

Success of an Agent-Assisted System that Reduces Email Overload

Andrew Faulring, Brad Myers, and Aaron Steinfeld
School of Computer Science, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213 USA
{faulring, bam, astein}@cs.cmu.edu

ABSTRACT

RADAR is a large multi-agent system with a mixed-initiative user interface designed to help office workers cope with email overload. RADAR agents observe experts performing tasks and then assist other users who are performing similar tasks. The Email Classifier learns to identify tasks contained within emails and then inspects new emails for similar tasks, which are presented in a novel task-management user interface. The Multi-task Coordination Assistant learns a model of the order in which experts perform tasks and then presents subsequent users with a suggested schedule for performing their tasks. A large evaluation of RADAR demonstrated that novice users confronted with an email overload test performed significantly better (a 37% better overall score with a factor of four fewer errors) when assisted by both agents. Additionally, in a post-test survey users perceived the test to be significantly more difficult when they did not receive assistance from both agents, indicating a preference for the AI-based assistance. We also observed a wide variation among users in the amount of agent advice that they followed.

Author Keywords

Task management, intelligent user interfaces, agents, email classification, intelligent planning, learning, RADAR.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces---Interaction styles, Graphical user interfaces (GUI); I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence---Intelligent agents, Multiagent systems.

INTRODUCTION

Email plays a central role in the work of many people. Unfortunately email client software is poorly suited to support the “collaborative quality of e-mail task and project management,” which results in people suffering from “email overload” [1]. Sometimes people read an email and then

immediately perform any tasks requested in the email. Performing similar tasks together reduces the overhead of switching between different task types. The efficiency of this strategy is significantly affected by the order in which emails are processed. In general it is difficult for people to quickly group similar tasks. Another way to group similar tasks is to inspect each email, manually create task metadata, and then generate an order for handling the tasks.

Several research projects have explored adding task-management features to email clients [1, 5, 10], and some email clients do provide features that facilitate task management such as tagging and separate to-do lists. Other research projects have explored how to make it easy to move tasks from email client software into dedicated task managers [9, 10, 12]. These studies focus on email because users often use email client software as a task manager since many tasks arrive as email messages and the information necessary to complete a task is commonly contained in the email as well. Hence, the inbox becomes an informal to-do list [2], even though email client software are not designed to perform the task-management duties that users demand from them [11]. Additionally, users resist doing that additional work to manually create tasks [11], and the manual grouping strategy forces them to read each email at least twice: once when creating the tasks, then again to actually do the task.

We have developed a mixed-initiative email system, which uses Artificial Intelligence (AI) learning techniques, to help reduce email overload. The mixed-initiative email system is a central feature of the **R**eflective **A**gents with **D**istributed **A**daptive **R**easoning (RADAR) project. RADAR is a large interdisciplinary project to build a suite of intelligent agents that help office workers complete routine tasks more efficiently [4]. The system works as follows. First, an Email Classifier observes the types of tasks an expert creates from emails and uses these training observations to learn a model, which it then uses to automatically create tasks for new emails. The found tasks are presented in a novel task-management user interface, which integrates a to-do list with support for performing the tasks. Second, the Multi-task Coordination Assistant (MCA) observes expert users performing tasks found by the Email Classifier and then learns models for efficiently performing a collection of such tasks. The MCA uses these learned models to assist users

*SUBMITTED TO IUI Workshop on
Users' Preferences Regarding
Intelligent User Interfaces:
Differences Among Users
and Changes Over Time
DO NOT CITE OR REDISTRIBUTE*

Incomplete Actions (11)						
Order	Description	Subject	Sender	Created	Modified	Creator
1	Modify Event: Demo M1: Driver Monitoring Systems	Attendance figures and new #	Amy Lim <lim12@ardra.org>	Today, 3:32 PM		RADAR
2	Modify Event	note schedule changes	Spence Pierre <spierro@ardra.org>	Today, 3:54 PM		RADAR
3	Modify Room: Flagstaff: Sternwheeler	Sternwheeler Capacity	Meredith Lorenz <lorenze@pittsburgh.flagstaff.com>	Today, 4:07 PM		RADAR
4	Modify Room: Flagstaff: Vandergrift	Sternwheeler Capacity	Meredith Lorenz <lorenze@pittsburgh.flagstaff.com>	Today, 4:10 PM		USER
5	Optimize the Schedule	no email		Today, 3:45 PM		RADAR
6	Website Update (VIQ): Modify Person: Austin Parton	Webpage	Austin Parton <aparton@ardra.org>	Today, 3:37 PM		RADAR
7	Website Update (VIQ): Modify Person	Attendance figures and new #	Amy Lim <lim12@ardra.org>	Today, 3:32 PM		RADAR
8	Website Update (VIQ)	Organization Wrong	Sonal Malhotra <smalh@ardra.org>	Today, 4:32 PM		RADAR
9	Website Update (WbE)	change phone numbers	Emily Halwizer <halwizer@ardra.org>	Today, 4:47 PM		RADAR
10	Place a Vendor Order	Tech. Request - flip charts	Maggie Foxenreiter <mfox@ardra.org>	Today, 3:33 PM		RADAR
11	Send a Briefing	Brief me, please	Jonathon Robertson <jrobertson@ardra.org>	Today, 4:42 PM		RADAR

