiving email attachments. One student attributed his/her fear to viruses,

while another user confessed that he/she does not know how to send or receive an email attachment. Two users said they avoid using email features, which they don't understand. Another student wrote that he/she does not use the Find or the Search feature because it is not intuitive. While there was no particular feature that the majority of users avoid, the inadequate presentation and explanation of features in email programs is clearly a problem for students.

To further investigate filtering, students were asked if they understand and know how to use filters. Nine percent of students confessed that they did not know how to use filters and 32% had an idea, but were not exactly sure. This feedback supports the conclusion that filtering is not clearly presented in current email programs. A simpler, more intuitive filtering system would probably increase the use of filters.

In general, students were receptive to automation in email. Most students (72%) told us that they would like to have their emails automatically sorted for them. Students were also enthusiastic about an email program that changes to accommodate their personal preferences. Two thirds of students surveyed said that they were interested in an email program that adapts to their preferences. Most students, however, were not comfortable using an email program that keeps track of their usage patterns and makes inferences about their intentions (for the exclusive purpose of adjusting the user interface). The unwillingness of students to give up this control raises questions about the viability of adaptive interfaces in email design.

4. PROPOSED SOLUTION

Based on the feedback received, we set out to design a user interface that addresses the main problems for college students: email overload and feature overload. We believe that role management addresses these problems in two chief ways. First, organizing messages by role and subrole reduces *email* overload—students often assume a fair number of enduring subroles between which messages could be divided. Second, the ability to select a current role permits hiding functionality irrelevant to that role, alleviating *feature* overload. A lighter feature set also leaves more room for special functionality in a given role, such as a school calendar.

In designing a role management user interface, criteria for simplicity were established. The interface had to be sufficiently *familiar* to current email users. Ideally, novices could use the program exactly like *existing* email software while they explored the role management functionality. The additional *overhead* from using role management ought to be minimal to reduce the switching penalty. Finally the interface had to “degrade gracefully” to still be useful and usable when no information was available about the roles of the users or when the users were not willing to change their practice.

The design process involved many brainstorming sessions and a selective synthesis of ideas into a single interface. After several revisions of sketched paper mock-ups, screenshots were generated using computer graphics tools to better resemble an actual interface. Those screenshots were used to collect feedback from potential users. Finally we implemented a visual prototype to illustrate some of the

interactions. Figure 3-4 and 5 shows sample screenshots of the proposed interface.

The most significant departure from standard email clients is the presence of role selection tabs. Each role represents a separate environment where only messages, contacts, and functionality relevant to the role are visible. In the example of Figure 3 two specific roles are available - School and Work - with the School role currently selected. The General role corresponds to the “standard” entry to the email interface where no roles specified. Roles can include subroles. This is illustrated in our example where each class (e.g. ANTH 240, ENEE 435) is a subrole of the School role and the Work roles has a “circuit project” and “reports” roles.

Once a role is selected, the information panel located on the right side of the screen summarizes the most important organizational information of the role. For the school role it shows University Announcements and the list of classes (Figure 3). For the Work role, it shows general announcements and the two projects “Circuit project” and “Reports” (Figure 5). For contextual cues, a different visual theme distinguishes each role; this is limited to color in our prototype but can include fonts, icon style, sound effects etc., to clearly indicate which role is currently being assumed. The roles are not mutually exclusive. A given message or contact is visible in any role to which it is marked as related. Essentially a role acts as a gateway to the messages, contacts, and functions relevant to the role.

Each role allows several views, which are modes of operation that monopolize most of the screen area. The Mail view is the default view and provides an interface to the user’s mailbox (depicted in Figure 3 and 5 for school and work respectively).

