Locked-out: Investigating the Effectiveness of System Lockouts to Reduce Errors in Routine Tasks

Jonathan Back
UCL Interaction Centre
University College London
London WC1E 6BT, UK
j.back@ucl.ac.uk

Duncan P. Brumby
UCL Interaction Centre
University College London
London WC1E 6BT, UK
Brumby@cs.ucl.ac.uk

Anna L. Cox
UCL Interaction Centre
University College London
London WC1E 6BT, UK
anna.cox@ucl.ac.uk

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Abstract

While frustrating and innocuous in many settings, errors can have disastrous consequences for the use of safety critical systems and medical devices. This work-in-progress investigates the effectiveness of an enforced lockout period for reducing errors in a routine task. During the lockout period the user can look at, but not interact with the device interface for a period of 10 seconds before they resume the task after an interruption. Results show that this lockout period can reduce sequence errors by up to 64%. Identifying ways to reduce the disruptiveness of interruptions is important for HCI research given that many devices are now used in settings where interruptions are commonplace.

Keywords

Human error, interruptions, lockout, resumption delay.

ACM Classification Keywords

H5.m. Information interfaces and presentation.

General Terms

Human Factors.

Introduction

When a nurse programs an infusion pump, he will, by no fault of his own, on a very rare occasion, complete the steps of the task in the wrong order. Such sequence errors can have disastrous consequences. What makes these errors particularly irksome is that improved training cannot reduce them; in fact, these errors tend to be made by users who have performed the same task countless times as part of their daily routine. Interventions that can effectively reduce sequence errors would therefore be of great value.

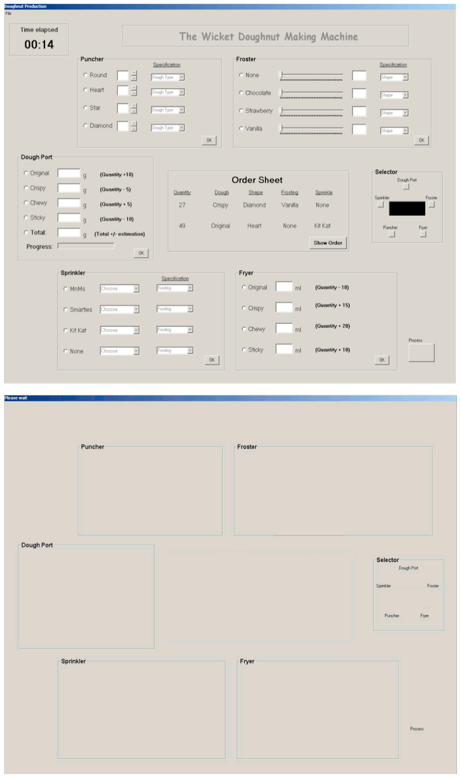
Sequence errors have been extensively studied in the laboratory using various game-like procedural tasks [2,5]. For instance, Li et al. [2] had participants learn how to use a simulated machine to make a specific quantity and type of doughnuts. Participants were occasionally interrupted while working on this Doughnut Task to perform a secondary task. Li et al. found that interruptions significantly increased the likelihood that participants resumed the task at the wrong point in the procedure.

Trafton, Altmann and Ratwani [5], using a similar paradigm to Li et al. [2], looked at the distribution of errors following an interruption. Participants in the study were interrupted in between the completion of one subtask and the commencement of the next. Trafton et al. found that errors were in close proximity to the correct action in the task sequence (i.e., perseveration errors and anticipation errors were the most common sequence errors).

Trafton et al. [5] explain their findings using the memory for goals model [1]. Memory for goals assumes that sequence errors are caused by a failure to recall the correct memory trace that represents where in the task the user was prior to the interruption. In other words, errors are caused by memory retrieval failures. It is well known that memory retrieval is sensitive to changes in speed-accuracy tradeoff criterion. For instance, Reed [4] has shown that recognition accuracy increases as the time allowed for retrieval increases. In this paper, we investigate the potential value of encouraging people to spend more time trying to recall where they were in a routine task prior to resuming it after an interruption.