Overflow Actions (1)						
Order	Description	Subject	Sender	Created	Modified	Creator
	Reply to Question	Vegetarian options?	Sandra Nubanks <snubanks@ardra.org>	Today, 4:02 PM		RADAR

Completed Actions (1)						
Order	Description	Subject	Sender	Created	Modified	Creator
	Modify Event: Workshop 1a: Intermodal Passenger Screening	Attendance figures	Amy Lim <lim12@ardra.org>	Today, 3:21 PM	Today, 3:45 PM	RADAR

Deleted Actions (1)						
Order	Description	Subject	Sender	Created	Modified	Creator
	Modify Speaker's Availability	Planning for History Week	Michelle Randal <mich-randal@gmail.com>	Today, 4:28 PM	Today, 4:34 PM	RADAR

Possibly Conference-Related Emails (1)						
Read	Subject	Sender	Date			
•	for my presentation	Laura Tindale <laurat2@ardra.org>	Today, 3:24 PM	Add an Action		
Blake, I didnt know who to contact about making sure to have a laptop available, and connected to teh AV equipment - ie projector. I want all that ready on the ...						

Other Emails (1)						
Read	Subject	Sender	Date			
•	car arrangements	Angie Randal <angiednacer6@gmail.com>	Today, 3:23 PM	Add an Action		
Ms K is counting on me to help out with the kids' dance class. The car is still in the shp. Can you drop me off over there? thanks :-)						

Deleted Emails (1)						
Read	Subject	Sender	Date			
	Precipitation Update	Weather Alerts <weather@weather.gov>	Today, 3:56 PM	Add an Action		
There is a 70% probability for thunderstorms with heavy rain in ALLEGHENY COUNTY this evening through tomorrow. Plan accordingly and be safe! Go to www.weather.gov ...						

Figure 1: The RADAR Action List provides a task-centric view of an email inbox. The “Incomplete Actions” (a), “Overflow Actions” (b), and “Completed Actions” (c) tables list the tasks contained within email messages, allowing the user to sort by task-centric properties. The bottom three Email tables (e, f, and g) contain emails for which no tasks have been created.

working on a similar set of tasks. The MCA proposes a schedule for performing tasks, emphasizes highly-important “critical” tasks, recommends tasks to skip if time is limited, and issues warnings if the user’s behavior deviates significantly from expert behavior. A novel progress bar shows the suggested schedule of future tasks as well as what has been accomplished to improve users’ situational awareness. The RADAR approach reduces the number of times a person has to read an email, while at the same time allowing the tasks within the email to be efficiently performed.

A large-scale user test has evaluated several versions of the RADAR system over the past three years [7]. The test measures RADAR’s performance using quantitative metrics acquired through data logging including an overall evaluation score that summarizes overall performance into a single objective number [7], along with qualitative metrics collected with a post-test user survey [8]. The most recent test in August 2008 showed that the AI assistance was helpful: users who received AI assistance performed 37% better compared with users who did not. Additionally, users were able to recognize when RADAR makes errors, correctly handling 89% of the tasks that the classifier erroneously suggested (false positives). Overall, these users incorrectly completed 2.6 tasks per user. In contrast, the users without AI assistance incorrectly completed 10.3 tasks on average, which accounted for 19% of the tasks they did. These users, who only had the direct manipulation interface, committed a factor of four *more* errors. After completing the test, users

filled out a questionnaire that assessed their perceptions of the system [8]. Users who did not received assistance from RADAR perceived the test to be significantly more difficult (4.87 vs. 3.75 on a 1–7 Likert scale where 7 meant most difficult). This preference for the RADAR version may be understated because the between-subject design prevents participants from directing comparing the different conditions. On a separate question, users were asked if the way in which the software assisted them was stressful. They did not report a difference in stress between conditions, indicating RADAR’s suggestions were presented in a helpful manner. Additionally, we observed much variation in the amount of RADAR advice that users heeded; some users completely ignored some advice, whereas other users considered all advice.

