Gwizdka, J. (2002). TaskView - Design and Evaluation of a Task-based Email Interface. Proceedings of the IBM Centers for Advanced Studies Conference - CASCON 2002. IBM: Toronto, Canada. 136-145.

TaskView – Design and Evaluation of a Task-based Email Interface

Jacek Gwizdka
Interactive Media Laboratory, Knowledge Media Design Institute
University of Toronto
5 King's College Rd., Toronto, Ontario, Canada M5S 3G8
cascon2002@gwizdka.com

Abstract

Email was originally designed as a tool for asynchronous communication. However, as the number of messages increased, so did their variety. A wide range of new and unforeseen email tasks reflects this variety. One of the most commonly performed activities in email is management of pending tasks. This research focuses on how to support this activity in email and explores solutions that use different external representations of messages and associated tasks. Central to this research is understanding the role of both external artifacts in managing multiple pending tasks, as well as internal representations and processes and how they can be linked to external representations. In a recent study we compared the effects of two email interfaces (Microsoft Outlook Inbox and TimeStore-TaskView) on efficiency and effectiveness of information finding in email messages. We found that TimeStore-TaskView interface was overall faster for finding information related to task dates, time and task overviews, while the Inbox interface was faster for finding information from subject lines, senders or from the message body. Based on the results from the study, we are in the process of designing a modified email prototype and a follow-up user study.

Keywords

Email interface, task management, prospective memory, external representations, visualization.

1. Introduction and Motivation

We are witnessing a dramatic growth of email, both in terms of the number of mailboxes and in terms of the number of messages. At the end of 2000 there were 891 million email accounts worldwide, 67% more than in 1999 [20]. In 1998 an estimated 3 billion messages were sent every day in the USA. There is also an increasing variety of information types carried by emails. As a result people spend a lot of time in email programs, which became their primary *electronic habitat* [4]. A habitat used not only for asynchronous communication, but also for a range of new and unforeseen email tasks, such as scheduling, management of to-do's, reminders and contacts [23]. Email is also used for documenting activities, and file transfer [4], [22].

2. Related Work

A number of email systems have focused on message categorization (e.g. [12], [14]). While email classification is important, there was often no clear sense what user task was being supported. Other systems have focused on supporting the "original" email task, that is on asynchronous communication, for example, by combining visualization of conversational threads with timelines [17] or by providing relevant context for reading and composing messages. There have been also innovations at the user interface, such as the placement of messages employing a pile metaphor to support tracking of tasks in email [1].

However, there are relatively few attempts to support a wider range of tasks in email. Two general approaches to such support are possible: 1) make it easy for information to be transferred from email to other media, where the tasks are typically performed; 2) add support for the "new" tasks within email programs. We pursue the latter approach.

2.1. Our Previous Studies

To explore the problem, two preliminary field studies of email and personal information management tool use were conducted [7], [8]. We found that people often use email programs as a time and task management tool (confirming the previous studies by other researchers [22]). Inboxes are often used to keep those messages referring to the future that cannot be dealt with upon their arrival. In the second study [8], 18 of the 19 participants used messages as reminders about email tasks, while 15 of the 19 used messages as reminders about non-email tasks and events. Currently, dealing with these messages in email requires users to periodically review the lists of messages. A few participants of the second study [8] consciously limited this review process to one screen-full of emails. This approach depended on their role in organization (for example, a manager knew, that if she misses an important email, the other party would remind again), and on the volume of email traffic. One participant employed this *limitation strategy* along with *re-emailing to oneself strategy*. He re-emailed himself those messages that required a future action when they started to disappear from screen. A few other participants dealt with this situation by flagging selected messages.

Microsoft Outlook, an example of a widely used commercial email client program, provides a number of features that support various aspects of managing pending tasks (e.g. a to-do list, a calendar, general email flags, specialized reminder flags along with a type of action required) as well as temporal information organization (e.g. journal). However, only a couple of Outlook users who participated in our study made use of some of those features. What is even more surprising, study conducted among Microsoft employees [3] found that, for the most part, they have not used these features as well.

In our studies, we found that information from these messages is often not transferred out of email, presumably because of the high cost of doing so. This high cost is often due to the lack of integration of email with other software applications or media. Thus this research focuses on supporting the management of pending tasks in email. The term *pending task* is used here to denote any activity that is to be performed in the future, such as attending a meeting, visiting a friend, writing a paper, or replying to an email.

