Now Check Your Input

Gould, Sandy; Cox, Anna L.; Brumby, Duncan P.; Wickersham, Alice

DOI:
10.1145/2858036.2858067

Now Check Your Input: Brief Task Lockouts Encourage Checking, Longer Lockouts Encourage Task Switching

Sandy J. J. Gould, Anna L. Cox, Duncan P. Brumby, Alice Wickersham
UCL Interaction Centre
University College London, WC1E 6BT, UK
{s.gould,brumby}@cs.ucl.ac.uk, {anna.cox,alice.wickerhsam.11}@ucl.ac.uk

ABSTRACT
Data-entry is a common activity that is usually performed accurately. When errors do occur though, people are poor at spotting them even if they are told to check their input. We considered whether making people pause for a brief moment before confirming their input would make them more likely to check it. We ran a lab experiment to test this idea. We found that task lockouts encouraged checking. Longer lockout durations made checking more likely. We ran a second experiment on a crowdsourcing platform to find out whether lockouts would still be effective in a less controlled setting. We discovered that longer lockouts induced workers to switch to other activities. This made the lockouts less effective. To be useful in practice, the duration of lockouts needs to be carefully calibrated. If lockouts are too brief they will not encourage checking. If they are too long they will induce switching.

Author Keywords
Lockouts; number entry; checking; interruptions; multitasking; crowdsourcing; naturalistic experimentation;

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Almost all computing tasks require people to enter data of one kind or another. For instance, nurses program infusion pumps to administer set doses of drugs; bankers enter amounts to be transferred between accounts; business people book flights for specific dates and times. Other examples abound. In these scenarios, mortality, profitability and punctuality are contingent on accuracy. Ensuring that data has been correctly entered is important. Nurses should check that the programmed rate of infusion matches the prescribed rate of infusion. Bankers should double check that the amount of money transferred is the amount intended. Business people should make certain before they book that the flight they have chosen is the one that they wanted.

CHI'16, May 07 - 12, 2016, San Jose, CA, USA

ACM 978-1-4503-3362-7/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858067

Alas, there are often reports of serious data-entry errors occurring because input has not been properly checked. Arriving the day before a meeting might be mildly irritating, but in many scenarios the consequences of not checking can be catastrophic and irreversible. In one example, a Canadian cancer patient died after an infusion device was programmed to deliver a day's worth of chemotherapy in one hour [51]. In another example, a New York Stock Exchange clerk bankrupted a firm by transferring 11 million shares (worth $500m) instead of the $11m worth of stock that they had intended [47].

To err is human. A major challenge for the HCI community is to design systems and devices that limit the likelihood of errors occurring in the first place. How can people be encouraged to check their input for errors? A potential solution comes from a domain with a long history of fatal and highly visible errors – railways (see [48]).

Welwyn Garden City, about twenty miles from London's King's Cross station, was the site of a major rail crash in 1935. The accident, which was later blamed on an inexperienced signalman, resulted in fundamental changes to railway signaling technology and practice in the United Kingdom. One of the safety-enhancing technologies introduced after the accident was the *Welwyn Winder*. This was a lockout device. The Winder contained a screw-activated switch that released signaling equipment. To activate the switch, a signalman had to rotate – wind – a handle for a couple of minutes (see Figure 1). The idea was that dangerous signaling maneuvers would be avoided by the use of the Winder; as a signalman wound the screw, he would have plenty of time to contemplate the action he was about to perform. Dangerous operations would be detected and aborted, provided, of course, that the signalman was not busy reading a newspaper or chatting to colleagues as he wound.

Welwyn Winders are no longer in use, but analogous techniques are employed in modern digital interfaces. For instance, Mozilla's Firefox browser makes its users wait for a moment before installing potentially nefarious extensions. Research has shown that enforcing these kinds of lockouts can improve performance in data-entry tasks [39] and reduce error rates [7] by forcing people to stop and think before they take action. Lockouts have been used successfully in safety-critical settings [22] and offer an alternative to easily-skipped

and often ignored modal dialogs (e.g., 'Are you *sure* you want to install this malware?') [14,53].

Although there are indications that lockouts may be useful in certain scenarios, previous work provides little guidance on, for example, how long a lockout must be in order to be effective. Perhaps more importantly, prior work gives us little indication of whether lockouts are effective in practice or simply encourage people to attend to other tasks while they are locked-out (see [7] for a detailed discussion).

The deficiencies in our understanding of computer-based lockouts raise questions that can be neatly framed by considering the operation of the Welwyn Winder: first, how did it come to be that the handle had to be wound for two minutes and not one or three? Second, did requiring the signalman to wind for two minutes and not one influence whether he read a newspaper as he wound? Finally, did interleaving reading and winding make him less likely to spot dangerous maneuvers? We address these questions in the timely context of computer-based data-entry.

We report two experimental investigations of the effect of lockout duration on checking performance. We examine whether lockouts induce switching behavior and whether such behavior is deleterious to checking performance. The first study, a lab experiment, investigates the relationship between lockout duration and error detection in a routine number-entry task. The results demonstrate that longer lockouts yield improved error detection. The second study, an online crowdsourced experiment, finds that in less controlled environments the effectiveness of lockouts in encouraging checking is compromised by their tendency to induce people to switch to other activities.