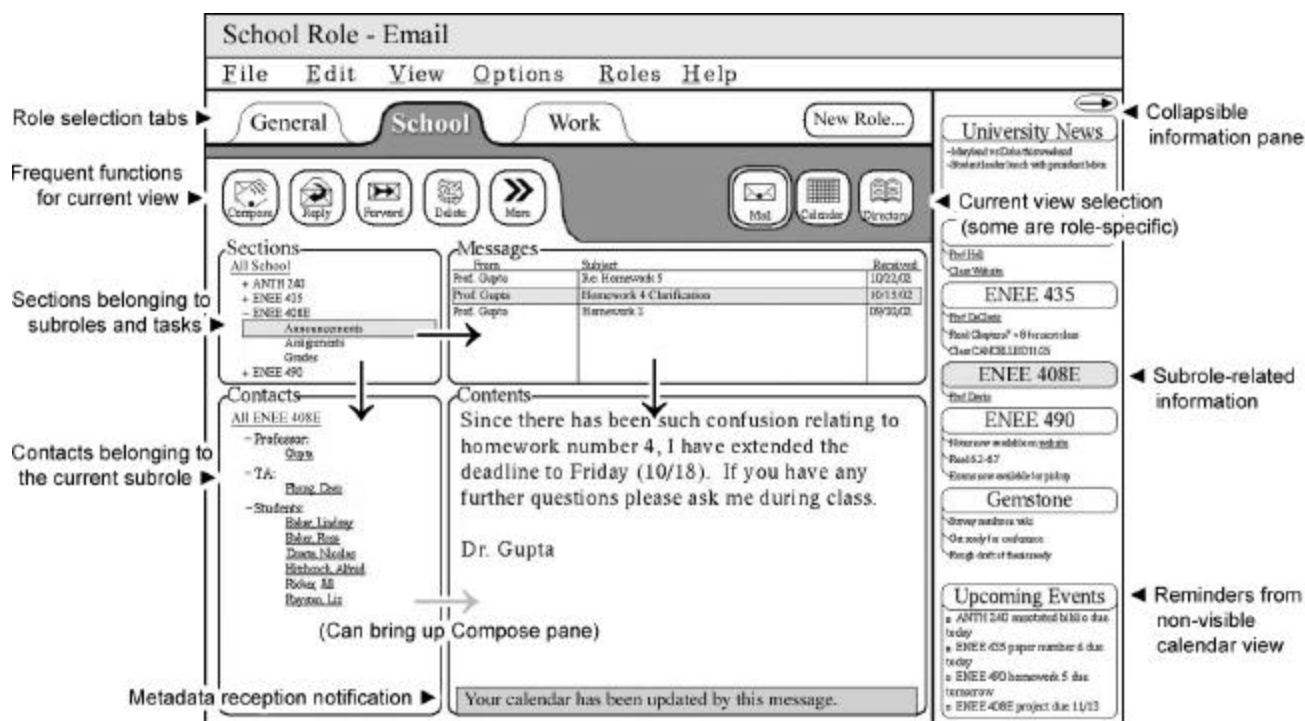


Figure 3. Role management email interface in School role under Mail view. Arrows added to the screen shot indicate the linkage between parts of the display. Here the student has clicked on ENEE 408E to show only the mail related to the ENEE 408E class. The contact list was filtered as well to show only students enrolled and instructors teaching that class. Further filtering of the mail list can be done, here only announcements are shown. One the ENEE 408E role is selected, a click on “calendar” will switch to the calendar for that particular class

An alternate view is the calendar view (Figure 4 and 6). Each view is accessed by clicking on a button on the right below the role tab².

The subroles belonging to a role appear in the Role Information Panel on the right of the screen and are also represented by a hierarchical treeview under “Sections”. This treeview allows users to create additional folders if needed in the role. Similarly to roles, subroles and folders behave non-exclusively—unlike the folders in common email programs. For example, when users switch from the Work to School role, they see all the messages and contacts relevant to

school, until they focus on a specific subrole/class. Similarly switching to the “ENEE 408E” subrole will display *all* messages related to that class, including messages from announcements, assignments and grades (Figure 3) and list all the contacts for that particular class. The treeview creates a filtering hierarchy that gives the user control over the breadth of information presented.

The views are vehicles for delivering role-specific functionality. The most general alternative view is the calendar. Figure 4 shows the calendar view of the School role. The school calendar displays data in a manner convenient for school-related tasks, such as presenting a semester

² Our prototype currently shows a “directory” view but this is no longer needed and was replaced by the contact box in the lower left corner of the mail view.

layout (opposed to of a quarter layout for the work calendar). The calendar can indicate with different colors class times, exams, assignments deadlines etc. Again, the School role view shows all the events of the school role, while focusing on a particular class (by clicking on the right side information panel) will focus on the schedule of that class.