3. Research Background

How should information that refers to the future be presented in computing devices? How should the management of pending tasks in email programs be supported? Theories and models of prospective memory, which aim to explain processes involved in *remembering to remember to do something in the future* (e.g. encoding, monitoring, reminding), are particularly relevant. Other areas of study include external and distributed cognition, where the external environment is seen as playing an integral role in cognition and task execution, external representations of information, which can have a critical impact on how that information will be used in a task [6], [25], and studies of temporal and spatial reasoning, which argue that people use spatially coded mental models in reasoning about temporal relations [21]. Specific user interfaces to future email should also provide suitable affordances and visualizations that utilize properties of visual perception.

The role of external representations and their interaction with internal representations and processes is important to understanding how people manage multiple pending tasks. It has been argued that office desks are organized to remind about tasks [13]. Spatial arrangement is used to represent activities, their priorities, temporal dependencies and relationships among multiple tasks. Management of everyday activities relies heavily on such spatial placement and manipulation of physical objects [6]. The same user habits can be brought to bear on the management of pending tasks in email, and more generally, in the computer's desktop interface. The research question that we would like to answer is what external representations of pending tasks provide the best support for their management.

4. Design

Time is inherent in the management of pending tasks. We outline the design space of alternative email user interfaces by describing the relationships between messages and time.

4.1. Time in Email

Email messages may contain multiple references to time. The two most common temporal references contained in messages are message arrival time and message reference time. The message arrival time is always in the past, while messages can refer to the past, the present, or the future. These two timelines are embedded, explicitly or implicitly, in email messages.

4.2. Email Interfaces

The linear, tabular format of email folders has been designed to support *one-touch model* of dealing with email messages [22]. In this model, once a new message has been read, the email user is expected to execute an immediate action which can be a combination of one or more of the following: 1) respond to the message, 2) delete it, or 3) file it. Message processing should thereby be completed. The message may later be retrieved from the archive, but it is not *active* any more. This model of processing, however, does not work for messages that cannot be acted upon immediately after their arrival. The problem is clearly apparent in the case of messages containing some type of future reference (e.g. messages carrying pending tasks). These messages need to be kept around, and, typically, they remain in the inbox. Email users perform thus the fourth possible action: 4) leave message in inbox. The one-touch model breaks down when more and more messages are left in the inbox. The model supports past and present only, it does not support the future. Users are forced to repeatedly review messages left in their inboxes, they cope with this constraint by employing a variety of strategies, for example, by limiting the reviewing process to one screen full of messages.

Such issues can be avoided by using the two timelines embedded in email messages to support their management. The message arrival timeline can be used to facilitate retrieval of messages by bringing human autobiographical memory to bear on retrieval of messages. The message temporal reference timeline can be used to support the management of messages referring to the future.

TimeStore, a novel, time-based email interface proposed and developed by Baecker R. et al. [24], is an example of the first approach. Messages are automatically organized by time and by sender and displayed on a two-dimensional grid (Figure 1). The two-dimensional representation allows locating messages by using cues from autobiographical memory: when the message was received and by whom it was sent. Autobiographical memory is a memory for events which we have experienced. The authors of the project reported in [11] that users liked the visualization of their email and found it useful for retrieval of both old (*inactive*) and new (*active*)

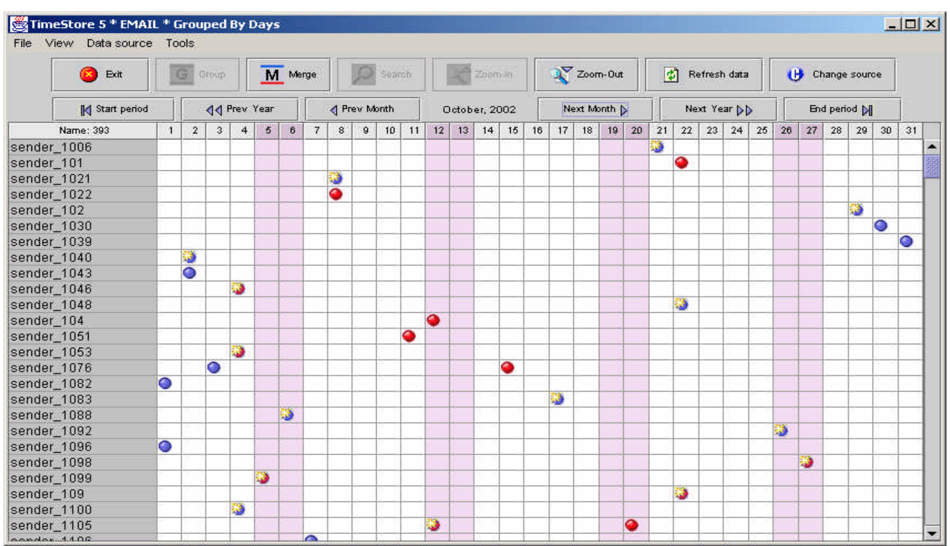

Figure 1. TimeStore interface [24] - one month view (auto-generated data).

messages. The use of message received time to organize email messages is similar to Outlook's feature called *Journal*. However, in the Outlook's Journal email messages are displayed on a horizontal linear timeline and there is no further organization of messages provided.