Related work

Many activities, particularly in safety-critical environments, do not have an undo function. Once an overdose has been delivered doctors cannot simply perform a rollback and have their patient return to a previous state. The irreversible nature of many real world errors has made understanding and preventing them a priority for researchers. This paper focuses on a particular class of errors, those made when entering numbers into systems and devices. These errors occur in safety-critical domains including healthcare [52] and finance [41] but also crop-up in more prosaic tasks like transcribing receipts for expense claims (see, e.g., [25]).

Removing people from data-entry tasks might help eliminate errors in some scenarios. It can also just move problems elsewhere. A study of a hospital ward where barcode readers had 'replaced' human data-entry found that 31 workarounds appeared in response to the introduction of barcode readers. The workarounds were developed to accommodate emergencies, to cope with missing and damaged labels, and to use products without barcodes [31]. Human intervention is still necessary for many data-entry tasks. Because of this, there has been a concerted effort to develop interfaces that make transcription errors less likely. Approaches have included redesigning interfaces to influence the likelihood of particular types of error occurring [40] and developing evidence-based taxonomies to help designers understand the kinds of errors that people will make [56].

Rather than simply focusing on reducing the likelihood of an error being made, researchers have also tried to increase the likelihood of errors being spotted. Wiseman et al. [57] experimented with inserting an additional check stage in a number-entry task. Their results showed that this step did not reduce error rates. This was not a surprising result; without a forcing function, why would people make the effort to check? In this paper we investigate whether requiring people to pause for thought makes them likelier to check their work.

One way of getting people to pause for thought is to give them no other option. O'Hara and Payne [39] tested whether preventing participants from starting a task for a short period improved performance on a problem-solving activity. They found that being locked-out for 7-s rather than 3-s meant that participants planned their actions more carefully. The result of this additional planning effort was superior performance.

Lockouts also seem to improve performance in routine procedural tasks like data-entry. Brumby et al. [7] experimented with introducing lockouts immediately after interruptions. They found that lockouts reduced the rate of errors made on resuming a task after an interruption. Additionally, post-interruption resumptions were faster if they followed a lockout. This reinforces the idea that getting people to pause for thought before acting can improve task execution performance.

Figure 1. A set of railway lockout devices. The handles on the right-hand sides are wound to set the device. © Dr J M Saxton

Why do lockouts work?

The cost of an action can be loosely divided between the cost of planning it and the cost of executing it. Careful planning is costly but may allow for an action to be executed more quickly or accurately. It has been suggested that as long as extra planning costs are expected to be offset by reductions in execution costs, people will exert effort on planning [38].

The presence of a lockout disturbs this calculus by making planning relatively less costly [39]. Normally, people will plan until the costs of extra planning do not produce worthwhile reductions in execution time. More planning *could* speed-up execution, but after a certain point it is not deemed to be worth the marginal costs. The introduction of a lockout changes how people determine what constitutes 'worthwhile' [45]. If a plan cannot be executed until after a lockout as ended, it makes sense to continue planning for the duration of a lockout: the extra planning might reduce action execution time, even if only by a small amount. Ceasing to plan before the end of a lockout and simply waiting for it to finish risks longer execution time for no benefit. So people, rationally, take the whole period of the lockout to plan their actions in expectation of shorter execution times. Empirical evidence supports this account of behavior [21,39].

If lockouts elicit additional planning because doing so reduces task execution times, are lockouts likely to be effective in encouraging checking? After all, checking occurs after an action has been planned and executed. If checking during a lockout can have no positive bearing on task execution, will people use lockouts to check even when there is no clear utility to be derived from doing so?

We hypothesize that lockouts will improve checking behavior. There is no direct trade-off with checking as there is between planning and execution. Instead, we think lockouts will improve checking because, just when people plan during lockouts, there is potentially little else to do. During a checking-phase lockout, the choice people face is to sit and do nothing or spend the time checking their input.

The critical point here is that we expect lockouts to elicit checking behavior *only while there is nothing else to do*. The theory of lockout efficacy we have outlined is predicated on people expanding their planning or checking efforts to fit the time they have been given. People plan or check because it has some marginal utility over doing nothing at all. What happens when people *do* have other things to do?

The presence of competing tasks might mean that people are able to derive more utility from a lockout period by switching to something else, rather planning or checking. People usually attempt to maximize utility in some form (although their decision making is not always optimal; see [28]). One of the factors that is likely to influence peoples' propensity to switch during lockouts is the cost of making switches. All switches have costs [4]. These costs come from re-orienting to different tasks and, if necessary, re-encoding information about the current state of a given task. The more complex a task gets, the costlier it is to switch to something else [5,50].

Whether lockouts induce people to switch to other tasks will depend on whether the perceived utility of switching exceeds the perceived costs of switching. When lockouts are very short, the costs of switching are proportionally higher: more of a lockout period is lost on switching costs. Given fixed switching costs, as lockouts get longer switching costs become proportionally smaller. We might therefore expect people to be more inclined to switch to other activities as lockouts lengthen.

Understanding the relationship between task switching and lockouts is crucial for determining whether lockouts might usefully be deployed in practice. We examine the relationship between lockouts and task-switching behavior in Experiment 2. First, though, we need to know whether lockouts are actually effective in encouraging checking behavior. We investigate this in Experiment 1.

EXPERIMENT 1

The aim of Experiment 1 is to investigate whether lockouts encourage people to check their input. Lockouts increase the effort spent on the planning of actions because a lockout of fixed duration makes extra planning a rational choice. It is not clear, however, how utility can be derived from checking.

