Multitasking and Monotasking: The Effects of Mental Workload on Deferred Task Interruptions

Dario D. Salvucci and Peter Bogunovich

Department of Computer Science
Drexel University
3141 Chestnut St.
Philadelphia, PA 19104, USA
salvucci@cs.drexel.edu

ABSTRACT

Recent research has found that forced interruptions at points of higher mental workload are more disruptive than at points of lower workload. This paper investigates a complementary idea: when users experience deferrable interruptions at points of higher workload, they may tend to defer processing of the interruption until times of lower workload. In an experiment, users performed a mail-browser primary task while being occasionally interrupted by a secondary chat task, evenly distributed between points of higher and lower workload. Analysis showed that 94% of the time, users switched to the interrupting task during periods of lower workload, versus only 6% during periods of higher workload. The results suggest that when interruptions can be deferred, users have a strong tendency to "monotask" until primary-task mental workload has been minimized.

Author Keywords

Multitasking, interruption, attention, problem state, chat, instant messaging.

ACM Classification Keywords

H.5.2. User interfaces: theory and methods; H.1.2 User/ machine systems: human factors, information processing.

General Terms

Human Factors, Design

INTRODUCTION

Computer users switch tasks extremely frequently, roughly every few minutes by one estimate [9]. Researchers have focused especially in the last decade on investigating the nature of task switching and interruptions and their many effects on user behavior and performance. One consistent finding is that task interruptions lead to a decrease in primary-task performance, most notably in terms of a *resumption lag* representing the additional time needed to resume the primary task after interruption [17, 20]. Another robust finding is that the timing of interruptions can affect performance: interruptions occurring at points of higher mental workload are more disruptive and lead to larger resumption lags than those occurring at points of lower mental workload [1, 3, 5, 6, 12].

In this paper we make a complementary claim: When users are alerted to interruptions at points of higher mental workload, they delay processing of the interruption until they have reached a point of lower mental workload. Most experimental work has used forced interruptions in which either the system displaces the primary task with a secondary task at a pre-specified time [1, 12, 17, 20], or participants are asked to respond immediately to an interruption [5, 6]. In contrast, many interruptions are *deferrable interruptions*: an external trigger notifies the user of a pending interruption, but the user may delay processing of the interruption until he or she reaches a desirable stopping point in the primary task. A few recent studies [6, 13, 16, 21] have suggested that, for deferrable interruptions, users indeed tend to "stabilize task state" [13] before responding. However, these studies did not carefully control mental workload, but analyzed workload informally or using hierarchical task models [e.g., 1].

We ran an experiment to test the above claim using an electronic mail customer-support task as the primary task and a chat (instant messaging) task as the interrupting secondary task. Mental workload was carefully controlled for the mail task by requiring that users mentally maintain a critical piece of information during two segments of the task. This type of temporary task-relevant information, which we call the *problem state* [18], has been found to act as a constraining bottleneck on multitasking performance: cognition can only maintain problem-state information for one task at a time [2], and thus task switching incurs additional costs from swapping problem states [4]. The problem-state bottleneck suggests that users will *monotask*—focus exclusively on the primary task—until task problem state has been eliminated (or at least minimized). Our experiment tests this hypothesis by examining whether users, having received an interrupting chat message, process and respond to the message only at points of minimal mental workload in the primary mail task.

EXPERIMENT

The mail task was a customer-service task in which the user answered emails about consumer products and prices. The

