

Measuring the Effects of Interruptions on Task Performance in the User Interface

Brian P. Bailey, Joseph A. Konstan, and John V. Carlis

University of Minnesota
Department of Computer Science and Engineering
Minneapolis, MN 55455

{bailey, konstan, carlis}@cs.umn.edu

ABSTRACT

As users continue offloading more control and responsibility to the computer, coordinating the asynchronous interactions between the user and computer is becoming increasingly important. Without proper coordination, an application attempting to gain the user's attention risks interrupting the user in the midst of performing another task. To justify why an application should avoid interrupting the user whenever possible, we designed an experiment measuring the disruptive effect of an interruption on a user's task performance. The experiment utilized six web-based task categories and two categories of interruption tasks. The results of the experiment demonstrate that (i) a user performs slower on an interrupted task than a non-interrupted task, (ii) the disruptive effect of an interruption differs as a function of task category, and (iii) different interruption tasks cause similar disruptive effects on task performance. These results empirically validate the need to better coordinate user interactions among applications that are competing for the user's attention.

1. INTRODUCTION

As users continue offloading more control and responsibility to the computer through applications such as interface agents, softbots, and push-based information windows, these applications must increasingly compete for the user's attention. User attention must be periodically gained in order to receive additional guidance from the user [4] or to provide a high degree of information awareness [1]. However, because a user is often engaged in another task, an application attempting to gain the user's attention should, if possible, wait for an "opportune moment" rather than immediately interrupting the user's current task.

Waiting for an opportune moment before interrupting someone's task is a social behavior commonly found in human-human interactions. Rarely does a person immediately interrupt another when that person is visibly concentrating on a task, rather, a person often waits for a more opportune moment to gain the other's attention such as when that person finishes or temporarily pauses their current task. The long-term goal of this work is to build a system that manifests this same courteous behavior among

applications competing for the user's attention. In the envisioned system, an application requesting the user's attention would be notified when an opportune moment arises and would then interact with the user at that time.

Although implementing a system that provides this functionality may intuitively appeal to the reader, our goal in this work is to empirically validate and justify its need. In this paper, we report on an experiment designed to measure the disruptive effects of computer-initiated interruptions on a user's task performance. We define any computer-based task presented to a user by a computer application while that user is currently performing another computer-based task as a *computer-initiated* interruption. The phrase *computer-initiated* is used to distinguish an interruption originating from a computer application from an interruption originating from one's external environment such as a telephone call or knock at the door.

In our experiment, subjects performed a series of tasks covering a variety of web-based task categories; adding, counting, image comprehension, reading comprehension, registration, and selection. While performing some of the tasks from each category, a subject was interrupted approximately halfway to completion, i.e., at an inopportune time, with either a news summary or stock-decision task. The remaining tasks were not interrupted and served as control tasks. We then analyzed the performance differences between the interrupted and non-interrupted tasks using the task category and interruption task category as factors.

Analysis of the data revealed that (i) a user performed slower on an interrupted task than a non-interrupted task, (ii) the disruptive effect of an interruption differed as a function of task category (memory load), and (iii) both interruption task categories caused similar disruptive effects on task performance across all task categories. The results of this experiment provide a quantitative justification as to why an application should not be allowed to immediately interrupt a user's task in order to gain their attention; rather, an application should wait for notification of an opportune moment such as when the user reaches a task boundary or during a period of low interaction.

The rest of this paper is organized as follows. In section 2 we explain the rationale for the experiment and define our

experimental hypothesis in the context of related research. In Section 3 we describe the experimental setup and then report on the analysis of the recorded measurements in Section 4. In Section 5 we summarize our findings, discuss their implications, and identify several directions for future work. Finally, in Section 6 we offer some concluding remarks.

2. Rationale for the Experiment

The goal of our experiment was to quantitatively measure the disruptive effects of computer-initiated interruptions on a user's task performance for a variety of web-based tasks. In order to formulate our hypotheses for the experiment, we reviewed research from both psychology and human-machine interaction investigating the similar effects of interruptions on task performance.