We expect the laboratory setting of the experiment to affect behavior. In particular, we expect the presence of an experimenter and the absence of alternative activities to influence participants' propensity to check their input. We expect that participants who encounter lockouts will be more likely to check than those who do not. This is because the only alternative for participants who are locked-out is to ignore the instructions and do nothing. As lockouts lengthen, we think it is reasonable to assume that participants will be less inclined to do nothing and 'wait out' lockouts.

Note that our focus here is on checking performance rather than typing performance. People are usually accurate typists; people make errors at a rate of 0.2% when typing numbers [41]. Over a population this is a large number, but over a sample it is small. In this experiment we ensure a predictable minimum number of errors by surreptitiously modifying participants' input as they work on the task. This approach was also taken by Olsen [41], who found it to be an effective technique for dissociating typing and checking behavior. Other investigations of number-entry have made use of simulated interactions such as key bounce errors to compensate for the relative scarcity of typing errors in experimental studies [40].

Method

Participants

Thirty-three participants (25 female) with a mean age of 19 years (*SD*=1 year) took part in the study. Participants were drawn from a psychology participant pool and received course credit for taking part.

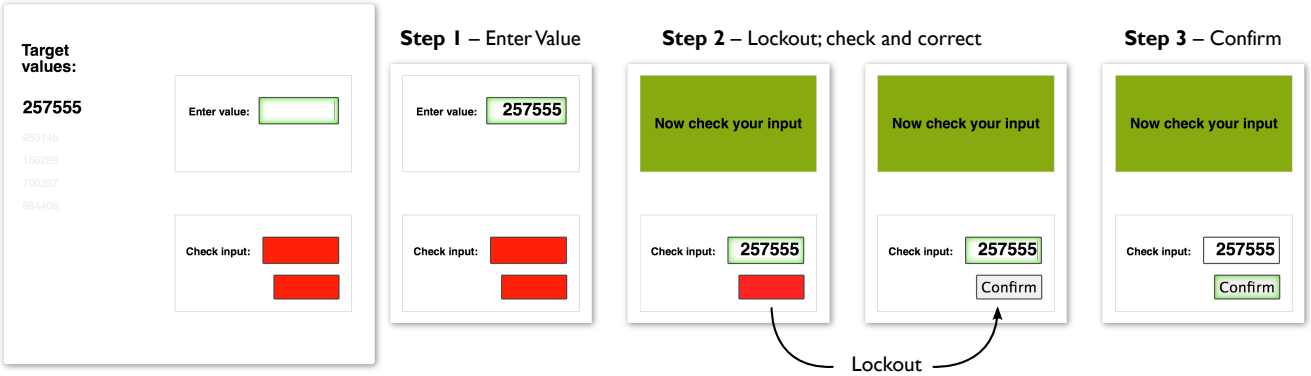

Figure 2. Participants are presented with the screen farthest left at the start of each trial. The first step is to copy the target value from the list. Then, participants check their input during a lockout period. After the lockout, participants confirm their input. The target number list is visible for the duration of trials, but is omitted from the all but the left most part of the sequence to save space.

Design

The experiment manipulated a single between-subjects factor, lockout duration. This had three levels, 0-s (i.e., no lockout), 3-s and 6-s. We measured whether participants confirmed input values that did not match the target.

Materials

A browser-based text transcription task was developed for the study. Participants were asked to copy six-digit numbers from a list on the left of the screen into a text field on the right. At the start of each trial, the input field was in focus and the check field and *Confirm* button were disabled. A standalone USB number-pad was provided for number entry.

Once participants had entered the target value into the input field, they pressed the tab key to move to the checking phase. This moved the number down the screen and into a second field. When participants moved to the checking phase, the check field was unlocked. The input field was replaced at the same moment with an instruction to check (see Figure 2).

During the checking period participants were locked-out from the task: they could not proceed because the *Confirm* button remained disabled until the end of the lockout period (see Figure 2). After participants had the opportunity to check their input, the *Confirm* button was unlocked and could be activated with the spacebar or return key. This ended one trial and simultaneously started a new one. The input field received focus and the check field and *Confirm* button were once again disabled. The previous target number disappeared and the next number in the list took its place.

The task was controlled entirely with the keyboard. Participants used the tab key to move between text boxes and buttons. The spacebar or return keys were used to activate buttons. The caret could be moved in text boxes using the cursor keys. Use of the mouse was also permitted for this purpose but it could not be used to perform any other actions: this was enforced by capturing all mouse and keyboard events in the task window. Incorrect keypresses or clicks were recorded in the background by the software but had no visible effect on the task itself.

Procedure

Participants took part in the experiment in a laboratory cubicle. After the purpose of the experiment was explained and any questions were answered, consent was obtained.

Participants first took part in a training phase, completing ten trials split over two blocks. After these trials, participants completed 48 blocks each comprising five trials for a total of 240 trials. The experiment took approximately 45 minutes. To reduce the risk of typing fatigue, a break lasting a minimum of five seconds was introduced between each block. Participants could rest for longer if they wanted.

In sixteen of the trials participants' input was modified before it was moved to the checking field (see Figure 2). In these *doctored* trials, two of the four middle values of the number were transposed. Numbers generated for these doctored trials did not have any repeated digits.