TASK-CENTRIC EMAIL PROCESSING

In creating the Action¹ List, we worked to transition the processing of email requests from the traditional email-centric workflow to a task-centric workflow. The Action List design provides a task-centric view of the user’s email

¹ One important discovery from the usability testing is that the term “action” works better than “task,” so we used the former term in the user interface. However, we will continue to use the term “task” throughout this paper, except when referring specifically to a user interface element.

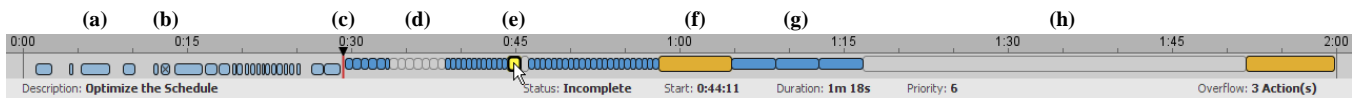


Figure 2: The Progress Bar shows completed (a) and deleted (b) tasks to the left of the current time (c), and the suggested schedule to the right. Non-critical tasks are blue (a, b, and g), critical tasks are orange (f), and expected tasks are gray (d and h). Details about the highlighted task (e) are shown in the status bar at the bottom.

inbox and supports a mixed-initiative interaction style for creating and completing tasks contained within emails. The Email Classifier automatically creates tasks from emails, which are displayed in the Action List. The Action List allows a user to inspect the tasks that the Email Classifier created, add ones that were missed (false negatives), delete ones that should not have been created (false positives), and launch web pages to perform some of the tasks.

The Action List contains seven tables divided into two groups: the first for tasks, and the second for emails (see Figure 1). The task group contains four tables for “Incomplete” (a), “Overflow” (b), “Completed” (c), and “Deleted” (d) tasks. Tasks that the user has yet to perform are split between the Incomplete and Overflow table, with the latter table containing tasks that the MCA recommends that the user should skip due to time constraints. Tasks completed by the user appear in the Completed table, which provide the user with a record of their progress. The Deleted table is intended for tasks that RADAR created erroneously and the user subsequently deleted.

The three green tables (e, f, and g) display emails that are not associated with any tasks. The first table (e) contains emails that RADAR thinks may contain tasks but for which it could not confidently identify the exact task type. This partial classification focuses the user’s attention on emails likely to contain tasks without risking problems that might result if RADAR incorrectly classified the task as being of a particular type. The second table contains other emails that RADAR did not identify as task-related (f). The third table contains emails that the user deleted (g).

All of the tables provide features that are similar to a traditional email inbox, such as columns for the subject, sender, and date. The entry for each email also includes an excerpt from the beginning of the email body to aid the user in determining whether an email contains a task without requiring the user open the email. Clicking on either the subject text or the “Add an Action” link displays the standard header and body sections along with the list of tasks that the user can add to the email.

PROVIDING TASK ORDERING ADVICE

RADAR’s Multi-task Coordination Assistant (MCA) provides guidance about the order in which to work on a set of tasks. MCA is designed to support near-term deadlines on the order of 1–8 hours and situations in which the amount of work exceeds the time allotted. RADAR learns task models upon which the advice is based through passive observation of experts performing similar tasks. The MCA includes a novel visualization in the form of a progress bar

(see Figure 2), which shows a user the completed tasks and the suggested schedule for incomplete tasks. MCA’s goal is to provide advice that improves performance and reduces overall performance variance.

User Interfaces for Suggesting a Schedule

The primary advice provided by the MCA is the suggested schedule, which specifies an order in which to perform outstanding tasks. The MCA suggests which tasks to skip when it calculates that there is not enough time remaining to perform all incomplete tasks. The MCA identifies “critical” tasks, which are particularly important for the user to complete. The MCA learns what tasks are generated after other tasks are completed, and so it also adds such “expected” tasks to the schedule. This advice provides the user with a more realistic understanding of upcoming work and eliminates major changes to the schedule that would otherwise occur when an expected task became real.