The General role is special because it encompasses messages and contacts from all roles. Consequently, the General role acts like a regular email application and satisfies the criterion of providing a foothold for users transitioning from non role-based email. The other purpose of the General role is to hold correspondence that either does not *fit* any defined role, or that the user has not yet *linked* to a role. The Sections treeview under General can reflect this by providing a child node encompassing

General-only content, which descends from a treeview root node that contains everything.

The information panel, located on the right side of the screen, can be used to screen the most important information (such as clickable top contacts and reminders) from each subrole. It could also be clicked to *select* the current subrole, as an alternative to the Sections treeview. Finally, the bottom portion of the information panel can provide a summary of views that are *not* currently visible. In Figure 4, it reminds the user that new mail has arrived, which could be found in the Mail view (clicking the reminder would switch to it). The information panel is essentially the automated display of specific types of information delineated in email messages accessible to the user via the role/subrole hierarchy.

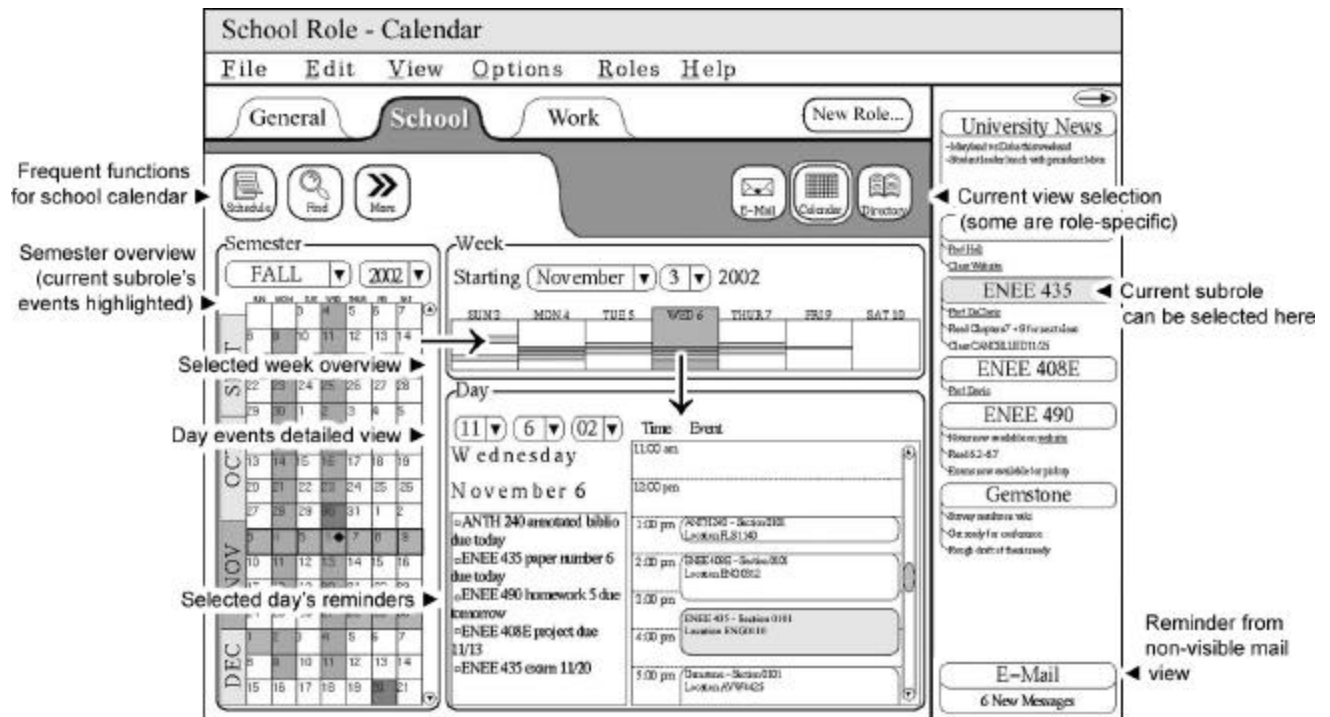


Figure 4. Role management email interface in School role under Calendar view. Here the ENEE 435 class (or subrole) has been selected. A semester calendar corresponding to the class duration is shown, color coded to represent class meeting time, assignments and exams, on top of a black and white view of the complete school role calendar. Day events are listed for all classes as well but ENEE 435 events are highlighted.