We use an enforced lockout procedure to make users spend time thinking about where they were before being interrupted. O’Hara and Payne [3] have previously demonstrated that an enforced lockout period can encourage users to plan more of their actions performing an interactive task so as to be more efficient. O’Hara and Payne had users edit a word processing document. In one condition each action was associated with a 7-second system lockout. While users took longer to complete the task with the lockout, they completed the task with fewer actions. This was because users planned their next action during the lockout period, and so made every action count.

In this paper, we evaluate the effectiveness of an enforced lockout period for reducing errors. We use the same general task paradigm as that used by Li et al. [2] and Trafton et al. [5], and introduce a lockout condition. During the lockout period users can look at, but not interact with the interface for a period of 10 seconds before resuming the task after an interruption. We expect that users will make effective use of this lockout period to recall where they were in the task structure and as a result make fewer errors.



Order Sheet

Quantity	Dough	Shape	Frosting	Sprinkle
27	Crispy	Diamond	Vanilla	None
49	Original	Heart	None	Kit Kat

Show Order

Figure 1. The Doughnut task interface (top) and the lockout interface (bottom).

Method

Participants. Twenty-four participants (13 female) were recruited from the UCL Psychology Subject Pool, with a mean age of 23.3 years (*SD* = 4.8). The experiment took approximately 55 minutes to complete and participants were paid £7 (approx. $11) for their time.

Tasks. A procedural task was used that required participants to program a fictitious doughnut-making machine to meet an order for a specific quantity and type of doughnuts. While working on the Doughnut Task, participants were occasionally interrupted and required to complete a secondary mental arithmetic task, called the Packing Task.

Figure 1 (top) shows a screen shot of the Doughnut task interface. To produce the doughnuts for a specific order, participants have to operate five different compartments in the machine in a specific sequence: (1) Dough Port, (2) Puncher, (3) Froster, (4) Sprinkler, and (5) Fryer. Each compartment needs to be activated, using the "Selector" on the right before any parameters can be entered. After entering all the specific parameter values from a separate Order Sheet participants clicked on an OK button. This caused the parameter information to be erased from the interface. This was done to ensure that there were no visual cues in the task environment that could be used to determine which component was being worked on. After values were entered into all five components, participants clicked on a "Process" button to obtain a report of what doughnuts were made.

The Doughnut Task is a procedural task that can be decomposed into discrete subtasks. The secondary Packing Task was used to interrupt participants in between subtasks of the Doughnut Task procedure.

The Packing Task required participants to perform mental arithmetic to work out how to pack a number of doughnuts into boxes that varied in their capacity. For example, 14 Doughnuts could be packed in 3 boxes that hold 4 Doughnuts each, and 1 box that holds 2 Doughnuts (i.e., (3x4)+(1x2)=14). Table 1 summarizes the order of steps required to produce doughnuts for a specific order, along with possible interruption points.

Design. A within-subjects design was used with three different conditions (no-interruption, interruption-only, and interruption+lockout). In the no-interruption condition participants completed a specified doughnut order without being interrupted by the Packing Task. This condition provides a measure of baseline performance on the primary Doughnut Task.

In the interruption-only condition participants were interrupted while performing the primary Doughnut Task and were required to complete the secondary Packing task. Two interruptions occurred randomly at one of the four interruption points (see Table 1).

In the interruption+lockout condition participants were again interrupted twice. But on completion of the secondary Packing Task, participants were inhibited from resuming the primary Doughnut Task for a 10-second lockout period. During this lockout period, participants could view a surrogate interface that showed the interface but with all task-specific information removed (see Figure 1, bottom). This lockout occurred following both of the interruptions for a specific interruption+lockout trial.

Procedure
Select Dough Port, Enter Parameters, Click OK
Select Puncher, Enter Parameters, Click OK
Possible Interruption Point
Select Froster, Enter Parameters, Click OK
Possible Interruption Point
Select Sprinkler, Enter Parameters, Click OK
Possible Interruption Point
Select Fryer, Enter Parameters, Click OK
Possible Interruption Point
Click Process

Table 1. Descriptions of the Steps in the Doughnut Task and the interruption positions. On interruption trials, interruptions occur at 2 out of the 4 possible interruption points.