All numbers were six digits long. The notional time to check a given number was therefore consistent across trials and manipulations.

Results

General performance

Each participant completed 240 trials. Eight of these trials were *doctored* – participants' input was modified to assess checking behavior. We first consider participants' general performance in the 224 trials that were *un-doctored*.

In each trial, participants had an opportunity to check and correct any errors before they confirmed their entry. Input that was still incorrect after this point was labelled as a *confirmed error*. During un-doctored trials participants made 0-12 confirmed errors (M=2, SD=3) for a rate of 0.9%. This demonstrates a good level of participant compliance.

One participant realized on their 70th trial that they could enter the first digit of the target number, tab to the confirmation field, and enter the remaining five digits. This sort of workaround behavior is not uncommon in routine transcription tasks [11], but this deviation was sufficient to warrant their exclusion from subsequent analyses.

Effect of lockout duration on checking accuracy

We counted a trial as inaccurate if it was incorrect when the *Confirm* button was hit. If an error was spotted and corrected before the *Confirm* button was hit, the trial was counted as accurate. We initially focus on these confirmed errors, rather than transcription errors because our primary interest is checking behavior rather than typing accuracy.

We first considered error rates in the 496 doctored trials that occurred across all included participants. As this measure is not contingent on typing accuracy, if participants were checking input before confirming they should have picked up on these errors and corrected them. Participants failed to spot 0-16 of the introduced errors (*M*=7, *SD*=6; *Mdn*= 7) for a rate of 48%. Participants spotted more introduced errors in conditions where a lockout was present (see Table 1).

	0-s	3-s	6-s
Error trials	$116/_{176}$	$67/_{176}$	$55/_{160}$
Correction rate	34%	62%	65%

Table 1. Error count and correction rates for doctored trials

Aggregate errors in doctored trials are count data with a small mean. We also have a ratio data predictor so a Poisson regression model is appropriate for analysis. We tested for over-dispersion [15] but no correction was necessary. The results of the regression show that lockout duration significantly (at the .05 level) predicted error rates (R^2=.13, *b*=-0.12, *t*(30)=4.25, *p*<.001). Longer lockouts seem to improve checking performance.

Using the same kind of Poisson regression that was used for examining doctored trials we looked at confirmed error rates in the trials that were *not* doctored. The regression suggests a significant but smaller effect of lockouts on error rates (R^2=.05, *b*=-0.12, *t*(30)=2.21, *p*=.03).

Error rates in the un-doctored trials are heavily contingent on typing accuracy: participants who make more typos have more errors to catch and so are likely to have higher errors rates. Participants who are perfect typists but make no checks will still have a 0% error rate. We compensate for this by looking at how many errors participants had made after the first input phase compared to after the second check phase. In other words, we look at how many errors participants make and then how many of those errors they spot.

After the input phase participants had made 1-15 errors (*M*=6, *SD*=4, *Mdn*=5), for an error rate of 2.8%. Participants in the 6-s condition were least accurate at this point (see Table 2). After the chance to check and correct their input, participants confirmed 0-12 errors (*M*=2, *SD*=3, *Mdn*=1). After the lockout, participants in the no lockout (i.e., 0-s) condition had the highest error rate (see Table 2).

To determine whether lockouts made people more likely to correct their errors, we took a mixed effects modeling approach. We again accounted for count data and small means. Lockout duration was modeled as a fixed effect. Potential error count is indicative of individual attentiveness and was modeled as a random effect. Our outcome was confirmed (i.e., uncorrected) error count. In this way we were able to examine the effect of lockout on error rate given participants' individual propensities to make errors.

We compared our alternative model to a null model that did not include the fixed effect of lockout duration. There was strong evidence to suggest our alternative model was significantly better fit than the null model (D_{AIC}=9.95, χ^2=8.9, *p*=.002). This confirms that lockouts are effective in increasing the chances of errors being caught.

Window switching behavior

Experiment 1 was run in isolated cubicles on university-provided computers, limiting opportunities for distraction. To ensure parity with Experiment 2, we tracked window switches in Experiment 1 by monitoring whenever participants switched to another window or browser tab using JavaScript *blur* and *focus* events. There were five switches in total during the experimental trials, each switch being made by a different participant. Switches ranged from 2.5–7.6-s (*M*=5.2-s, *SD*=2.0-s). None of the switches were initiated during lockout periods.

	0-s	3-s	6-s
Potential errors	$64/_{2464}$	$59/_{2464}$	$78/_{2240}$
Errors caught	$30/_{64}$	$50/_{59}$	$60/_{78}$
Potential error rate	2.6%	2.4%	3.5%
Correction rate	51%	85%	77%
Final error rate	1.4%	0.4%	0.8%

Table 2. Correction rates after making an initial typo. Table includes un-doctored trials only.

Discussion

The results of Experiment 1 demonstrate that lockouts encourage people to check their input in routine number-entry tasks, although there are some indications that longer lockouts have diminishing marginal effectiveness.

The finding that lockouts can improve post-task checking is novel; it shows that lockouts can encourage people to add an additional checking step to the execution of a task. Prior work has only demonstrated the effectiveness of lockouts in improving the quality of pre-execution planning [3,7,39], which takes place irrespective of the presence of a lockout.

Our evidence demonstrates that implementing lockouts in number-entry tasks reduces the likelihood of errors being committed. We could, for instance, assert that lockouts should be built into online banking tools to prevent 'fat finger' errors. Or we could suggest that infusion pumps on wards should force nurses to take a moment to check their parameters before they set infusions running.