The second type of timeline, the temporal reference of messages, is used to facilitate the management of pending tasks embedded in messages in TimeStore-TaskView. The TimeStore-TaskView interface is based on TimeStore and uses the same graphical representation (Figure 2). In TimeStore-TaskView, tasks embedded in messages are represented by small icons on a two-dimensional grid with temporal and other task information shown on the horizontal and vertical axis, respectively. Other task attributes include sender, subject, or keywords extracted from the message body (user selectable). Navigation back and forward in time is provided. Displayed time period can be between one day and one year. The message body can be viewed by double clicking on the corresponding task icon.

The main focus of this research is on the presentation of pending task information. Hence, we do not deal with extraction of temporal task attributes from email messages. An initial assumption was made that this information had been extracted and that one message corresponds to one task. TimeStore-TaskView presents only active messages, that is messages with future references containing pending tasks. The presentation is in future time, referencing pending tasks, while in TimeStore the presentation is in past time, arranged according to message arrival time. The main differences between external representations containing past or future references are shown in Table 1.

Characteristic	past	future
Temporal-reference	past	future
Time-frame	years→today	today→months
Typical number of items	10s - 1,000,000s	10s -100s
Temporal-perspective	own – auto-biographical	own & others' - not auto-biographical
Retrieval goal	archive and current	pending tasks, deadlines, delayed intentions

Table 1. Main differences between external representations containing past or future references.

5. User study

A user study was conducted to examine the proposed visualization of pending tasks as implemented in the TimeStore-TaskView interface. A typical email inbox (Microsoft Outlook) served as a benchmark email interface (Figure 3). Messages used in the study contained pending tasks.

In the Outlook Inbox those tasks contained in messages are presented as textual fields in a

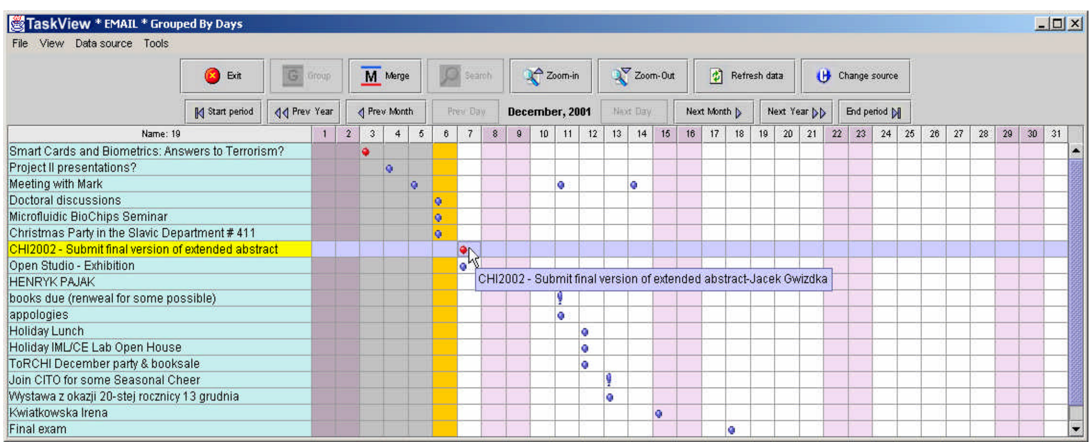

Figure 2. TaskView interface (based on TimeStore [24]). Shown is monthly view with pending tasks sorted by time.

tabular view of Inbox folder. In the TimeStore-TaskView interface tasks contained in messages are represented on a two-dimensional grid.

5.1. Hypotheses

The overall expectation was that the temporal visualization used in TaskView would increase the effectiveness and efficiency of pending task information retrieval. It was also expected that users would prefer the TaskView interface. Five specific hypotheses were formulated:

Efficiency Hypothesis. Efficiency as measured by performance time to complete information retrieval tasks will be higher overall in TaskView than in the Inbox condition.

Effectiveness Hypothesis. Effectiveness as measured by recall from memory and number of correctly performed IR tasks will be higher overall in TaskView than in the Inbox condition.