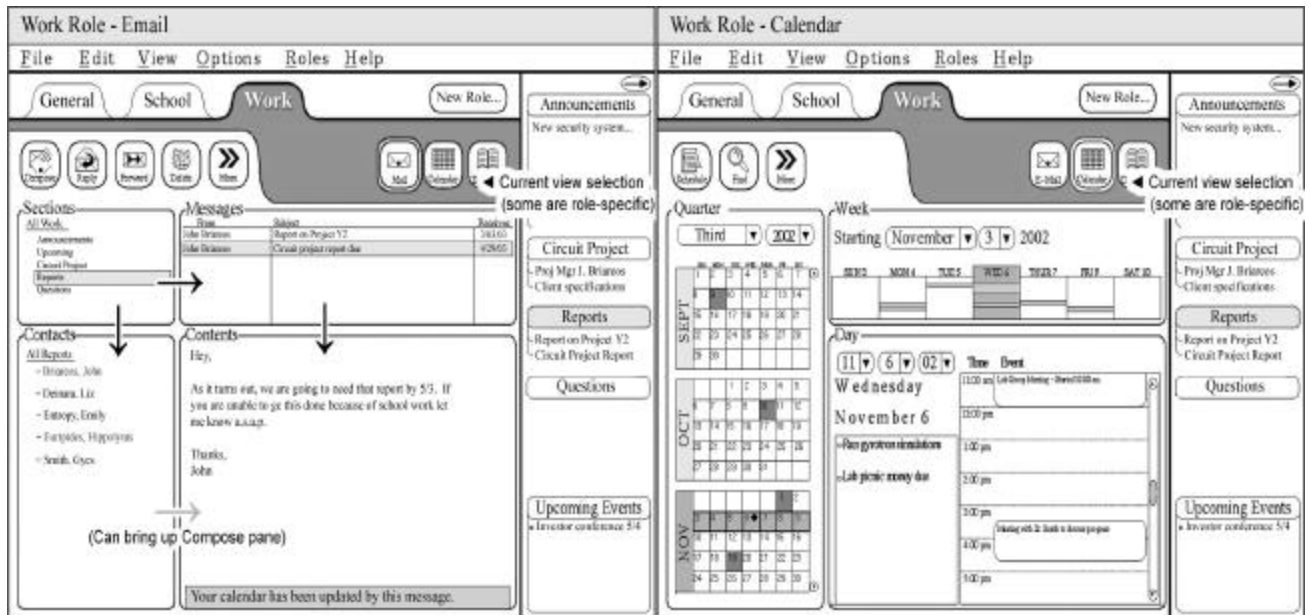


Figure 5 and 6. Role management email interfaces in the Work role for Mail (left) and Calendar views (right). The work role is not as well defined as the school role. Still users can define customized calendars (here a quarter calendar), or different options – e.g. a different signature file and automatic spell checking. The contacts are limited and different from the school role, and play an important part in characterizing the role itself.

5. FEASIBILITY CONSIDERATIONS

Consider the following scenario of use: Matt, a typical college student, was just accepted at the University of Maryland. He is requested to come to school with a computer and encouraged, possibly required, to download and get familiar with a recommended (role based) email program that has been tailored to university students. When he installs the software, school calendar is already populated with class registration deadlines, university holidays, and the last day of class. When Matt registers for classes he received automated acknowledgment email messages that includes metadata information about the class. This information is used by the email program to setup the school role for Matt. His calendar is updated (after he reviews the information and acknowledge the automatic loading in his calendar) and the contact list already includes information about the

instructor and the teaching assistant. When class starts a reminder email indicates a classroom change and loads the contact information of classmates.

When reading email Matt can now chose to read all his email at once (using the General tab), or focus on his School role first, then review the other messages. While he is reading his school email, he sees in the information panel on the right that the ENEE 430 professor has highlighted that the upcoming group project 1st deadline is approaching. In one click he can switch to that class subrole and review the class calendar, which is useful since - as many other undergrad students – he never manages his personal calendar. He switches to the email view, but can't quite remember the name of the fellow classmate he is supposed to work with so he scans the list of classmates. He remembers the name... and sends email to setup a meeting.

A few months later, Matt gets a part-time job in a local company. At first, all his work related email appears in the General role. After a few weeks, Matt has received emails from many people in the company and he spends 5 minutes setting up his Work role. He drags messages sent by work colleagues onto the Work role to add their names in the Work contact list. A few months later he already works on two projects so he creates two subroles for Work. Years later the company adopts a role-based email system and when Matt graduates and quits his job, he can pass his role to another fellow student by emailing him the role information (calendar, contacts, selected important emails, todo lists, reports) all at once. Soon he will also delete his entire school role all together...