We do not make these assertions because our laboratory study does not test a critical assumption that we have made about lockouts. Our findings in Experiment 1 are contingent

on the fact that participants in the lab have little else to do but follow instructions and use the lockout period to check their input. In the absence of other tasks to perform and without other devices seeking their attention, participants can only choose to check their input or do nothing at all.

Our day-to-day lives rarely present us with such a choice. In reality we are constantly interrupted [17] while we juggle multiple tasks [35]. Safety-critical domains are not immune. Medical practitioners, for instance, are frequently interrupted as they go about their work [6,12,54]. With lockouts, the issue is whether people find other ways of using their time if they are prevented from making progress on the task they are working on. Returning to the signalman turning the Welwyn Winder: did he spend those two minutes monitoring the dial on the Winder, or did he leaf through his newspaper to make the lockout period pass more quickly?

EXPERIMENT 2

The results of Experiment 1 show that longer lockouts encourage more thorough checking behavior. In the context of utility-maximization accounts of behavior during lockout periods (e.g., [39]), this is puzzling. There was no clear motivation for participants in Experiment 1 to check their input during lockout periods, yet they did. Participants may have checked their input because they were asked to do so by the experimenter and they had nothing else to do. The behavior we saw may have simply been a demand characteristic. The same could also be true of prior investigations of lockouts.

If the results of Experiment 1 were an experimental artifact, behavior in response to lockouts might be considerably different outside the controlled setting of the laboratory. For instance, lockouts might induce people to switch to other tasks rather than encouraging checking. This would be problematic for two reasons. The first is obvious; if people are doing other tasks, they cannot be checking. The second reason relates to the effects of interruptions on performance.

Interruptions can reveal new and important information. Indeed, a good proportion of self-interruptions occur because people want to find out information that will help advance their current activity [13,26]. Dealing with interruptions comes with costs, however. Myriad laboratory studies have demonstrated that interruptions negatively affect performance [24,37,42]. Even very short interruptions have measurably deleterious effects [1]. These negative effects are particularly pronounced for activities – like checking – that occur as post-completion steps after the main goal of a task has been completed [8,33,34,47].

To understand whether lockouts induce self-initiated task switching, we leveraged the crowd in a naturalistic investigation of lockouts. Taking an in-the-wild approach, we were able to test whether lockouts improve checking performance or just induce people to switch to other activities. We already know that online participants frequently switch to other activities, sometimes for prolonged periods [19]. In this study, rather than attempting to control self-initiated task-switching behavior and its effects on performance, multitasking is anticipated and incorporated into the design of the experiment. Specifically, we look for evidence that lockouts induce participants to switch to other activities.

In Experiment 1 we tested three lockout durations. In this experiment we use a continuous design with 101 lockout durations. We made this change because a continuous design offers the potential to more accurately map the relationship between lockout duration and task-switching. Switching behavior is determined, in part, by the costs of switching. These costs are influenced by the demands of a task. Without a clear idea of the switching costs associated with our task and how they might interact with lockout duration to influence switching frequency, it makes sense to sample a wide range of lockout durations. Is the relationship between lockout duration and switch frequency linear? Is there a lockout period that does not induce switches? A continuous design helps us to answer these questions.

Experiment 2 also uses a different sample population to Experiment 1 (which made use of a university subject pool). In Experiment 2 we recruited Amazon Mechanical Turk (AMT) workers. This was primarily a practical decision to increase sample size for the continuous design. The AMT sample has some biases, for example it is better educated and younger than average [43], but these are also the characteristics of an average psychology subject pool. Any differences in the populations are not apparent in the data produced. A number of human performance studies have found that data from AMT workers is comparable to that collected through traditional subject pools (see, e.g., [16,23,30,36,44]). We can be confident, therefore, that any differences in behavior arising in Experiment 2 are the result of differences in the contexts of participation and not due to underlying differences in the two populations.

We expect the relationship between lookout duration and checking accuracy to be moderated by the context of participation. Without an experimenter watching, participants might be more inclined to switch to other activities during lockout periods. Being otherwise occupied, combined with the disruptive effects of interruptions, might negatively affect checking performance. We predict increasing lockout durations will increase participants' propensity to switch to other tasks during lockout periods.

Method

Participants

A total of 202 participants (100 female) with a mean age of 34 years (*SD*=10 years) were recruited through Amazon Mechanical Turk. Participants were US residents and had completed at least 100 assignments with an approval rate of 90%. Participants were paid $2 for ~20 minutes of their time, including introduction and debriefing.

Design

Rather than limit our investigation to the same three lockout durations used in Experiment 1, in this experiment we used a continuous design. We tested lockouts from 0-s to 10-s in 100-ms increments for a total of 101 lockout durations. Two participants were randomly allocated to each duration. We measured whether participants confirmed input values that did not match the target. We also measured the frequency of switches away from the experimental task.

Materials

Experiment 2 used the same task as Experiment 1 with modifications for online deployment. Annotations were added to the training trials to aid participants as they learnt how to work the task, with a particular focus on keyboard-based navigation. Annotations appeared next to the widget participants were attempting to operate if they hit the wrong key or tried to click it with the mouse. For instance, if participants tried to hit the return key in any of the text fields, an annotation appeared next to the field advising them to use the tab key. Likewise, any keying activity during the lockout invoked an annotation reiterating the presence of the lockout. The task enforced transcription by preventing the selection of text. This was to prevent participants from copying and pasting values from the target list to the input field.