Flexibility of Closure Hypothesis. Participants with low level of flexibility of closure (FC) will perform worse (in terms efficiency and effectiveness) in Inbox than in TaskView. FC measures how well people can find a pattern in a distracting background. We employed this cognitive measure as finding an email message requires locating the message representation (graphical or textual) among other email messages. High and low measures of flexibility of closure are similar to field independency and dependency, respectively [5].

Visual Memory Hypothesis. Participants with low visual memory (VM) will perform worse (in terms efficiency and effectiveness) in TaskView than in Inbox. VM is used by people to remember graphical information presented on screen.

STM Hypothesis. Participants with high short-term memory (STM) will perform better in Inbox than in TaskView. STM plays a role of an input buffer and a set of memory registers used in human information processing. It mediates verbal and visual information incoming from our senses. This hypothesis derives from the role that STM plays as an input buffer and from the expectation that Inbox requires more information to be kept in the input buffer than TaskView.

5.2. Method

A mixed factorial design was used with user interface as an independent within subject factor (2 levels: Inbox and TaskView). There were two sessions. Each subject used a different interface in each session. The design was balanced with respect to the order of interface use. There were

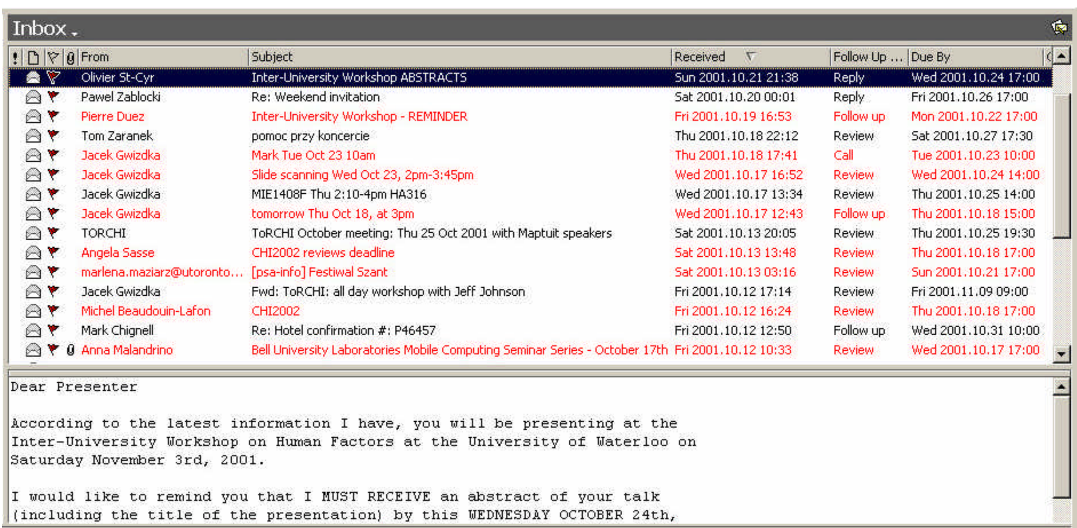

Figure 3. Outlook Inbox interface. In addition to the fields present in the standard Inbox view (various flags, *From*, *Subject*, and *Received*), shown are two fields (*Follow Up By* and *Due By*) that correspond to the temporal information presented graphically in the TaskView interface (Figure 2).

also a number of independent between subject factors which are described in section 5.6.

5.3. Apparatus

Two email programs were used in the experiment: Outlook and TaskView. The user interfaces for these programs, Inbox and TaskView respectively, were described in section 4. The programs were installed on a desktop PC in the experimenter's office. Participants' interaction was recorded using the Camtasia software for capturing activity on the computer screen. Email inboxes in both programs were populated with the same 44 messages containing pending tasks. The messages were selected from a larger corpus of emails created from real-life messages sanitized by removing identifying information that would indicate who was speaking or being referred to in each message. A set of 49 questions about pending tasks contained in the messages was created. The questions were categorized into two main types: "*C*" and "*D*". Type "C" questions refer to message subject, sender, or content. Type "D" questions refer to pending task date, time or task overview (e.g. number of tasks in a specified period of time). In the remainder of the paper, finding answers to questions of type "C" or "D" is referred to as task "C" or task "D" respectively.

5.4. Participants

21 subjects participated in the experiment. 18 participants were graduate students from the University of Toronto (7 Master students and 11 PhD students) and 3 participants were full-time employees from outside companies or government agencies. There were 7 females and 14 males. Participants were screened for at least moderate use of email and for the use of email to receive task information. On the average, participants used email for 6 years. Participants were paid $30 for their time ($10 per hour).