Of course many technical aspects have to be worked out to make this scenario come true. One important question is how the program could determine what messages and contacts fit under what roles. In a general email interface this would be very difficult to achieve but in our context a lot of structure can be imposed automatically thanks to the formal organization of University activities in classes and terms, which leads us to believe that a school role can realistically be setup. Secondly, our experience as students as well as the results of our surveys suggests that many students' lives are clearly compartmented between school, work, old friends and family, and "others", therefore increasing the chance of a correct role identification based solely on the names of the people listed in the email. Students often use several email addresses to separate their emails - and roles. Those different addresses could also be used to filter email in a unified interface. The next approach is to assume nothing about a

unmarked message until the user adds it to the desired role (a one-click and drag could do this). All subsequent correspondence with the added person is included in that role until the user cancels the association. Finally, if the sender used a role based client, the role the message was sent from could provide information valuable to this process (i.e the message was most likely related to that role). Unassociated messages and contacts remain only in the General role. This method could be seen as a simple form of *filtering*.

Another issue is delivering the desired role-specific functionality. For example, automatically adding contacts and dates or finding important news may be beyond standard email capabilities. One way to facilitate such features is with back-end functionality at the institutional level. This may require a university to provide a customized email client and to run a server with class news or schedule updates. University staff could define new rules as needed (e.g. the president of a student-lead activity would receive a new role with a calendar of deadlines, a specific list of contacts and a budget view). Some of the centralization could be avoided by imbedding metadata in the email. Using ideas promoted by the Semantic Web research the email client could provide tools for embedding role associations, news items, schedule changes, or other meta-information into messages. Upon receiving such messages, the client could reliably read and act upon the information (with requisite security considerations). Metadata-unaware email clients would simply ignore the data.

Role based functionality is more likely to be useful for roles such as School, where the university or professors provide supports and benefits can be gained without any extra

work from users. On the other hand people under Friends and Family roles are not likely to provide servers or to embed metadata... In these cases, the interface needs to “degrade gracefully”. Merely identifying the people involved in the role might be useful. The software can provide templates for commonly used roles like Friends and Family, including tools to help the user with this process. Ultimately, when no assumptions can be made about a role, the system continues to provide the benefits of splitting content between roles and subroles—a simple form of filtering controlled by the user. In the worse case users read their email in the General role and nothing is lost.

For advanced users customization of the roles will increase the benefits of using roles. For example roles could use a different signature (formal for work, informal for school, home address for friends and family.) Automatic spell checking might be enabled in the Work role but not the School role. In addition to the increased likelihood of a remaining engaged in a role-based activity, the time saved by having those simple differentiations between roles can very well compensate for the extra time spent switching role.

6. DESIGN TESTING AND FEEDBACK

As a preliminary assessment of this role management interface, we conducted scenario testing and interview sessions. Twenty students from the University of Maryland were interviewed during November-December of 2002. The testing procedure involved printed prototype mock-ups, and was designed to measure the subject’s understanding of the interface. Before testing, initial impressions were asked and recorded. Several scenarios

calling for simple tasks were then presented. No prior training or demonstration was provided. The subjects were encouraged to verbalize their thought process, and their remarks were recorded. A follow-up interview was conducted afterwards.

From the initial impressions, many of subjects considered the interface “busy”. These subjects were asked what information they would eliminate, and how well the information was organized. Several subjects thought that the information panel was not always useful, and thought it should be collapsible. The calendar’s weekly and daily views seems too detailed, since many students seldom used calendars to record personal information. Feedback on the organization of information was generally positive, and the hierarchical views in the Calendar received praise. One student commented that it was easy to focus on short-term activities without losing sight of long-term goals. The majority of subjects recognized the purpose of role folders right away; a few initially mistook the Work role tab for campus job searches (which in fact could be the default setup for students who do not have a job yet!)