Procedure

The study was created as an *ExternalQuestion* on AMT. This allowed us to host the study on our own server. Participants who met the selection criteria could view the study information before accepting the assignment (i.e., before consenting to take part). Once participants accepted the assignment they were assigned a condition and given a link to follow that opened the experiment in a new browser tab.

Before the instructions for the task, participants were given general instructions on online participation. This included browser compatibility information and a reiteration of the requirement to use a physical keyboard. These instructions were also included in the pre-acceptance information.

Next, participants were presented with instructions on the operation of the task. This consisted of a text description followed by a set of six slides explaining the steps required to operate the task. In an effort to encourage participants to read the instructions, it was not possible to continue to the training trials until the final slide had been reached.

Participants completed ten training trials split over two blocks. During these trials, incorrect keypresses and clicks invoked annotations that indicated the correct action. After completing the training trials, participants completed 80 experimental trials split over 16 blocks. Participants had a break of at least five seconds between blocks. After the experimental trials, participants completed a 22-item questionnaire and were debriefed. All items were scored on a five-point Likert scale: “Strongly disagree”, “Disagree”, “Neither agree nor disagree”, “Agree” and “Strongly agree”.

Participants’ progress through the experiment was automatically tracked. Once they had finished the experiment their data were automatically submitted. Participants were given a two-hour window in which to complete the assignment. Participation was recorded in a database with WorkerIDs and AssignmentIDs (AMT unique identifiers). Workers were informed at various stages that they should not participate twice, but in the event that workers accepted a second assignment they were met with a screen thanking them for their participation. The only option was to return (i.e., drop-out from) the assignment.

Participants were paid once all assignments in a HIT were completed. This was usually within one hour. The assignment could not be submitted until the task was completed. No assignments were rejected.

Results

General task performance

Each participant completed 80 trials. Eight of these trials were *doctored*. We first consider participants’ performance in the 72 trials that were not modified. In each trial, participants had an opportunity to check and correct any errors before they confirmed their entry. Input that was still incorrect at this point was labelled as a *confirmed error*.

We found that seven participants deviated significantly from the instructions. Four of these participants had 100% confirmed error rates (i.e., they did not complete the task as instructed). Their data were excluded. Three other participants discovered the same ‘shortcut’ as the participant in Experiment 1. These participants used the strategy on fewer trials than the participant in Experiment 1, but we still excluded them to maintain parity between the experiments.

In the unmodified trials, the 195 remaining participants made between 0-9 confirmed errors (M=0.3, SD=0.02, Mdn=0) for a rate of 0.5%. Of these, 155 (79%) of participants made no confirmed errors. Overall, this suggests that participants exhibited a high level of compliance given the absence of any strong constraints on behavior.

Effect of lockout duration on checking accuracy

In Experiment 1, we found that increasing lockout duration increased checking accuracy. We wanted to know whether this was also the case for Experiment 2. We again looked at accuracy in doctored and un-doctored trials. As before, doctored trials transposed two digits of participants’ input to dissociate checking accuracy and typing accuracy.

Participants failed to catch the errors introduced in 0-8 of the doctored trials (M=4, SD=3, Mdn=4), for an error detection rate of 53%. Forty-nine participants did not spot any of the errors introduced in the doctored trials. Thirty-seven participants corrected all of the errors that were introduced.

We fit error counts for the doctored trials using a Poisson regression model. Lockout duration significantly predicted error detection (R^2=.03, b=-4.9×10^{-5}, $t(193)$=4.08, p<.001). Longer lockouts reduced the chance of participants missing

errors. The fit of the model is weak, however, particularly compared to the model in Experiment 1; lockout duration accounts for a much smaller proportion of the variance here.

We returned again to look at data from the un-doctored trials. After the input phase, participants had made 0-17 errors (M=1, SD=2, Mdn=0) for a potential error rate of 1.1%. After the chance to check, participants discovered 58% of the errors that they had made. After the checking phase the error rate was 0.5%. Participants typed extremely accurately: 117 (60%) of them made no errors during the input phase.

We again used a mixed model to investigate whether lockout duration had an effect on checking after we accounted for individual differences in participants' propensities to make errors. We found evidence for an effect of lockout duration on accuracy once again (D_{AIC}=3.27, χ^2=4.3, p=.04). This time, however, the evidence for an effect was much weaker.

Why the difference in lockout effectiveness between Experiment 1 and Experiment 2? There is plenty of evidence to suggest that participants in Experiment 2 were attentive, but lockouts appear to have had less of an influence over checking behavior. Why might that be? We had hypothesized that remote participants might be tempted to switch to other activities during lockout periods and that this would compromise their checking performance. so we next examined participants' self-initiated switching behavior.

Effect of lockout duration on switch frequency

We wanted to know whether longer lockouts were more likely to induce participants to switch to other tasks. We measured switches by keeping track of focus events on the task window. When a participant switched to other tabs or applications the loss of focus was detected. When a participant returned to the task, another event was fired. In this way it was possible to compute the prevalence, duration and timing of computer-based switches.