Zijlstra et al [5] found that interrupting a user during a series of text editing tasks caused that user to complete the tasks *faster* than when performing the same tasks without interruption. Furthermore, the more often a user was interrupted during the text editing tasks, the faster that user completed those tasks. In contrast, Kreifeldt and McCarthy [3] found that interrupting a user while performing a series of calculator-based tasks caused that user to complete the tasks *slower* than when performing the same tasks without interruption. Clearly, the conclusions derived from these two independent studies are inconsistent and further investigation is warranted. However, it does seem intuitive that task re-orientation after an interruption would cause some performance degradation. Thus, our first hypothesis states that an interrupted task will require more time to complete than a non-interrupted task within the same task category.

H1: An interrupted task will require more time to complete than a non-interrupted task within the same task category.

Kreifeldt and McCarthy also speculated that the more memory load a task placed on a user at the time of interruption, the more time that user needed to re-orient themselves to the primary task after completing the interruption. Thus, by ordering our task categories in terms of increasing memory load, a related increase in task completion time should also be observed. However, a precise measure of memory load required by a task is beyond the scope of this work, and thus, we only consider qualitative differences among the task categories. From our experience gained in a previous experiment [1], we felt that the adding and counting tasks would require the highest memory load, the selection and registration tasks would require the lowest, and the image and reading comprehension tasks would lie somewhere in-between.

H2: A task requiring greater memory load at the time of interruption will demonstrate a greater disruptive effect in terms of task performance than a task requiring a lower memory load. Specifically, the

adding and counting tasks should be disrupted the most while registration and selection tasks should be disrupted the least.

Gillie and Broadbent [2] conducted a series of experiments investigating why some interruptions are more disruptive than others. The experiments utilized a computer game in which subjects navigated to different locations collecting specified objects along the way. In each experiment, the interruption dimensions of length, similarity, and complexity were manipulated to better understand their disruptive effects. The authors concluded that the length of an interruption is not a factor in determining whether one interruption is more disruptive than another. However, the authors did find that an interruption of higher complexity or of greater similarity to the primary task is more disruptive to task performance than an interruption of lower complexity or less similarity. Because we did not explicitly manipulate the complexity of the interruption task categories, our third hypothesis only deals with similarity.

H3: An interruption task of greater similarity to the primary task will demonstrate a larger disruptive effect in terms of task performance than an interruption task of lower similarity. Specifically, the combination of the news summary task and the reading comprehension task will demonstrate the largest disruptive effect in terms of task performance.

The user experiment designed to test these three hypotheses is described next.

3. User Experiment – Measuring the Effects of Interruptions on Task Performance

The experiment measured the effects of two categories of interruptions on task performance for a variety of web-based tasks. Within each task category, the task completion time for an interrupted task was compared against the task completion time for a non-interrupted task. The differences in performance times between the interrupted and non-interrupted tasks were then analyzed using a full-factorial ANOVA with the task and interruption category as factors.

3.1 Experimental Design

The experiment used a repeated measures, full-factorial, six-by-two design. The two factors were (1) task category and (2) interruption category. The dependent variables were the time on task (TOT) and time on interruption (TOI).

3.1.1 Task Categories

The six web-based task categories used in the experiment were:

- *Addition.* Four numbers, each consisting of four digits, were presented to the subject. The numbers were right aligned in a 4 row x 1 column table. The subject's task was to add the numbers and then enter the correct sum into a text field positioned underneath the last number.