The Progress Bar (see Figure 2) appears at the bottom of the screen just above the Windows Taskbar and always remains on top without obscuring other windows. Time is represented on the horizontal axis, which in this case spans two hours. An inverted black triangle and a vertical red line represent the current time (c), which moves from left to right. Each box represents a task. Tasks to the left of the current time represent completed (a) or deleted (b) tasks, providing a record of the user’s progress so far. The width of a task box represents the time that the user spent working on the task. The suggested schedule is visualized by the tasks to the right of the current time. The width of those task boxes represents the amount of time that the MCA expects the task to require. Blue task boxes represent non-critical tasks (a, b, and g). Orange task boxes, which are also slightly taller, represent critical tasks (f). Gray boxes represent expected tasks (d); expected critical tasks appear as taller gray boxes (h). The user can quickly inspect any task by moving the mouse over its task box (e), which updates the status bar at the bottom with the highlighted task’s description, status, actual/planned start time, actual/planned duration, and priority (actual for completed and deleted tasks, planned for incomplete tasks). Double-clicking on a box opens the corresponding task. The highlighted task, along with all other tasks of the same type, is drawn with a thicker border to allow the user to see where that type of task is distributed throughout the schedule. The number of overflow tasks, which are the ones the MCA proposes to skip due to time constraints, appears at the bottom right.

MCA advice also appears in other parts of the RADAR user interface. First, the Action List’s “Order” column shows the

position of each future task within the suggested schedule (see Figure 1(a)). Only tasks in the “Incomplete Actions” table are included in the suggested schedule; the “Order” column is blank in the other tables. Sorting the “Incomplete Actions” table by the “Order” column shows the schedule as an ordered to-do list. Second, tasks that MCA suggests that the user skip are shown in the “Overflow” table. Third, after the user completes or deletes a task, RADAR redisplay the task’s form in a finished state to provide feedback that the command succeeded. This confirmation screen also includes a link to the next suggested task in the schedule. This “Next Suggested Task” link allows the user to navigate to the next recommended task without having to return to the Action List to find it. Finally, the MCA displays pop-up warning dialogs when the user significantly deviates from the suggested schedule. In particular, the warnings are issued if the user works on a critical task much earlier than experts did (Early Critical), if the user has not yet started working on a critical task by the time most experts had (Late Critical), or if the user starts working on a critical task that is not the next critical task on the suggested schedule (Wrong Critical).

Training the MCA

The primary goal of the MCA training process was to provide MCA with an opportunity to infer the high-level strategy through passive observations of experts using that strategy to perform the two-hour study. The MCA learns models of expert behavior by observing experts performing tasks using the same user interfaces that novice participants will later use. Experts did the two-hour study using a version of the system for which the MCA learning components were watching rather than recommending. Other AI components operated normally. For example, the Email Classifier had already analyzed the emails and identified tasks. The training used three different sets of emails (none of which was the test email set), which provides variability to prevent overtraining. In a real deployment, the MCA would be changed to use on-line learning, so it would continuously adapt to users’ choices.

EVALUATION

We evaluated RADAR using a conference planning test to determine how effective it is at assisting novice users based upon learned models of expert performance. The test compared participant performance among three conditions: *None*, which has no learned models; *TC* (task-centric), which has all the learned models except for the MCA ones; and *TCO* (task-centric-with-ordering) condition with all AI systems active. The three conditions allow us to measure the impact of two parts of the RADAR system described in this paper. After describing the study design, we report data logged during the test that offer insights into how well the designs worked.

The Conference Planning Test

Project members, in cooperation with external evaluators, developed a system-wide user test to evaluate how well

RADAR’s user interface and AI technologies assist a novice user. This section provides an overview of the conference planning test; a full description of the test can be found elsewhere [7]. Over the past three years, five separate studies have used this test to evaluate different RADAR versions. The latest study presented here evaluated the current RADAR 3.0 system.

The test presents participants with a simulated conference-planning scenario. Participants assume the role of the conference planner, filling in for the regular planner, Blake, who is indisposed. The simulated four-day, multi-track conference has keynotes, plenary talks, banquets, paper sessions, poster sessions, workshops, and so forth. Participants in our study must handle the outstanding conference planning tasks which have arrived in email, including many requests from the conference attendees. Blake’s inbox contains these emails, which can be categorized as follows:

- **Scheduling:** Participants must update the database of event *constraints* (A/V requirements, meal preferences, attendee availabilities², and so forth) and conference room *properties* using an appropriate web form. The Schedule Optimizer [3] uses information in this database to generate the conference schedule.
- **Website:** Attendees request corrections to their contact information on the conference website. The study participants must also update the website’s conference schedule based upon the output of the Schedule Optimizer using a variety of forms.
- **Informational:** Attendees request information about the conference, generally concerning how the schedule has changed. The participants must author a reply email.
- **Vendors:** Attendees specify meal preferences and A/V requirements for events which then have to be forwarded to vendors using the vendor’s web forms.
- **Briefing:** The conference chair requests a briefing that summarizes the participant’s progress at the end of the test. Blake’s inbox only contains one such email. The participant must invoke a special Briefing Assistant tool [6] to handle this request.