5.5. Procedure

The study consisted of four on-line questionnaires and two sessions conducted in the experimenter's office. The sessions were spread at least 2 days apart (2 to 7 days). Participants used a different email interface in each session.

Before coming to the first session, participants filled out an on-line survey containing demographic and email-habit questions. Each session consisted of study protocol explanation, user interface training, and the main task. The main task was to find information about pending tasks in email messages. Information finding was driven by multiple-choice questions displayed on screen. 21 questions were drawn randomly from a larger set of 49 questions. After the main task was completed, a couple of cognitive tests were administered (different tests were administered in each session. All tests that were used are listed in section 5.6). In the "TaskView" session, after using the new interface, participants filled out a subjective preference questionnaire. At the end of each session, participants were asked to freely recall information about pending tasks which they had looked up in email messages earlier in the session. After each of the sessions participants filled out an on-line questionnaire containing the same set of questions as they answered during the session (order of questions was randomized). Additionally, after the first session, participants filled out a 44-item Big Five personality trait questionnaire [2].

5.6. Measures

Apart from the within-subject factor, there were a number of independent between subject factors:

1. scores on four tests of cognitive abilities: flexibility of closure (CF2), visual memory for shapes (MV1), building map memory (MV2), and short-term memory (measured by auditory digit span test) [5]. Flexibility of closure is closely related to field independence / dependence [5].
2. personality traits measured by 44-item Big Five Inventory [2].
3. self-reported email habits, focusing on handling of pending tasks in email.

Dependent measures included:

1. efficiency, measured by the time participants taken to answer questions
2. effectiveness, measured by the number of:
   a. correct answers during the task,
   b. correct answers one day after the session,
   c. items freely recalled immediately after the task

5.7. Results

Paired t-tests assessed whether or not there was a difference in performance between the two user interface conditions: TaskView and Inbox. No differences were found with respect to effectiveness measured by the number of correct answers during and after the sessions, nor for free recall from memory. There was also no difference for the total performance time. However, there was a significant difference between the two interfaces in performance time for task "C" (t_{20}= –3.192, p=0.005) and for task "D" (t_{20}=2.246, p=0.036). As can be seen from Figure 4, for task "C" participants were faster in Inbox (median performance time t=34 sec.) than in TaskView (t=47 sec.), while for task "D" (Figure 5) participants were faster in TaskView (t=22.5 sec.) than in Inbox (t=29 sec.).

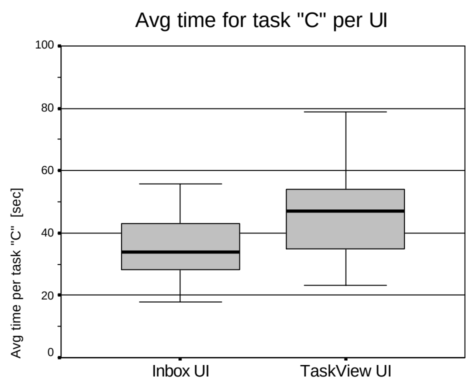

Figure 4. Average time for task "C" for two UIs.

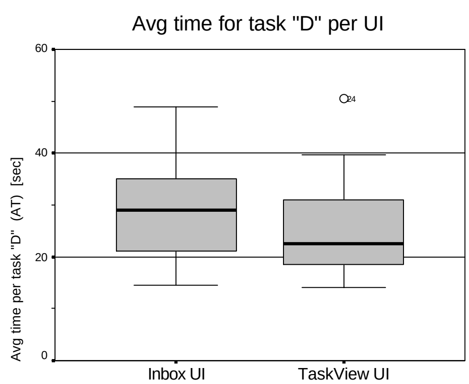

Figure 5. Average time for task "D" for two UIs.

Next multivariate analyses of variance (MANOVA) with repeated measures were conducted. Using the two performance time variables (for task "C" and "D"), there was a borderline significant interaction between interface design and the level of combined visual memory score[1] ($F_{2,18}$=2.585, p=0.103, ?^{2}=0.223[2]). This was due to the interaction having a borderline significant effect on the task "C" time ($F_{1,19}$=3.425, p=0.080 ?^{2}=0.153).

The corresponding data chart is shown in Figure 6, where it can be seen that high visual memory people tend to be less affected by the interface condition than low visual memory people, who tend to be slower.