- *Counting.* A set of 40 words was arranged in a 10 row x 4 column table and presented to the subject. The 40 words were randomly chosen from a base set of six words, i.e., each of the six words was repeated several times in the table. The subject's task was to first count the number of words in the table that correctly matched a target word chosen from among the base set and then to enter this count into a text field.
- *Image Comprehension.* A completed tournament bracket (graph) starting with eight teams was presented to the subject. The subject's task was to answer five questions regarding the outcomes of the pairings.
- *Reading Comprehension.* A short passage (~4-5 sentences) was presented to the subject. The subject's task was to read the passage and then to answer three questions regarding its content.
- *Registration.* Eight registration-style questions were presented to the subject, e.g., name, age range, and political affiliation. The subject's task was to enter the requested information using one of three defined interaction formats; toggle buttons, drop-down list, or free-form text field.
- *Selection.* A set of 40 words (along with checkboxes) was arranged in a 10 row x 4 column table and presented to the subject. The 40 words were randomly chosen from a base set of six words, i.e., each of the six words was repeated several times in the table. The subject's task was to select each word in the table that correctly matched a target word chosen from among the base set.

The task categories were designed to be of varying complexity and duration (~15-40s). Because a subject would need to perform more than one task from each category, multiple sets of similar tasks were designed. The tasks were implemented using HTML.

3.1.2 Interruption Task Categories

The two interruption task categories used in the experiment were:

- *Reading comprehension.* A short (3-5 sentence) news



Figure 2. An example of a news interruption task.

summary was presented to the subject. The subject's task was to first read the summary and then to select the most appropriate title for it from among three choices. Each news summary and its actual title were obtained from an existing news site to enhance realism. An example news summary task is shown in Figure 1.

- *Stock decision.* A brief stock scenario was presented to the subject. The scenario was comprised of a fictitious company's name and the quantity, date, and price of shares previously purchased of that company. In addition, the current stock price and a one sentence "news-flash" regarding the company's financial status were presented. The subject's task was to first read and analyze the scenario and then to select one of five actions; *do nothing*, *buy a few more shares*, *buy a lot more shares*, *sell a few shares*, or *sell all the shares*. An example stock decision task is shown in Figure 2.

The interruption tasks were designed to last approximately 10-30s. Because a subject would receive more than one interruption from each category, multiple sets of similar interruption tasks were designed. The interruption tasks were also implemented using HTML.

3.2 Subjects

25 subjects (15 male, 10 female) were recruited to participate in the experiment. Subjects were a mix of undergraduates, graduate students, and local professionals between the ages of 18 and 40 and had at least one year of computer experience. Each subject was compensated for their participation with a five dollar lunch coupon.

3.3 Hardware/Software

The experiment was conducted on a Pentium III 460 MHz machine with 128MB of RAM running Windows NT. Both the primary and interruption tasks were designed using HTML. JavaScript 1.2 was used to implement the dynamics of the experiment, e.g., randomizing, sequencing, and displaying the primary and interruption tasks, and logging the performance measurements. Netscape Navigator 4.7 was used to execute the experiment. Each subject's screen interaction was recorded for later analysis using Lotus ScreenCam.



Figure 1. An example of a stock interruption task.

3.4 Experimental Procedure

Each subject completed 18 timed tasks from the six different task categories (3 tasks per category). Prior to starting each task category, the experimenter gave a verbal description of the category and the subject's task, allowed the subject to perform a practice task, and answered any questions. Subjects were instructed to complete each task *as quickly as possible while maintaining accuracy*. After questions were answered, the experimenter left the testing area and the subject performed three timed tasks from the current category. One of the three tasks was interrupted using the news interruption, another with the stock interruption, and the remaining task was not interrupted and served as a control task. Each interruption task was presented approximately halfway through the primary task. This same process was followed for the remaining five task categories. The presentation order of the task categories, tasks within each category, and interruption tasks was counterbalanced across all subjects. The experiment lasted no more than 45 minutes for each subject.

3.5 Measurements

During the experiment, the task and interruption data was logged to a file for subsequent analysis. The recorded measurements were:

- *Time on Task (TOT)*. The amount of time spent performing the primary task. This measurement did not include the time spent on an interruption task.
- *Time on Interruption (TOI)*. The amount of time spent on an interruption task.

Along with these measurements, the task and interruption categories were also logged to the data file.