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

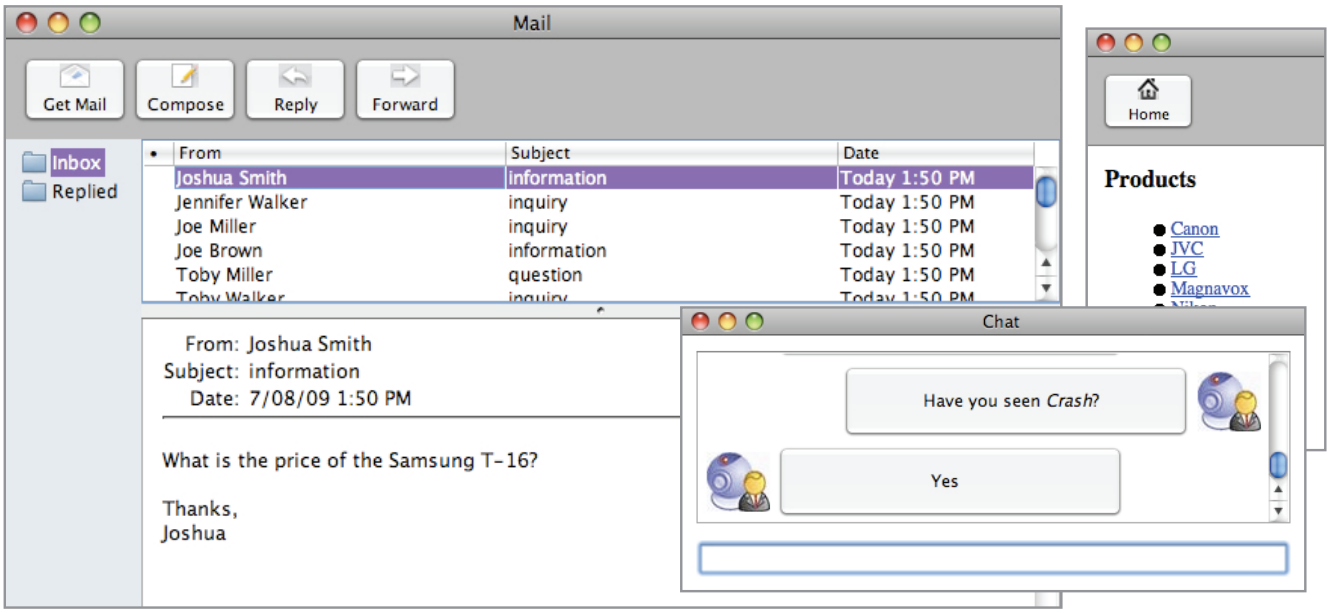

Figure 1: Screen shots of the mail, browser, and chat windows. (Note: In the experiment, these windows were sized and positioned to overlap to a large degree, forcing the user to switch between windows and thus recording these task-switching actions.)

mail and other system windows are shown in Figure 1. The user first selected and read an email, each of which asks for the price of a particular product. The products were generated using real manufacturers with fictitious model numbers comprising one letter and two digits (e.g., "Canon H-44," "Sony M-76"). To find the price of a product, the user switched to a browser window and clicked on (1) the manufacturer, and then (2) the model number. With the price shown on the resulting browser page, the user pressed a button on the mail interface to reply to the email, opening a message composition window. The user then typed the price and clicked a button to send the message. Finally, the user dragged the replied-to email to an archive folder.

The manipulation of mental workload was incorporated into the mail task by requiring the maintenance of temporary information, or *problem state*, during certain stages of the task. After reading the email product information, the user had to remember this information while finding the price in the browser. (Users were instructed not to use copy/paste.) Then, after reading the price, the user had to remember this price while initiating and typing out the response email. The basic steps of the mail task are shown in Figure 2 (discussed further in the next section). The steps depicted as white ovals indicate points at which there is no problem state; for example, the user did not need to remember information to select an email, but only needed to read and maintain product information before proceeding to the next step. Thus, these steps represent points at which a user-delayed interruption would be more likely. In contrast, the steps depicted as gray ovals indicate points at which information did need to be maintained, making user-delayed interruption less likely.