Participants also must deal with a conference crisis, which involves the loss of use of a significant number of rooms in which conference events had already been scheduled. Participants now need to find new rooms for the conference and adjust the schedule such that each event is placed in a room that satisfies the event’s constraints, such as capacity, available equipment, seating arrangement, and so forth.

In order to increase the validity of the tests, each study uses a unique *email set*. The email sets have comparable difficulty and task distribution. The sets differ in the exact nature of the crisis—the specific rooms and times lost—and the

² Speakers are not necessarily expected to attend the entire conference. Hence, the schedule needs to accommodate the availability of each speaker.

details of the other email requests. Additionally the email set, which simulates Blake’s real email, contains other “distracter” emails, including personal emails, which are unrelated to the conference. The AI components are not allowed to train on any of the actual email sets. The sets are created by an outside consultant and kept secret until the test.

This test is designed to be hard for the participants—and it is. Of the hundreds of people who have participated in pilot or test sessions, including RADAR researchers, no one has completed all of the tasks within the allotted two hours. We therefore think this approximates what a real person experiences, where it is often impossible to handle in one sitting all the emails that are pending.

Email Set

The email set for this year’s study had 123 emails, 83 of which contained a total of 153 tasks. The number of tasks is greater than the 102 task labels mentioned earlier, since some emails required multiple tasks of the same type. However, the classifier only reports if an email contains at least one task per type; it cannot determine how many tasks of that type are actually required. The other 40 “distracter” emails were unrelated to the conference.

Method

Conditions

The test used a between-subjects design with a single independent variable, *Assistance*, which has three levels: *None*, *TC* (task-centric), *TCO* (task-centric-with-ordering).

In the *None* condition, most of RADAR’s intelligent components were disabled. Specifically the Action List initially had no email-based tasks since the Email Classifier was disabled, all the MCA advice was disabled, and the Progress Bar only showed the task history. The main differences from the Action List in Figure 1 were that the “Order” column, the “Overflow Actions” table, and “Possibly Conference-Related Emails” table were not displayed, and the action tables were initially empty.

In the *TC* condition, all of RADAR’s AI components were enabled except for the MCA. The Action List contained the tasks that the Email Classifier found along with the “Possibly Conference-Related Actions” table. Again, the Progress Bar only showed the task history. The main differences from the Action List in Figure 1 were that the “Order” column and the “Overflow Actions” table were not displayed.

In the *TCO* condition, all MCA functionality was enabled, as described in the previous sections.

Sessions

Each test session could include up to 15 participants and lasted up to 4.25 hours. In the first phase, participants learned about the conference-planning test and participated in hands-on training with the software. Following a break, participants started the two-hour testing session, which included another break after one hour. Then participants

completed a survey and receive payment, including extra payments if they achieved specified milestones.

Participants

Participants were recruited from local universities and the general population using a human participant recruitment website. Participants were required to be between the ages of 18 and 65, be fluent in English, and not be affiliated with or working on the RADAR project. The study include 23 participants in the *None* condition, 28 participants in the *TC* condition, and 28 participants in the *TCO* condition. The number of participants varied among conditions since not all session yielded 15 usable data sets due to no-shows, participants who dropped out, participants who failed to make a good-faith effort, and software crashes or configuration issues that invalidated the data.

Results

Evaluation Score

An evaluation score summarized overall performance into a single objective score ranging from 0.000 to 1.000 [4, 7]. It was important that this score be tied to objective conference planning performance rather than a technology-specific algorithm (e.g., F1 for email classification). This technology-agnostic approach allows us to compare performance across conditions given any technology. The evaluation score was designed and developed by external program evaluators. The score function summarized overall performance by giving points for satisfying certain conditions, coupled with costs and penalties. These conditions included quality of conference schedule (e.g., constraints met, special requests handled), adequate briefing to the conference chair, accurate adjustment of the web site (e.g., contact information changes, updating the schedule on the website), costs for the rooms, food, and equipment for the conference, and penalties for requesting that others give up their existing room reservations. The score coefficients were 2/3rd for the schedule, 1/6th for website updating, and 1/6th for the briefing quality.

On this measure, *TCO* participants clearly outperformed *TC* participants, who in turn outperformed *None* participants (ANOVA, $F(2,76)=83.7, p<0.0001$):

Assistance	N	Mean	Std Dev
None	23	0.550	0.072
TC	28	0.706	0.063
TCO	28	0.754	0.035

A subsequent Tukey post-hoc test found that the three conditions were significantly different from each other. All but 3 of the 28 *TCO* participants earned higher scores than the average score of the *TC* participants (see Figure 3). Additionally, the standard deviation of the evaluation score dropped 44% from the *TC* to *TCO* condition and the long-tail of performance in the *TC* condition disappeared, as we had hoped.