4. Analysis

Before analyzing the recorded data, we reviewed each subject's screen interaction to ensure that subject performed the tasks correctly (without substantial error). Measurements from a task performed incorrectly were excluded from the analysis. For each subject, the differences in TOT between an interrupted and non-interrupted task were calculated. These differences as well as the TOI measurements were analyzed using a full-factorial ANOVA with task and interruption category as factors. The TOT differences and the TOI measurements have been graphed as a function of both task and interruption category in Figures 3 and 5. Figure 4 shows the TOT differences as a percentage increase relative to the control (non-interrupted) tasks. Detailed analysis of the results are structured in terms of the initial hypotheses.

H1: Task completion times for interrupted vs. non-interrupted tasks

The TOT differences between the interrupted and non-interrupted tasks are shown in Figure 3. Mainly consistent with our first hypothesis, a subject required more time to complete an interrupted task than a non-interrupted task for all but the registration category (Adding, $t=6.3377$, $p<.05$;

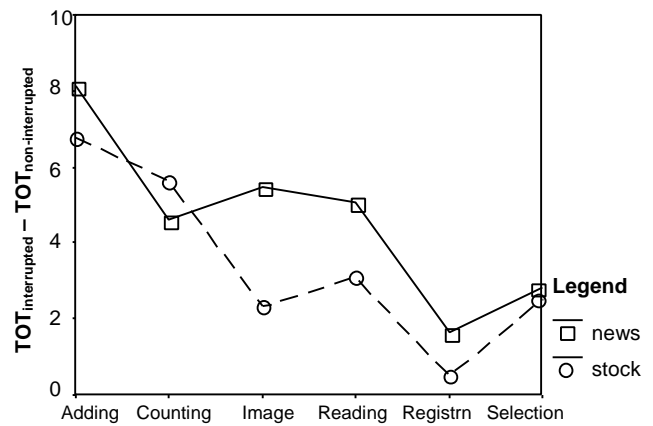


Figure 3. The differences in task completion time (TOT) for interrupted vs. non-interrupted tasks. The differences are displayed as a function of both task and interruption category.

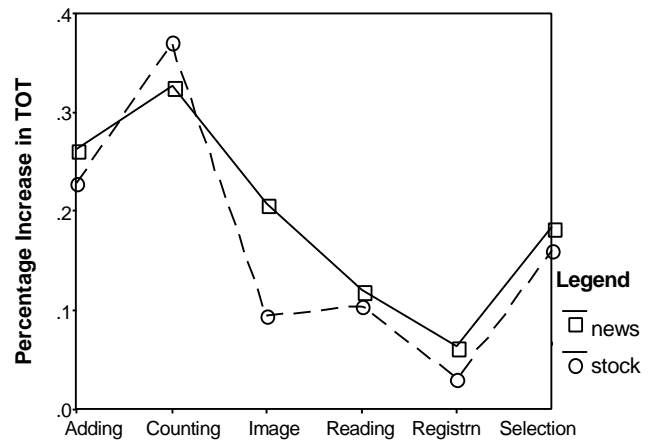


Figure 4. The relative increase in task completion time (TOT) for interrupted vs. non-interrupted tasks.

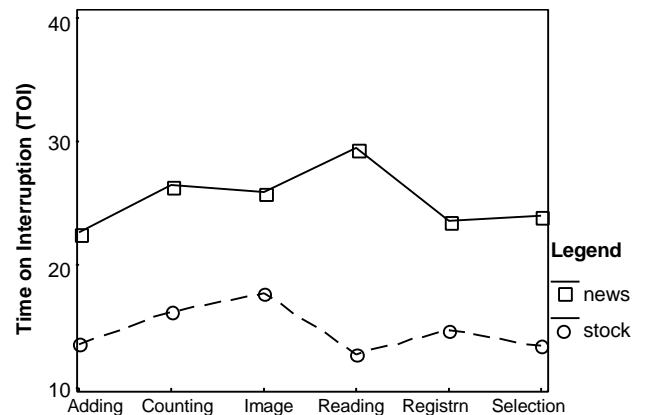


Figure 5. Time on Interruption (TOI) as a function of both task and interruption category. TOI remained relatively stable for both interruption categories regardless of the task category interrupted.