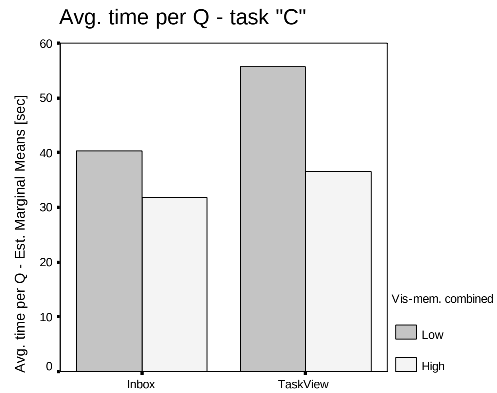

Figure 6. Interaction between UI (for task "C") and combined visual memory score.

There was a corresponding main effect for task "D" time. Tasks "D" times were significantly slower ($F_{1,19}$=5.062, p=0.036 ?2=0.210; median 29 seconds for the Inbox compared to a median 22.5 seconds for TaskView) when using the Inbox interface.

Similar analysis was performed for flexibility of closure. Although no significant interaction effect between interface design and the level of flexibility of closure (field independence) was found ($F_{2,18}$=2.167, p=0.143, ?^{2}=0.194), the magnitude of ?^{2} indicates a large effect size. For the corresponding univariate test the interaction effect on task "D" was significant at the level of p=0.05 ($F_{2,18}$=4.515, p=0.047 ?^{2}=0.192). As can

[1] combined visual memory (MVCOM) was calculated as :

MVCOM = (MV1 + MV2) / 2

[2] ?^{2} refers to the partial ETA squared reported by SPSS. ?^{2} measures the proportion of variance (PV) that estimates the size of an effect [[15]].

be seen from Figure 7 there is a tendency for low flexibility of closure subjects to perform disproportionately slower in the Inbox interface than in the TaskView interface (approximately 9 seconds slower), while subjects high on flexibility of closure perform about the same in both interfaces (with a difference of less than a second).

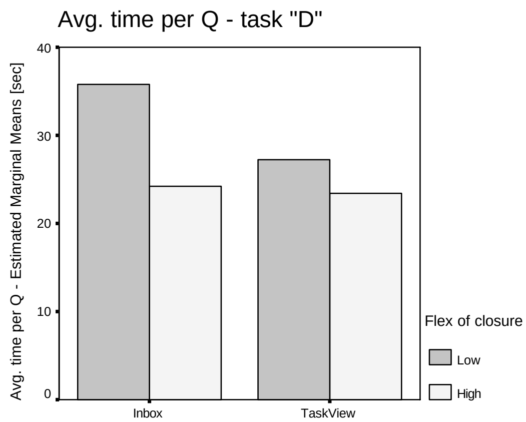

Figure 7. Interaction between UI (for task "D") and flexibility of closure.

There was a corresponding main effect for task "C" time. Tasks "C" times were significantly faster ($F_{1,19}$=9.659, p=0.006 ?^{2}=0.337; median 34 seconds for the Inbox compared to a median 47 seconds for TaskView) when using the Inbox interface.

No significant interaction effects were found for short-term memory. Also, all participants indicated their subjective preference for the TaskView interface for managing pending tasks.

6. Discussion and Design Implications

The TaskView interface performed better in type "D" of information retrieval from messages related to pending tasks. At the same time, the study showed the limitations of the current version of TaskView. The benefits on performance time were observed for task "D", while the performance for task "C" was worse in TaskView than in Inbox. The *Efficiency Hypothesis* was thus confirmed only for task "D". This suggests that the next iteration of the TaskView interface should be redesigned to achieve at least the level of Inbox' efficiency for task "C", if not to improve it. One possible simple modification would be to display sender and subject together in the left-hand column in TaskView. This would avoid having to switch between displaying sender versus subject information.

The study also showed the role of individual differences in interacting with external representations which mediate the management of pending tasks. Tabular, textual inbox and two-dimensional graphical layout of tasks with additional textual information (i.e. list of senders, subjects or keywords) in TaskView put different demands on different people. Users with low visual memory tended (for task "C") to perform worse in TaskView (confirming the *Visual Memory Hypothesis*). Users with low flexibility of closure tended (for task "D") to perform slower in Inbox (confirming the *Flexibility Of Closure Hypothesis*). Embedding messages in the inbox among other messages may require more discrimination, which creates a disadvantage for this population group. These results suggests that it is important to consider alternative interfaces for different population groups and for different tasks. The *Short-term memory Hypothesis* was not confirmed.