The chat task was based on a standard messaging interface in which messages were shown sequentially, shown in Figure 1. Occasionally, a system-generated prompt message would arrive in the chat application. The messages asked a question about whether the user had seen a film (randomly selected from the 5 Academy Award Best Picture nominees from the past 10 years). Half of the time, a follow-up question asked whether the user liked or would like to see "it"—that is, the user had to recall what film was mentioned in the last message, and this last message was not visible

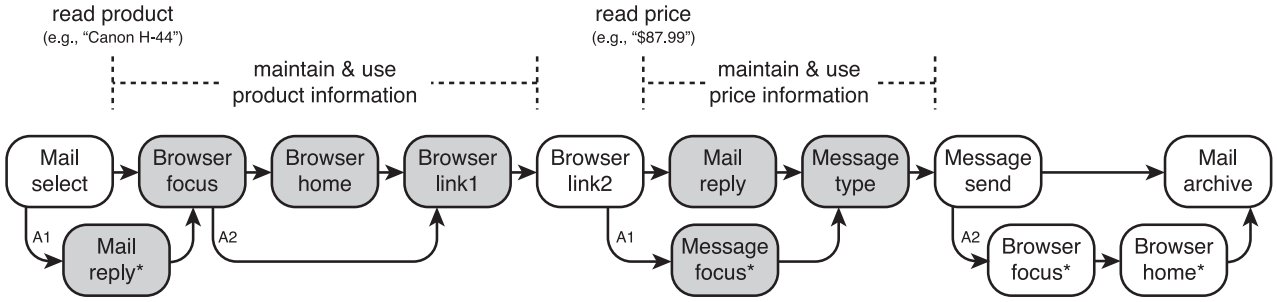

Figure 2: The most common sequence of steps in the mail task (upper sequence), and two alternate sequences (A1 and A2). Steps depicted in gray require maintenance of problem state; steps depicted in white do not.

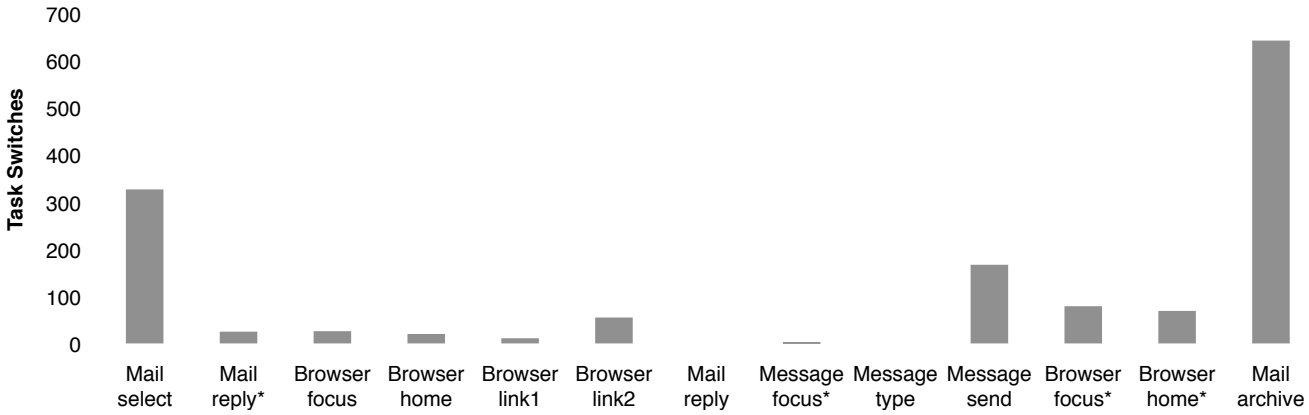

Figure 3: Number of switches to the chat task after a given mail-task event. Note that the vast majority of task switches occurred when problem-state information did not need to be maintained (shown in white).

(though the user could scroll up to see it); thus, the chat task had its own problem-state information (the current film being discussed) which could potentially interfere with the mail task. When a prompt message arrived, the chat interface alerted the user to the message by generating an audible alert and coloring the background of the chat window yellow. Users were instructed to respond to the chat message as soon as they felt comfortable. To respond to the message, the user had to switch to the chat window (the message content was not visible otherwise) and enter "yes" or "no" in response to the question.