Counting, $t=8.890$, $p<.05$; Image, $t=3.431$, $p<.05$; Reading, $t=2.556$, $p<.05$; Selection, $t= 8.417$, $p<.05$; Registration, $t=1.249$, $p<.11$). Although the average difference for the registration task category did not reach a significant level, the difference was in the positive direction. Furthermore, this result seems reasonable as the registration tasks ostensibly required the lowest memory load at the point of interruption. After finishing an interruption task, a subject only had to reorient themselves to the previously suspended question. Further discussion of memory load will be saved for the results of the second hypothesis.

A main effect of interruption category was not present in the data nor were there any significant interactions. These results suggest that the memory load of a task at the time of interruption does indeed contribute to that interruption's disruptive effect on task performance.

H2: The effect of memory load on task performance

Our second hypothesis was only partially confirmed. From the graph of Figure 3, it is clear that interrupting some tasks is more disruptive than interrupting others. Analysis of the data confirms this observation. A main effect of task category was present in the TOT difference data ($F(5,260)=4.078$, $p<.05$). In addition, the amount of disruption does seem to have a positive relationship with the memory load of a task at the time of interruption. Pairwise differences among the task categories partially confirms this positive relationship. Adding had a significantly larger (at the .05 level of confidence) disruption due to an interruption than all other task categories except for counting. Counting and reading both had a significantly larger disruption than registration. No other significant differences were found in the data.

Figure 4 also demonstrates the disruptive effect of an interruption. A subject spent between 5% and 40% longer on an interrupted than a non-interrupted task. Again, a task having a higher memory load at the time of interruption demonstrated a relative increase in task completion time greater than that for a task having a smaller memory load.

Somewhat surprising was the large disruption an interruption caused for the selection tasks. From the recorded screen interactions, subjects visibly had trouble recalling the spatial position at which they previously suspended a selection task. Because no visual context cue indicated the last item compared, a subject often backtracked to the beginning of the last column with a selected item and then continued the selection task from there. As a result, the memory load required by a selection task was more fragile than we had originally anticipated.

H3: The effect of similarity

There was no main effect of interruption category in the TOT difference data nor was there any significant interactions. Thus, our third hypothesis was not confirmed. An interruption of higher similarity to the primary task was not found to be any more disruptive than an interruption of

lower similarity, which contradicts the findings of Gillie and Broadbent [3].

However, our results reaffirmed that the length of an interruption does not contribute to the disruptive effects of an interruption. Figure 5 shows that the news interruption task required more time to complete than the stock interruption task. Analysis of the TOI data confirms this observation ($F(1,260)=77.164$, $p<.05$). However, both interruption categories demonstrated a similar disruptive effect on task performance across all task categories as shown in Figure 3.

5. DISCUSSION AND FUTURE WORK

The results of our experiment demonstrate that task re-orientation after a computer-initiated interruption causes a significant increase in task completion time for a variety of web-based tasks. A computer-initiated interruption disrupts task performance similar to an external interruption such as writing multiplication tables [3]. Thus, a computer application should avoid interrupting the user's current task whenever possible.

However, this result contradicts a conclusion of Zijlstra et al [5] who found that a subject completed a task faster when that task was interrupted. The contradictory results may be due to differences in experimental design. In [5], a subject was not instructed to perform the tasks as quickly as possible and was allowed to work at a convenient pace, whereas in our experiment, a subject was explicitly instructed to work as quickly as possible while maintaining accuracy on the tasks.

A second finding of our experiment was that an increase in task completion time seems to have a positive relationship with the memory load of a task at the time of interruption. Again, this finding is inline with that suggested in [3], but contradicts the findings discussed in [2]. However in accordance with [2], we found that the additional length of an interruption task had no additional disruptive effect on task performance. Finally, we did not find a significant interaction between an interruption and a highly similar task, e.g., between the news interruption and the reading comprehension task. This result also contradicts a finding of [2]. In sum, our experiment shows that the disruptive effect of an interruption on task performance depends on the memory load of a task at the time of interruption.