We focused specifically on switches that occurred entirely within lockout periods as well as those that were initiated (but not completed) during lockouts. Longer lockouts made the experiment last longer, increasing the potential for external interruptions (e.g., instant messages). By only considering switches started during the lockout period we guard against this. It also directly addresses the question of whether lockouts induce switches to other activities.

Of the 195 participants in the sample, 91 made no measurable switches during the experimental trials. (This excludes switches during breaks.) If we consider only the switches that started during the lockout period, 104 participants made no measurable switches. The 91 participants that did initiate switches during lockouts made a total of 880 switches during lockout periods. For those who switched during lockouts, the median number of switches was three, with a mean of ten (SD=14). As Figure 3 shows, much of the switching activity was concentrated among a few participants; 26 of them made ten or more lockout-induced switches.

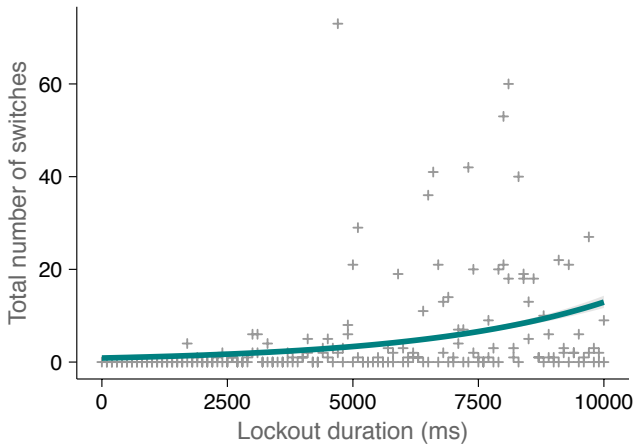

Figure 3. Frequency of self-initiated switches for 195 participants. Teal line shows the Poisson model fit, the narrow grey border on trend link indicates 95% CIs.

We constructed a Poisson regression model to explore the relationship between lockout duration and the total number of switches that participants made during lockout periods. Our model showed that longer lockouts predict increasing in-lockout switching (R^2=.22, b=2.7×10^{-4}, t(193)=19.79, p<.001). When participants were locked-out for longer periods they switched to other tasks more often.

The model also provides an estimate of the maximum duration of lockout before it is likely to induce switches. Because of the nature of the model, as lockout duration increases, the expected number of switches first grows slowly but then begins to grow quickly. Table 3 gives model predictions for maximum lockout durations that are likely to produce fewer than the specified number of switches. Due to the shape of the distribution, lockouts have to be very short to avoid inducing switching behavior. If a small degree of switching is permissible, the model suggests that there is a window in which people will tolerate lockouts of substantial length. Outside this window switching behavior increases rapidly with small increases in lockout duration.

Target switch count	Maximum lockout duration
< 1	512-ms
< 2	3081-ms
< 3	4585-ms
< 5	6478-ms
< 10	9048-ms

Table 3. How long can a lockout be without inducing switches? This table shows target switch counts against maximum lockout duration as predicted by the regression model.

Inferring switches using activity metrics

Our method of recording switches gives us reliable insights into the prevalence of computer-based switching behavior. However, it likely underestimates participants' propensity to switch because it can only 'sees' computer-based switches. It does not know if people watch TV or play with their phones during lockouts. To compensate, we attempt to infer non-computer switches from activity metrics.

We used the period of time between a lockout ending and participants confirming their input as a proxy for switching behavior. If participants switched to other tasks during the lockout period they might have returned to the task some time after the end of the lockout. Therefore, longer gaps between the end of a lockout and the entry being confirmed might imply the occurrence of switching behavior.

For these data, we dropped the two participants with zero-second lockouts because their data were not meaningful when considering post-lockout reaction times. Participants took an average of 2045-ms (*SD*=3381-ms) to hit the *Confirm* button when it became available after the lockout. For those participants who checked their input and experienced very short lockouts, this reaction time data will also include some of the time spent checking.

We constructed a linear regression model with lockout duration as predictor and post-lockout response time as outcome. Lockout duration did not significantly predict post-lockout response time (R^2=.02, b=0.16, $t(192)$=1.90, p=.060).

Although there was no robust evidence that longer lockouts increased input confirmation time, there were indications that non-computer switching behavior may have occurred. Two participants took over 10-seconds on average to confirm their input after the lockout ended. Eight other participants had averages of over five seconds. This analysis is also likely to underestimate the prevalence of switching behavior because it gives no insight into non-computer switches that occur entirely within the period of the lockout.

Participant perceptions of lockouts

We solicited participants' perceptions of the lockouts in a post-experiment questionnaire. One participant's responses were not recorded due to a technical issue. Participants' feelings about the lockouts were predictably negative.

In response to the statement "I would have preferred this study if the 'confirm' button was immediately available after the number moved into the 'check input' box" (Q16), the modal response was *Strongly Agree* (median *Agree*). In response to the statement "I found it frustrating that the confirm button was not instantly available" (Q15), the modal response was *Strongly Agree* (median *Agree*).

We were interested to know whether there was a relationship between lockout duration and self-reported frustration. To this end we used an ordinal logistic regression with lockout duration as a predictor and dummy-coded responses to Q15 as the outcome. The model showed that lockout duration significantly predicted subjective reports of frustration ($b=2.2\times10^{-4}$, $t(193)$=4.65, p<.001). Longer lockouts were reported as being more frustrating.