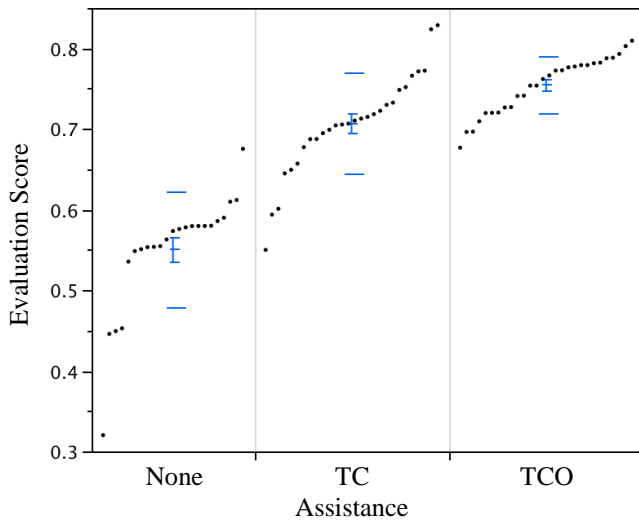


Figure 3: The evaluation scores show that the MCA advice in the TCO condition significantly improved performance, reduced the performance variation, and eliminated the long-tail of performance seen in the other conditions.

Post-test Questionnaire

The post-test questionnaire measured participants’ preferences and perceptions. After the test, participants were asked how much they agreed with this statement: “The task was difficult to complete” (1–7 Likert scale, 7 being strongly agree). Participants reported a significant difference for perception of overall task difficulty (ANOVA, $F(2,72)=3.1917, p<0.05$). A Tukey post-hoc test showed a significant difference between the TCO and None condition (3.75 vs. 4.87). The mean for the TC condition, 4.39, was between the two other conditions but was not significantly different from them.

There was some concern that providing a suggested schedule and recommending tasks to skip might actually increase participants stress if they interpreted that advice as an indication that they were underperforming. After the test, participants were asked on a 1–7 scale “The way the software gave me work was stressful.” An ANOVA showed that there were no significant differences among the conditions ($F(2,72)=0.17, p=0.84$) so our concern luckily was unfounded.

Email Classification and the Task-Centric Action List

The RADAR evaluation uses novice users, who initially may have difficulty effectively judging whether the email classifier’s labels are correct. Too many false positives might confuse them, causing them to waste time, so we tuned the classifier to favor precision over recall. Examination of the classifier’s behavior shows that it did perform as desired. This year’s email set has 123 emails containing 102 tasks labels. The classifier correctly found 47 tasks and incorrectly suggested 6 other tasks (false positives): $precision=0.887 (47/53)$ and $recall=0.461 (47/102)$.

We examined how well the task-centric user interfaces helped participants evaluate the suggestions of the Email Classifier. The following table lists the average number of tasks for each outcome.

	TC		TCO	
True Positives (TP)	47.0		47.0	
Viewed	43.6	100.0%	38.4	100.0%
Completed	38.4	88.0%	34.1	89.0%
Deleted	2.0	4.5%	1.9	4.8%
Ignored	3.3	7.5%	2.4	6.1%
Not Viewed	3.4		8.6	
False Positives (FP)	6.0		6.0	
Viewed	5.8	100.0%	5.3	100.0%
Completed	0.7	12.3%	0.8	14.3%
Deleted	2.7	46.9%	2.5	48.3%
Ignored	2.5	42.6%	2.0	37.4%
Not Viewed	0.6		0.7	
False Negatives (FN)				
Completed	4.5		2.0	
True Negatives (TN)				
Completed	4.3		1.8	
FP & TN Completed	5.0		2.6	

For the 47 correctly classified tasks (TP) that participants inspected (Viewed), participants completed the majority of them, rarely erroneously deleting any. Additionally, for the six incorrectly classified tasks (FP) that the participants inspected (Viewed), participants deleted or ignored the vast majority of them, only occasionally erroneously completing one. However, participants did not complete many tasks that the classifier missed (FN), and they also created and then completed some tasks when they should not have (that were correctly not marked as tasks: TN). The TC participants completed over twice as many TN compared with the TCO participants (4.3 vs. 1.8; $t(54)=2.5152, p<0.02$). Overall, taking all types of commission errors together (FP Completed + TN Completed), the TC participants incorrectly completed on average 5.0 tasks, and the TCO participants incorrectly completed 2.6 tasks.