The design of this experiment utilized a general set of web-based task categories as opposed to a more specific set of web-based tasks such as searching for a specific piece of information or purchasing a specific item on an e-commerce site. We chose a more general as opposed to a more specific set of task categories because we felt that the latter may not generalize to other tasks and that most real-world tasks use some combination of the more general categories used in this experiment.

The major implication of our work is that a computer application must be courteous to the user's current task. In other words, if an application needs to gain the user's

attention, then it should not immediately interrupt the user's current task. Rather, an application should wait for an opportune moment such as when the user reaches a task boundary or during a period of low interaction. As this work demonstrates, interrupting the user at an inopportune time causes a significant decrease in task performance. Waiting for a more opportune moment before immediately interrupting someone is a common practice in human-human interaction, and as this work demonstrates, manifesting this same behavior among computer applications allows users to complete their tasks faster.

Designing and building systems that better coordinate the asynchronous interactions between the user and various computer applications is an area of research that deserves further attention. Specifically, we see the following opportunities for future work:

- Building a system that recognizes some basic set of tasks performed by a user. As this work demonstrates, the higher memory load a task places on the user, the more disruptive an interruption is. Thus, a system that recognizes some basic set of tasks would be useful in determining the best task or time at which to interrupt the user.
- Building a system that observes or predicts when the user is switching between tasks. The goal would be for an application to gain the user's attention after the current task is completed but before a new task is initiated. Whenever a task boundary is observed, the system would notify the next application waiting to gain the user's attention.
- Integrating other parameters of an interruption such as urgency or task relevance into the calculation of an "opportune moment."
- Extending our experiment to explore the effects of interrupting users while viewing video or listening to audio segments. As audio and video become more pervasive throughout the WWW, understanding the disruptive effects of interrupting users interacting with these media will be essential.

6. CONCLUSION

Interruptions are a part of our everyday experience and are increasingly becoming a part of our experience interacting with computers. Computer applications such as interface agents, softbots, and push-based information windows are increasingly competing for the user's attention. As competition for the user's attention increases, the number of interruptions a user has to deal with on a daily basis will also increase. In this paper we provided a quantitative justification as to why interrupting a user in the midst of a task should be avoided whenever possible. The experimental results confirmed that a computer-initiated interruption has a disruptive effect on the user's task performance, and that this disruptive effect has a positive relationship to the memory load required by the task at the time of interruption. As a result, we must build systems that

are capable of recognizing or predicting opportune moments for gaining the user's attention such as when the user reaches a task boundary or during a period of low interaction. Once an opportune moment has been recognized, the next requesting computer application can be notified. Although a great deal of further work is needed to develop a system with these capabilities, as this work demonstrates, this system has value for the user. Furthermore, our work demonstrates that when a computer application behaves in a more socially responsible manner regarding task interruptions, users can perform their tasks more efficiently.

7. REFERENCES

- [1] Bailey, B.P., J.A. Konstan, and J.V. Carlis. Adjusting Windows: Balancing Information Awareness with Intrusion. *Proceedings of the 6th Conference on Human Factors and the Web*, 2000.
- [2] Gillie, T. and D. Broadbent. What Makes Interruptions Disruptive? A Study of Length, Similarity, and Complexity. *Psychological Research*, 50 (1989): 243-250.
- [3] Kreifeldt, J.G. and M.E. McCarthy. Interruption as a Test of the User-computer Interface. *Proceedings of the 17th Annual Conference on Manual Control*. JPL Publication 81-95, pp. 655-667, Jet Propulsion Laboratory, California Institute of Technology.
- [4] Maes, P. Agents that Reduce Work and Information Overload. *Communications of the ACM* 37, 7 (July 1994), pp. 30-40.
- [5] Zijlstra, F.R.H., R.A. Roe, A.B. Leonora, and I. Krediet. Temporal Factors in Mental Work: Effects of Interrupted Activities. *Journal of Occupational and Organizational Psychology*, 72 (1999): 163-185.