In the scenarios, the subjects had little trouble recognizing the involved interface features, including the view selection buttons, the information panel, and the contacts treeview. Asked to look up the dates of next semester’s spring break, 75% correctly selected the Calendar view and manipulated the pull-down semester menu. Note that the testing was done on paper prototypes and without training so we could only observe if their first intuition about where to find the information was correct (opposed to seeing how they would explore the interface until they found what they needed). Starting from the Calendar view, the subjects were asked to email their class

instructor. 60% took the shortest path by using the instructor email link in the information panel, while the rest preferred to switch to the Mail view and use the contact list. When asked to check for snow-related class cancellations, 85% correctly selected the University News link in the information panel of the School role. To send a mass email to everyone in a class, 70% made the optimal choice by clicking the class's root node in the Contacts treeview inside the Mail view.

The follow-up interviews examined the interface's perceived viability as a *personal information manager* (PIM). This issue is relevant to the target audience, since 65% of the subjects admitted to using a date book or another kind of scheduler. 70% said they would consider using a program like the one presented in place of their current planner (if they had one). 65% said they would use the program to check their daily agenda (remember that many undergraduate students don't even own a calendar and rely on instructors' constant reminders). While such statements may not predict actual usage, they suggest a generally positive reaction. The remaining questions involved automation, and the students remained opposed to unattended changes: 75% wanted to be notified of changes to their schedule and to be asked their approval.

The non-functional nature of our prototype precluded testing the interface's capacity for managing email overload. However, the preliminary testing results suggest that major *features* of the interface are presented well enough to be readily used—including special school-related features. This indicates that feature overload may be reduced when functionality is customized by role.

7. CONCLUSIONS

Our research findings on campus email use have several implications for those designing future email clients: Students use email differently from the average business user and would benefit from specially designed interfaces. They communicate with a variety of well-defined - usually not overlapping - groups of people, and traditionally rely on multiple email addresses to separate their roles. Although school-related email use is heavy, functionality beyond simple messaging is sparsely used.

Current email clients do not adequately meet students' needs. Many students use planners and report a willingness to use email for personal information management (PIM) tasks; however, the majority fail to use the functionality provided by current software for such tasks. Feature overload is the apparent culprit. Customizing the interface to their needs and providing PIM features from the start by facilitating the download of university or class schedules is likely to increase use and streamline email and calendar management.

The same problem applies to tools for managing email overload. Most students keep a significant portion of their incoming mail. They would like to have the messages automatically sorted into folders, but don't seem to know how—or worry that messages will be misplaced. Role management may contribute to lessening the above problems. It addresses both email and feature overload by acting as a user-controlled filter and selecting only relevant messages, contacts, and functions to be presented.

Role Management is not a silver bullet. It obviously does not entirely solve the problem of email overall and software complexity but it allows users to focus on

particular roles when they need to, speeding the browsing of email and contact lists which have become significantly smaller, review calendars that can be focused on given roles, and potentially giving access to a to-do list or directories of documents that can be filtered by roles as well, all in an integrated environment.

Linking messages or contacts to a particular role remains a challenge for a general role management interface but we strongly believe that the University environment can provide the structure and support for role management interfaces that would help organize a large body of users with limited (at first) organizational skills.

The proposed interface illustrates one way that role management might be implemented; initial reactions from students are enthusiastic. We hope others will continue developing the idea of role management for University students or for other similarly structured environments. Developing a fully functional prototype is a challenge but should be the next step in evaluating the practicality of this approach

Acknowledgements

We want to thank Ben Bederson, Bill Billam, Evan Golub, Ross Malaga and Kent Norman for their feedback as thesis discussants. We also thank the Gemstone staff for their support during the project.

REFERENCES

1. Gallup Organization, (2001). Almost All E-Mail Users Say Internet, E-Mail Have Made Lives Better. Princeton, NJ.
2. Ducheneaut, N., & Bellotti, V. (2001). E-mail as Habitat: An Exploration of Embedded Personal Information Management. 30-38. New York: ACM.