The overall task environment was coded in Java Swing to emulate the standard Macintosh applications. All user events (namely mouse actions and keystrokes) were logged by the system. An important aspect of the overall task was that the windows were sized and positioned to overlap to a very large degree. Thus, the user could not see the relevant content of any window unless he or she actively switched to that window by clicking on it; this constraint was necessary to ensure that the system could log switches between tasks, including switches to the chat window to read a chat message, and switches between the mail and browser windows.

Procedure

After being introduced to both tasks, participants performed trials of the mail task and were occasionally interrupted by a chat prompt message. Each trial of the mail task involved responding to a single mail message, including lookup of the product price, sending the response email, and moving the original mail to the archive folder. During each trial, a chat prompt was generated at a pseudo-random point in the trial: the system tracked the user's events during the trial and, after one of eight different events, triggered a chat prompt 50-200 ms after the event—to avoid tying the prompt directly to the event but also to make it unlikely that the user could generate another event before the prompt. The experiment concluded when participants answered chat prompts for all 50 films.

Participants

A total of 20 users (7 female and 13 male) participated in the study. One (female) participant exhibited a radically different behavioral pattern than the other participants and was excluded in the data aggregation below.

RESULTS

We first analyzed behaviors in the mail task alone to understand the sequences of events by which users performed the task. After informal analysis of the recorded protocols, we performed an analysis of the transitions between all events. The main sequence shown as the upper sequence in Figure 2 was found to be the dominant behavior (roughly 80% of mail trials). Users also exhibited two common alternate strategies: clicking "Reply" immediately after reading an email to begin blank response email before browsing for product information (labeled A1 in the figure), and resetting the browser to the home page after sending the response email rather than before browsing (A2).

Using the events found in the sequence analysis, we computed the number of times users switched to the chat task after each mail event. The results are shown in Figure 3, using the same event coloring as Figure 2—events shown in white indicate points of lower mental workload, whereas those in gray indicate points of higher mental workload. As is evident in the figure, users exhibited a strong tendency to switch tasks at points of lower workload (white events). For example, the most task switches occurred after the final event of a trial (Mail archive), and the second-most after selection of the next email (Mail select); at both points, the user did not need to remember problem state for the next step. The last four columns show that users also tended to switch tasks after the response email had been sent, again all points during which no information needed to be remembered. In the middle of the trial, the largest number of task switches occurred after the Browser-link2 event: after clicking this second link, the price information was easily readable on the browser

screen, and thus again no problem state was maintained for the next event.

We can summarize these results by aggregating the number of task switches for each type of event: despite the fact that the prompts occurred roughly equally for points of higher and lower workload (52% vs. 48% respectively), 94% of all user task switches occurred at points of lower workload versus only 6% at points of higher workload. Thus, users showed a strong tendency to postpone the actual processing of the interruption (i.e., reading and responding to the chat message) until points of lower mental workload during which no problem state needed to be maintained.

DISCUSSION

Our experimental results indicate that when users have the option to defer an interrupting task, they have a strong tendency to *monotask* until primary-task mental workload has been minimized. This builds on the results of recent studies [6, 13, 16, 21] by closely controlling workload as the carry-over of specific pieces of information from one task step to the next, allowing for a more detailed analysis of the points of task switching. In an alternate view to automated systems that monitor task workload and deliver notifications when workload is low [8, 10, 11], this result suggests that users themselves can capably handle incoming alerts and defer processing of interruptions until points of lower workload. We would suspect (though further research would need to confirm) that this ability also generalizes to user self-interruptions and discretionary multitasking [14].