Employed measures of effectiveness were based on the number of correct answers and on recall from memory. Performance time and the number of correct answers given by study subjects are not independent; more correct answers can be found at the cost of time taken to locate information required to answer questions. Participants took as much time as they needed to answers correctly questions (79% to 100%, with two outliers at 74% and 63%), therefore we observed a lot of variation in performance times and very little variation and no significant effects in the number of correct answers. Lack of observed effects of the two interfaces on recall from memory can be explained by noticing a number of potentially confounding factors. In analyzing items written down by the study subjects in the process of free memory recall, we saw that these items were often influenced by multiple choice questions and sometimes used similar wording. The questions were asked in the same way in both interface conditions, and, thus, they influenced the process of encoding pending task information by participants in their memory. Hence, we cannot answer whether there is no effect or only a small effect of user interface on remembering. The *Effectiveness hypothesis* remains unconfirmed.

7. Conclusions and Future Work

The study demonstrated the value of the TaskView interface and the benefit of task visualization on efficiency of finding information in messages related to pending tasks. It revealed the trade-off between two different types of information retrieval tasks: task “C” and task “D”. The TimeStore-TaskView interface benefited the task “D” performance, while it adversely effected the task “C” performance. The benefits depended on the type of the person (cognitive ability).

Spatial memory has been shown to be effective in the management of documents in desktop environments [16]. We suggest that it can also be used effectively to manage email messages containing pending tasks. By using location-based organization of messages users avoid the expensive process of creating, naming and managing folders. We are designing and implementing a prototype email UI, where users will be able to create their own visual organization of messages on a 2D plane. The interface is based on zoomable user interface (ZUI). An early sketch of the prototype is shown in Figure 8.

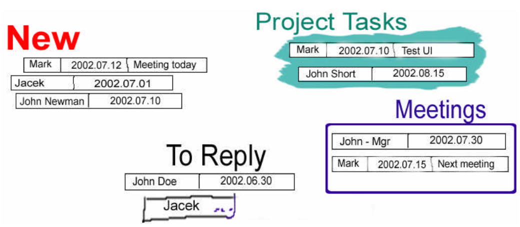

Figure 8. Email Land – new prototype email interface.

8. Summary

This research has shown the resilience of people with different high cognitive abilities to different interfaces. These users remained relatively unaffected by an interface, while people with low cognitive abilities performed significantly different in the two interface conditions. This phenomenon may be similar to novice – expert effects, where novices perform better using an interface, while experts perform equally well on different interfaces. For example, it has been found [19] that expert searchers performed equally well using a conventional (Boolean) query interface and a dynamic hypertext interface, while novice searchers were worse on the conventional interface but close to expert performance on the dynamic hypertext interface.

Email messages containing future references are handled poorly in current email systems. This research examines how external representations of task information at the user interface can improve management and awareness of pending tasks that are encoded within email messages. The physical environment, in which people perform everyday activities, is highly spatial and flexible. We expect that bringing some of these characteristic into the email environment will better support a variety of tasks performed in email. Based on the results from the study, we are designing a modified email prototype and a follow-up user study.

Acknowledgments

Thanks to Prof. Mark Chignell, my thesis supervisor, for his continued support. Thanks to Prof. Ron Baecker for making TimeStore code available for my research and to Peter Wolf for his help with TimeStore code. One of the studies [8] was conducted at Xerox PARC in collaboration with Dr. Michelle Baldonado. This research has been funded by scholarships from NSERC, OGS, and by grants from CITO and BUL.

Note on Implementation

TaskView was implemented in Java. It uses JACOB, a JAVA-COM Bridge written by Dan Adler (http://danadler.com/jacob/), to communicate with email clients (such as Microsoft Outlook or Qualcomm Eudora). JACOB allows one to call COM Automation components from Java on any Win32 Java Virtual Machine.

About the Author

Jacek Gwizdka is currently a PhD candidate in Human Computer Interaction at the University of Toronto in the Interactive Media Lab where his academic supervisor is Dr. Mark Chignell. Jacek has been involved in studying email use for a couple of years [7],[8]] and is currently exploring alternative email interfaces [9],[10]. Jacek maintains http://www.emailresearch.org./ – an information resource devoted to research related to email. The website contains bibliographies, lists of research labs, projects, individual researchers and commercial email products. Jacek is a co-organizer of a CSCW'2002 workshop on future email interfaces.

Jacek holds a Master degree in electrical engineering from Technical University of £ódz, Poland, as well as a Master degree in Industrial Engineering (Information Systems) from University of Toronto. He has conducted research at Xerox PARC, Hewlett Packard Research Labs, and Fuji Xerox Palo Alto Labs. Jacek's research interests include: studying humans augmented by information technology, as well as designing interfaces supporting *future* information and tools for sensemaking. Jacek has also several years of experience in industrial software development. More information is available at: http://www.gwizdka.com

References

[1] Bälter, O. & Sidner, C. (2000). Bifrost Inbox Organizer: Giving users control over the inbox, Lotus Research TR 00-08.