We counted the number of tasks participants in the None condition found and subsequently completed. For those participants the Email Classifier was disabled, so they had to inspect emails for tasks. Those participants correctly completed 43.7 tasks on average but incorrectly completed 10.3 tasks on average (equivalent to TN), the errors accounting for 19% of the tasks they completed. So while the TCO participants did make errors based upon AI suggestions, the participants without the assistance made a factor of up to four times *more* mistakes (10.3 vs. 2.6).

The number of classified emails varied among participants. In the TC condition, participants viewed between 36 and 52 emails with 20 of 28 participants looking at all 52 emails. In the TCO condition, participants viewed between 28 and 52

emails with 10 of 28 participants looking at all 52 emails³. Overall, we observed that about half of the participants chose to look at all of the emails to which RADAR assigned a task, while others chose to ignore some. The number of participants who ignored emails was higher in the *TCO* condition likely due to the MCA's recommendation to skip tasks.

In the "Possibly Conference-Related Emails" table in the Action List (see in Figure 1(e)), RADAR listed 28 emails. The number of those emails viewed by participants varied from 0 to 28 in both conditions, although the average number was higher in the *TC* condition (3.4 vs. 9.6; $t(54)=3.0858, p<0.005$). Again, participants varied significantly with respect to whether or not they took advantage this aspect of RADAR's assistance.

Effects of the MCA's Task Strategy Recommendations

Since participants earned significantly better evaluation scores in the *TCO* condition than in the *TC* condition, we examined the completed tasks to see how MCA advice may have impacted the score. *TC* participants completed more total tasks (65.5 vs. 55.3; $t(54)=2.4770, p<0.02$) and more non-critical tasks (54.0 vs. 44.9; $t(54)=2.671, p<0.02$) than *TCO* participants did.

The MCA identified five critical task types: "Optimize Schedule" (run the Schedule Optimizer), "Publish Schedule" (run script that updates the schedule on the conference website), "Bulk Website Update" (change the same kind of information for many people on the website), "Reschedule Vendor Orders" (fix the vendors associated with events that moved in the schedule), and "Send a Briefing" (write a briefing for the conference chairperson). The following table shows the number of participants in each condition who completed each of the critical tasks at least once.

Task Type	TC	TCO
Optimize Schedule	27	28
Publish Schedule	27	28
Bulk Website Update	13	25
Reschedule Vendor Orders	3	6
Send Briefing	25	28

The "Reschedule Vendor Orders" task takes about 30 minutes to do so few participants in either condition finished it completely, though *TCO* holds a slight, albeit non-significant edge. However, the percentage of correctly scheduled vendor orders (a measure of partial progress) was significantly higher in the *TCO* condition than in the *TC* condition (51% vs. 29%; $t(54)=2.3400, p<0.05$). In both conditions this percentage ranged from 2%–100%.

Additionally, the percentage of money wasted on incorrectly scheduled vendor orders (another measure of partial progress) significantly dropped in the *TCO* condition (30% vs. 66%; $t(54)=3.3061, p<0.01$). Again, the variation was

wide in both conditions, with the percentage ranging from <0.1% to 100%.

The following table shows that participants generally complied with the critical task warnings that MCA issued.

Task Type	Issued	Complied	%
Late Critical	93	83	89%
Wrong Critical	25	14	56%
Early Critical	1	0	0%
Total	112	97	83%

Compliance with the "Late Critical" warnings was high. However, participants did not allow follow the "Wrong Critical" alerts. Five of these subjects seemed to be averse to quitting what they were currently working on. This could be exacerbated by the fact that subjects are instructed that critical tasks are special, and therefore they might believe that finishing the current one is more important than following the warning's advice.

In the *TCO* condition, the average position of a task in the suggested schedule at the time that it was finished (either completed or deleted) was 5.0. Finished tasks were in the top position 21% of the time and within the top five 62% of the time. Since the *TC* condition does not provide a suggested schedule, we computed the position of the task in the Action List when it was finished. In the *TC* condition, the average position of tasks when it was finished was 11.6. Finished tasks were in the top position 18% of the time and within the top five 37% of the time.

Finally, we found no significant difference for the number of times that participants followed the "Next Suggested Task" link (19.2 in *TCO* vs. 17.8 in *TC*; $t(54)=0.3246, n.s.$). However, the number of times that participants followed the "Next Suggested Task" link varied significantly: 0–56 times in *TCO* and 0–58 times in *TC*.