If a participant resumed the task in the wrong place after an interruption (i.e., repeating a step or skipping over a required step), they were unable to continue the task until the correct step was performed. In terms of dependent variables, we were primarily interested in the frequency of sequence errors made in each condition. In addition, we also calculated the time taken to resume the Doughnut Task following an interruption in the interruption-only condition.

Procedure. Participants were given instructions on how to perform each task and conducted at least two practice trials on the primary Doughnut Task. To guarantee that participants knew how to perform the Doughnut task correctly before moving onto the main experimental trials, participants continued to practice the Doughnut Task until they could complete at least one error-free trial.

Participants were informed that they would occasionally be interrupted while working on the Doughnut Task and that they would be required to perform the secondary Packing Task. They were instructed on how to perform the Packing Task and given the opportunity to practice the task. Participants were also told that on some trials they would be locked-out from the Doughnut Task after completing the Packing Task. They were told that this lockout period would last for 10-seconds and that during this time they could look at, but not interact with the task interface. Participants were told that they might use this lockout period to reflect on the steps they had already done or retrace their steps.

After training, participants completed 12 trials, which were divided into four blocks of three trials each (one trial per condition). Within each block the order of trials per condition was randomized. After performing half of the experimental blocks, participants were given the opportunity to take a short break.

Results

We were primarily interested in the frequency of sequence errors made in each condition. An error is defined as any action where the user worked on a subtask at the incorrect point in the task sequence. We were particularly interested in comparing the error-rates between the different conditions at the possible interruption points in the task (see Table 1).

Figure 2 shows the error-rate across the different conditions demonstrating that participants made very few sequence errors when there were no-interruptions (only 18 errors were observed across 768 opportunities). In the interruption-only condition the error-rate increased dramatically (42 errors were observed over 192 opportunities). But introducing an enforced 10-second system lockout following the interruption substantially reduced the error-rate (only 15 errors were observed across 192 opportunities). A one-way ANOVA found a significant effect of condition on error-rate, $F(2,22)=29.60$, $p<.001$. A post-hoc comparison revealed that the error-rate for the interruption+lockout condition was less than that in the interruption-only condition, $t(22)=5.326$, $p<0.01$. This suggests that the enforced lockout reduced the effect of being interrupted on error-rate. However, the error-rate in the interruption+lockout condition was nonetheless above the 5% threshold and therefore is still considered to be systematic.

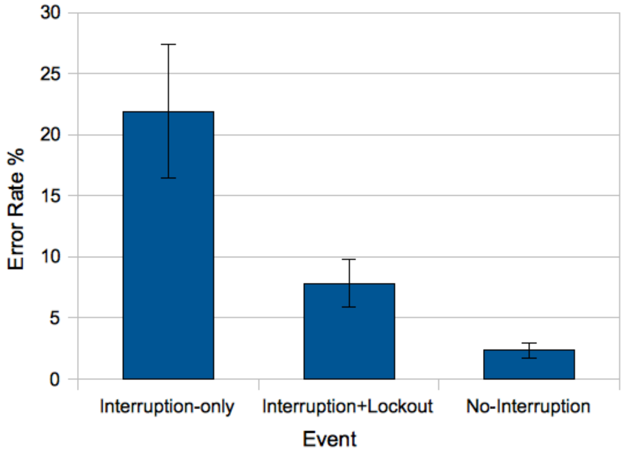


Figure 2. Error-rates across different conditions.

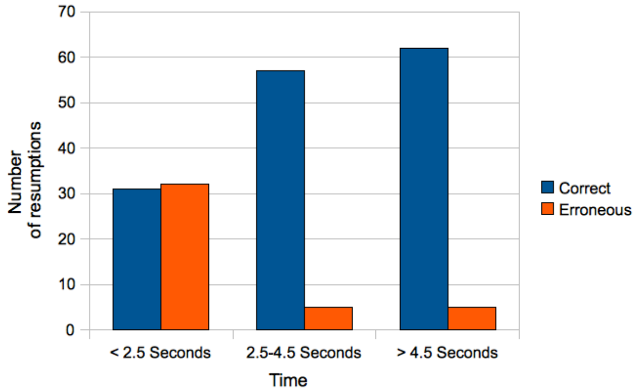


Figure 3. Number of resumptions in the interruption-only condition.