3. Economics and Statistics Administration, & National Telecommunications and Information Administration. (2002). A Nation Online: How Americans Are Expanding Their Use of the Internet. Washington, DC. Retrieved November 18, 2002 from <http://www.esa.doc.gov/508/esa/ANationOnlineEXSFeb02.htm>
4. Jones, S. (2002). The Internet Goes to College: How Students are Living in the Future with Today's Technology. Washington DC: Pew Internet and American Life Project. <http://usinfo.state.gov/usa/t091602.htm>
5. Sidner, C., & Whittaker, S. (1996). Email Overload: Exploring Personal Information Management of Email. Proceedings of the SIGCHI conference on Human factor in computing systems. 276-283. New York: ACM.
6. Ducheneaut, N., Bellotti, V., Howard, M., & Smith, I. (2003). Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool, Proceedings of the conference on Human factor in computing systems. 345-352. New York: ACM.
7. Shneiderman, B., & Plaisant, C. (1994). The Future of Graphic User Interfaces: Personal Role Managers, People and Computers IX. 3-8. Glasgow, Scotland: British Computer Society. Early version also at <http://www.cs.umd.edu/local-cgi-bin/hcil/sr.pl?number=HCIL-94-05>
8. Biddle, B.J., & Thomas, E.J. (1979). Role Theory: Concepts and Research. New York: Krieger Publishing Company.

9. Roos, L.L., & Starke, F.A. (1981), "Organizational Roles", in Handbook of Organizational Design Vol. 1: Adapting Organizations to Their Environment, P.C. Nystrom & W.H.Starback [eds.], Oxford University Press.
10. Sarbin, T.R., & Allen, V.L. (1968), "Role Theory", in Handbook of Social Psychology (2nd Edition), G. Lindzey & E. Aronson [eds.], Addison Wesley.
11. Singh, B., & Rein, G. (1992). Role Interaction Nets (RINS): A Process Description Formalism. (Technical Report CT-083-92). Austin, TX: MCC.
12. Plaisant, C., Shneiderman, B. (1995). Organization overviews and role management: Inspiration for future desktop environments, Proc. IEEE 4th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. New York: ACM. Also at <http://www.cs.umd.edu/local-cgi-bin/hcil/sr.pl?number=HCIL-95-11>
13. Kandogan, E., & Shneiderman, B. (1996). Elastic Windows: improved Spatial Layout and Rapid Multiple Window Operations, Proc. Advanced Visual Interfaces, 29-38. New York: ACM.
14. Kandogan, E., & Shneiderman, B. (1997). Elastic Windows: evaluation of multi-window operations, Proceedings of the SIGCHI conference on Human factors in computing systems. 22-27. New York: ACM.
15. Kaptelinin, V., UMEA: Translating Interaction Histories into Project Contexts, Proceedings of the CHI 2003 conference on Human factor in computing systems. New York: ACM.
16. Venolia, G. D., Ceustaedter, C., Understanding Sequence and Reply Relationships within Email Conversations: A Mixed-Model Visualization, Proceedings of the CHI 2003 conference on Human factor in computing systems. New York: ACM.
17. Barreau, D.K. (1995). Context as a factor in personal information management systems. Journal of the American Society for Information Science, 46 (5), 327-339.
- 18) *Nardi B., Barreau , D., Finding and Reminding" Revisited: Appropriate Metaphors for File Organization at the Desktop, SIGCHI Bulletin Vol.29 No.1, January 1997*
19. Fertig, S., Freeman, E. and Gelernter, D. (1996). ["Finding and reminding" reconsidered.](#) ACM SIGCHI Bulletin, 28 (1), 66-69.
20. Freeman, E., Fertig., S. Lifestreams: Organizing your electronic life. In AAAI Fall Symposium: AI Applications in Knowledge Navigation and Retrieval, November 1995, Cambridge, MA.

Appendix 1: The gemstone team:

H. Ross Baker is a computer scientist with an interest in linguistics; he is now a graduate student in the Linguistics Department of Northwestern University.

Nicolas Duarte is an Electrical Engineer with an interest in molecular electronics and bioengineering; he is now a graduate student seeking his Ph.D. in the Electrical Engineering department of Pennsylvania State University.

Aydin Haririnia is a biochemist with an interest in protein NMR and crystallography, he is now a graduate student in the Department of Chemistry and Biochemistry at the University of Maryland.

Dawn Klinessmith is a civil engineer with an interest in structural engineering; she is a student in the civil and environmental engineering department at the University of Maryland, College Park.

Hannah Lee is a recent graduate of the University of Maryland with an interest in Physiology and Neurobiology.

Leonid Velikovich is a computer scientist with an interest in computer graphics; he is an undergraduate student in the Computer Science Department of the University of Maryland

Alfred Wanga is an electrical engineer with an interest in electronic materials and devices. He is now a graduate student at Penn State University.

Catherine Plaisant was the mentor of the team