Discussion

The participants clearly found the AI's assistance helpful in performing their tasks, and they were able to understand and override the AI's suggestions. Participant reported that the test was more difficult without the AI assistance. The manner in which the AI assisted the user did not increase stress, indicating that the advice was presented in a helpful manner. When assisted by the AI, participants performed objectively better and reported a subjective preference for that condition over the others.

We looked for reasons why participants did not seem to be following the MCA's recommendation for the specific next task to do. It appears that users often were skipping the top one or two tasks over and over, suggesting that they did not want to do those specific tasks for some reason, but did not want to remove it from the list. Thus, participants were relying on the MCA to give them strategic advice of an overall order, but felt comfortable looking within the top few recommendations. However, even within the *TCO* condition we found a variation in the amount of advice that par-

³ One outlier in the *TCO* condition only viewed 15 emails.

ticipants followed. This lends support to our mixed initiative user interface rather than one that just presented the next task to the user. Our pop-up alerts for critical tasks also proved to be a successful way to focus the user's attention on critical tasks they seemed to be ignoring in the other views.

CONCLUSION AND FUTURE WORK

Now that the RADAR techniques have proven so successful in our lab study, we are eager to transition them to a real email system with on-line learning. The lessons learned from the iterative design of the test version will be invaluable in making such a transition. The main hurdle is making the AI components sufficiently robust for use with real-world tasks and emails, and in situations where it may be more difficult to integrate the AI and the user interface with real forms that can be used to perform the tasks.

ACKNOWLEDGMENTS

The authors thank Michael Freed, Geoff Gordan, Jeffery Hansen, Jordan Hayes, Javier Hernandez, Matt Lahut, Ken Mohnkern, Pablo-Alejandro Quinones, Bradley Schmerl, Nicholas Sherman, Stephen Smith, Fernando de la Torre, Pradeep Varakantham, Jigar Vora, Yiming Yang, Shinjae Yoo, and Gabriel Zenarosa. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010.

REFERENCES

1. Bellotti, V., Ducheneaut, N., Howard, M., Smith, I. and Grinter, R.E. Quality Versus Quantity: E-Mail-Centric Task Management and Its Relation With Overload. *Human-Computer Interaction* 20, 1/2 (2005), 89–138.
2. Ducheneaut, N. and Bellotti, V. E-mail as Habitat: An Exploration of Embedded Personal Information Management. *interactions* 8, 5 (2001), 30–38.
3. Fink, E., Bardak, U., Rothrock, B. and Carbonell, J.G. Scheduling with Uncertain Resources: Collaboration with the User. *Proc. IEEE SMC*, IEEE Press (2006), 11–17.
4. Freed, M., Carbonell, J., Gordon, G., Hayes, J., Myers, B., Siewiorek, D., Smith, S., Steinfeld, A. and Tomasic, A. RADAR: A Personal Assistant that Learns to Reduce Email Overload. *Proc. AAAI-08*, AAAI Press (2008), 1287–1293.
5. Gwizdka, J. TaskView: Design and Evaluation of a Task-based Email Interface. *Proc. CASCON*, IBM Press (2002).
6. Kumar, M., Das, D. and Rudnicky, A.I. Summarizing Non-textual Events with a ‘Briefing’ Focus. *Proc. RIAO*, Centre De Hautes Etudes Internationales D’Informatique Documentaire (2007).
7. Steinfeld, A., Bennett, S.R., Cunningham, K., Lahut, M., Quinones, P.-A., Wexler, D., Siewiorek, D., Hayes, J., Cohen, P., Fitzgerald, J., Hansson, O., Pool, M. and Drummond, M. Evaluation of an Integrated Multi-Task Machine Learning System with Humans in the Loop. *Proc. PerMIS*, NIST (2007).
8. Steinfeld, A., Quinones, P.-A., Zimmerman, J., Bennett, S.R. and Siewiorek, D. Survey Measures for Evaluation of Cognitive Assistants. *Proc. PerMIS*, NIST (2007).
9. Stylos, J., Myers, B.A. and Faulring, A. Citrine: Providing Intelligent Copy and Paste. *Proc. UIST*, ACM Press (2004), 185–188.
10. Whittaker, S., Bellotti, V. and Gwizdka, J. Email in Personal Information Management. *CACM* 49, 1 (2006), 68–73.
11. Whittaker, S. and Sidner, C. Email Overload: Exploring Personal Information Management of Email. *Proc. CHI*, ACM Press (1996), 276–283.
12. Zimmerman, J., Tomasic, A., Simmons, I., Hargraves, I., Mohnkern, K., Cornwell, J. and McGuire, R.M. VIO: A Mixed-initiative Approach to Learning and Automating Procedural Update Tasks. *Proc. CHI*, ACM Press (2007), 1445–1454.