[2] Benet-Martinez, V. & John, O. P. (1998) Los Cincos Grandes across cultures and ethnic groups: Multitrait multimethod analyses of the Big Five in Spanish and English. Journal of Personality and Social Psychology, 75, 729-750.

[3] Cadiz, JJ., Dabbish, L., Gupta, A. & Venolia, G.D. (2001). Supporting Email Workflow. Microsoft Research Technical Report MSR-TR-2001-88. September 2001.

[4] Ducheneaut, N., Bellotti, V. (2001). Email as habitat: an exploration of embedded PIM. Interactions, 8(5), ACM Press. 30-38.

[5] Ekstrom R.B. et al. (1976). Manual for Kit of Factor-Referenced Cognitive Tests 1976. Education Testing Service: Princeton, NJ, USA.

[6] Gruen, D. (1996). The Role of External Resources in the Management of Multiple Activities. PhD Thesis. UCSD: San Diego, CA, USA.

[7] Gwizdka, J. (2000). Timely Reminders: A Case Study of Temporal Guidance in PIM and Email Tools Usage. Proceedings of CHI'2000. ACM Press. 163-164

[8] Gwizdka, J. (2001). Supporting Prospective Information in Email. Proceedings of CHI'2001. ACM Press. 135-136.

[9] Gwizdka, J. (2001). Reinventing the Inbox: Supporting Prospective Information in Email. CASCON 2001. Poster.

[10] Gwizdka, J. (2002). Reinventing the Inbox: Supporting Prospective Information in Email. Proceedings of CHI'2002. ACM Press. pp.

[11] Jovicic, A. (2000). Implications for the design of email management software. M.Sc. Thesis, Dept. of Computer Science, University of Toronto.

[12] Kiritchenko, S. & Matwin, S. (2001). Email Classification with Co-Training. CASCON 2001.

[13] Malone, T. (1983). How do people organize their desks? Implications for the design of office information systems. ACM TOIS. 1, 99–112.

[14] Mock, K. (2001). An Experimental Framework for Email Categorization and Management. SIGIR 2001. ACM Press.

[15] Murphy K.R., Myors B. (1998). Statistical Power Analysis. A Simple and General Model for Traditional and Model Hypothesis Tests. Lawrence Erlbaum Associates, Publishers: Mahwah, NJ, USA.

[16] Robertson, G., et al. (1998). Data mountain: using spatial memory for document management. UIST 1998 ACP Press. 153–162.

[17] Rohall, S.L., et al. (2001). Information Visualization, Electronic Mail, Threads, Trees. IEEE Symposium on Information Visualization (InfoVis) 2001, October 22-23, 2001, San Diego, CA.

[18] Scaife M., Rogers, Y. (1996). External cognition: How do graphical representations work? International Journal of Human-Computer Studies 45, 185-213.

[19] Tam, J., Bodner, R. and Chignell, M.H. "A Demonstration on How Dynamic Hypertext Can Improve Performance", Proceedings of the Human Factors and Ergonomics Society, Albuquerque, New Mexico, September, 1997.

[20] United Messaging. (2001). 2000 Year End Mailbox Report. http://www.unitedmessaging.com/form.cfm?ID=BA6NA8&CID=13

[21] Vandierendonck, A. & De Vooght, G. (1998). Mental models and working memory in temporal reasoning and spatial reasoning. In V. De Keyser, et al. (eds.). Time and the dynamic control of behavior. Göttingen: Hogrefe & Huber. 383-402

[22] Whittaker, S., Sidner, C. (1996). Email overload: exploring personal information management of email. Proceedings of CHI 96. ACM Press. 276-283.

[23] Whittaker, S., Jones, Q., and Terveen, L. (to appear). Managing Long Term Conversations: Conversation and Contact Management. In HICCS 2002

[24] Yiu, K., Baecker, R.M., Silver, N., and Long, B. (1997). A Time-based Interface for Electronic Mail and Task Management. In Proceedings of HCI International '97. Vol. 2, Elsevier. 19-22.

[25] Zhang, J.& Norman, D. A. (1994). Representations in distributed cognitive tasks. Cognitive Science, 18, 87-122.

[26] Zhang, J (1996). A representational analysis of relational information displays. International Journal of Human-Computer Studies, 45, 59-7