The error-rate analysis shows that an enforced lockout period can substantially reduce the likelihood of an error being made following an interruption. Given that most errors were made in the interruption-only condition, we consider how quickly participants chose to resume the task following an interruption on these trials. In particular, we are interested in whether participants were more, or less, likely to make an error if they took more time before resuming the primary task.

Figure 3 shows the distribution of resumption times following an interruption. The data have been separated by whether a correct or an erroneous action was performed. It can be seen that at very short resumption times (less than 2.5s), the likelihood of an error being made is at chance. With increasing resumption time (greater than 2.5s), participants are more likely to perform the correct action and are unlikely to make an error. Statistical analysis shows that resumptions times were indeed much shorter on average when participants made an error (*M*=2.77s, *SD*=1.38s) than when the correct action was executed (*M*=4.09s, *SD*=1.59s), *F*(1,22)=29.60, *p*<.001.

Discussion

Interruptions are disruptive to many, and in some cases can cause users to make an error by repeating or skipping over important task steps. The results of this work-in-progress suggest that some of the disruptive effects of interruptions can be mitigated – though not eliminated – by an enforced lockout period following an interruption. This lockout period gave participants time to reflect on where they were in the task sequence before the interruption. We found that prohibiting users from resuming a task following an interruption for a period of 10 seconds reduced sequence errors by 64%.

Our data also show that in the interruption-only condition, where there was no enforced lockout, fewer errors were made when participants took more time before resuming the primary task. Taken together these data suggest that sequence errors are sensitive to changes in people's speed-accuracy tradeoff criterion. That is, those who 'dive in' without retracing their steps are more prone to error.

While speed-accuracy tradeoffs have been frequently observed in many contexts (e.g., [4]), it is not immediately clear to us how memory for goals [1,5] would explain these data. One possibility is that with greater time, multiple attempts could be made to retrieve the correct step from memory. But given that memory retrieval is assumed to be a stochastic process simply increasing the frequency of retrieval attempts should not necessarily lead to an increase in the likelihood that the correct memory is retrieved. Moreover, the theory assumes that the activation of a memory generally decays over time, making it less likely that it is later recalled. In contrast, our data show that with increased resumption delay there is a lower chance of error (i.e., people are better able to recall where in the task they were). It is an open question whether memory for goals can explain these data.

In summary, this work-in-progress demonstrates the potential value of encouraging users to stop and pause before resuming a task following an interruption. This is an interesting departure from previous work that has tended to focus on minimizing resumption lags. We show instead that, in some cases, longer resumption lags can be beneficial because they reduce the likelihood of errors being made. Identifying ways to reduce errors caused by interruptions is important for HCI research given that many devices are now used in settings where interruptions are commonplace. Future work will investigate how users take advantage of lockouts and what triggers self-imposed resumption delays, while situated studies will investigate how people use programmable devices in medical contexts.

Acknowledgements

This work was funded by EPSRC Grants EP/G059063/1 & EP/G043507/1. We thank Richard Young, Ann Blandford, Paul Curzon & the UCLIC Cognitive Science Reading Group for useful discussions on this work. We also wish to thank Simon Li for use of the doughnut-task.

References

[1] Altmann, E.M. & Trafton, J.G. (2002). Memory for goals: An activation-based model. *Cognitive Science*, *26,* 39-83.

[2] Li, S.Y.W., Blandford, A., Cairns, P., & Young, R.M. (2008). The effect of interruptions on postcompletion and other procedural errors: An account based on the activation-based goal memory model. *Journal of Experimental Psychology: Applied, 14*, 314-328.

[3] O'Hara, K.P. & Payne, S.J. (1999). Planning and the User Interface: The Effects of Lockout time and Error Recovery Cost. *International Journal of Human-Computer Studies, 50*, 41-59.

[4] Reed, A.V. (1973). Speed-accuracy trade-off in recognition memory. *Science, 181*, 574-576

[5] Trafton, J.G., Altmann, E.M., & Ratwani, R.M. (2009). A memory for goals model o sequence errors. In Proc. *9th International Conference of Cognitive Modeling* (paper 225), Manchester, UK.