Our experiment also helps to clarify one source of mental workload, namely the problem state—temporary information needed for task processing. The maintenance of problem-state information, such as the product name needed while browsing, serves as an important form of workload that can also be associated with a central bottleneck in multitasking behavior [4, 18, 19]. This internal cognitive workload is likely also correlated with external, observable indicators of mental workload (e.g., pupil dilation [3]). Certainly the required information for this experiment—a single product or price—is a rather simple example of problem state; more complex tasks, such as writing a research paper, may involve large-scale conceptual problem states needed to reason about a particular domain (roughly speaking, the information needed for a single working sphere [15]). Future experiments along these lines could further evaluate user monotasking for such complex tasks, potentially involving both longer-term monotasking and occasional forgetting of deferred tasks [7].

ACKNOWLEDGMENTS

This work was funded by ONR grant #N00014-09-1-0096.

REFERENCES

1. Adamczyk, P. D., & Bailey, B. P. (2004). If not now, when? The effects of interruptions at different moments within task execution. *Proc. CHI 2004*, 271-278.
2. Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.
3. Bailey, B. P., & Iqbal, S. T. (2008). Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ToCHI*, *14,* 1-28.
4. Borst, J. P., & Taatgen, N. A. (2007). The costs of multitasking in threaded cognition. *Proc. ICCM 2007*, 133-138.
5. Cutrell, E. B., Czerwinski, M., & Horvitz, E. (2000). Effects of instant messaging interruptions on computing tasks. *Proc. CHI 2000 Extended Abstracts*, 99-100.
6. Czerwinski, M., Cutrell, E., & Horvitz, E. (2000). Instant messaging: Effects of relevance and timing. *Proc. HCI 2000*, 71-76.
7. Dismukes, R. K., & Nowinski, J. (2007). Prospective memory, concurrent task management, and pilot error. In *Attention: From Theory to Practice*. New York: Oxford.
8. Fogarty, J., Hudson, S. E., & Lai, J. (2004). Examining the robustness of sensor-based statistical models of human interruptibility. *Proc. CHI 2004*, 207-214.
9. Gonzalez, V. M., & Mark, G. (2004). "Constant, constant, multi-tasking craziness": Managing multiple working spheres. *Proc. CHI 2004*, 113-120.
10. Horvitz, E., Cadie, C., Paek, T., & Hovel, D. (2003). Models of attention in computing and communication: From principles to applications. *Communications of the ACM*, *46*, 52-59.
11. Hudson, S. E., et al. (2003). Predicting human interruptibility with sensors: A Wizard of Oz feasibility study. *Proc. CHI 2003*, 257-264.
12. Iqbal, S.T., & Bailey, B.P. (2005). Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. *Proc. CHI 2005*, 1489-1492.
13. Iqbal, S. T., & Horvitz, E. (2007). Disruption and recovery of computing tasks: Field study, analysis and directions. *Proc. CHI 2007*, 677-686.
14. Jin, J., & Dabbish, L. A. (2009). Self-interruption on the computer: A typology of discretionary task interleaving. *Proc. CHI 2009*, 1799-1808.
15. Mark, G., Gonzalez, V. M., & Harris, J. (2005). No task left behind? Examining the nature of fragmented work. *Proc. CHI 2005*, 321-330.
16. McFarlane, D. C. (2002). Comparison of four primary methods for coordinating the interruption of people in human–computer interaction. *Human-Computer Interaction*, *17*, 63-139.
17. Monk, C. A., Trafton, J. G., & Boehm-Davis, D. A. (2008). The effect of interruption duration and demand on resuming suspended goals. *Journal of Experimental Psychology: Applied*, *14*, 299-313.
18. Salvucci, D. D., & Taatgen, N. A. (2010). *The Multitasking Mind*. New York: Oxford University Press.
19. Salvucci, D. D., Taatgen, N. A., & Borst, J. P. (2009). Toward a unified theory of the multitasking continuum: From concurrent performance to task switching, interruption, and resumption. *Proc. CHI 2009*, 1819-1828.
20. Trafton, J. G., et al. (2003). Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. *IJHCS*, *58*, 583-603.
21. Wiberg, M., & Whittaker, S. (2005). Managing availability: Supporting lightweight negotiations to handle interruptions. *ACM ToCHI*, *12*, 356-387.