Despite these negative feelings, participants seemed to indicate that the lockout made them perform better. In response to the question "I used the time before the 'confirm' button became available to check my input", the modal and median responses were *Strongly Agree*. In response to the statement "If I did not have the opportunity to check my input, my performance would have been less accurate", the modal and median responses were *Agree*.

Accuracy of JavaScript lockout timing

The duration of lockouts during the task was set in milliseconds and scheduled using JavaScript's *setTimeout* function. This function allows commands to be executed some number of milliseconds in the future. The temporal accuracy of this function in scheduling events is very high when a window is in focus but it is not guaranteed. If other windows are active or if a system is under high load, the function may not execute with the delay specified.

Given the switching behavior that we revealed earlier, we took the prudent step of determining whether the lockout durations were accurately enforced during the experiment. We compared the intended lockout durations with the durations that were observed. We took the mean measured lockout duration for each participant and computed the absolute difference between the measured mean and the specified duration. We found that these aggregate means varied from the intended duration by a mean of 18-ms (*SD*=51-ms). A Pearson correlation revealed a near perfect correlation (r>.999). Our manipulations of lockout duration were accurately enforced by browsers.

Discussion

In Experiment 2 we investigated whether lockouts would still encourage checking in the context of a naturalistic online experiment. The data reveal a much-weakened link between lockout duration and checking accuracy. Our switching data suggest that this difference is attributable to longer lockouts inducing participants to switch to other tasks.

Although participants felt that the lockouts made them more accurate, this feeling was not supported by objective data. As one might expect, most participants found the lockouts to be frustrating and there was a statistically significant relationship between lockout duration and frustration; longer lockouts led to higher self-reports of frustration.

GENERAL DISCUSSION

In this paper we have reported two investigations of the effect of lockouts on checking performance. Our first experiment tested whether lockouts could improve checking performance in a laboratory task. We found that longer lockouts boost checking performance. Our second study investigated whether lockouts would still work in an online environment where participants could perform other tasks concurrently. We found that lockouts were no longer as effective and attributed this to participants switching to other activities during lockout periods.

This paper makes novel and significant contributions to three areas. First, we contribute a new perspective on interface and task design to the number-entry literature. Second, we contribute to the body of lockouts research, deepening our understanding of the practical hurdles that need to be

overcome if lockouts are to be implemented. Finally, we make a methodological contribution; our approach makes novel and positive use of the diminished control that comes with online crowdsourced experiments. We consider each of these contributions in turn.

Lockouts in number entry

Much of the number-entry literature has focused on how numbers are typed rather than explicitly on interventions to encourage checking [40,55,56]. This paper builds on Wiseman et al.'s [57] efforts to develop interventions that encourage post-entry checking. Our results suggest that for a particular set of number-entry tasks, lockouts might be a design approach that is worth exploring in more depth.

There are, however, two reasons to be cautious about applying lockouts in the context of number-entry. The first is that we show longer lockouts increase participants' feelings of frustration. This is potentially problematic because negative affective states reduce accuracy in number-entry tasks [9]. There are also practical issues with implementing lockouts. The cumulative time lost to, for instance, alarms on medical devices already costs hospitals large sums in lost time [32]. Introducing even very short lockouts may have large impacts when aggregated over an entire healthcare system. Whether it would be worth the trade-off for the promise of improved accuracy would need to be considered.

Lockouts as a design pattern

Our second contribution is to our understanding of how people interact with lockouts. Previous efforts have focused on the planning stages of tasks [7,39], but here we have shown that, in a laboratory setting, lockouts can also encourage people to perform additional checking steps after they have completed the primary goal of a task.

We have also demonstrated that although lockouts are effective in controlled environments, taking them into less constrained settings can have unintended consequences. In this paper we show that longer lockouts make participants more likely to engage in self-initiated multitasking behavior. This behavior means that checking steps are omitted and that participants are subject to negative effects of interruptions.

Our results show that utility-maximization theories of lockout efficacy (e.g., [39]) only hold true if people are faced with a choice between using the period of a lockout to work or do nothing. If people have the option of allocating their time to other tasks during lockout periods, they may find more utility in switching than staying to plan or check. By running Experiment 2 in a less controlled setting, we were able to study this effect empirically: people deemed other tasks to be more important to them than checking and switched.

Crowdsourcing naturalistic experiments

Finally, we make a novel methodological contribution. There has been plenty of work showing that online studies produce reliable data [20,29,30,44]. Tools for spotting issues with data quality have also been developed with some success [49]. Work in the area has until now focused on mitigating the diminished control inherent in online experimentation (e.g., [18,29,46]).

In this paper we have shown that crowd experiments, by virtue of not having the same conventions and constraints as laboratories, allow us to address research questions in a more natural setting. Running Experiment 2 using AMT gave us insights into the relationship between lockouts and self-interruptions that we may have been entirely blind to if we had continued with a lab-based program of investigation. We are not aware of work that has made a conscious decision to leverage diminished control in crowdworking settings to address new research questions in this way.

Our empirical results also have lessons for crowdsourcing more broadly. Maintaining worker attention is a challenge for those setting tasks on crowdworking platforms [10,29,46]. Often *attention checks* are used to see if participants are paying attention. Our data suggest that very brief lockouts might provide a marginal increase in attentiveness in routine tasks, but that longer lockouts are likely to have little positive effect on performance. Making workers pause has improved attention in surveys [27]; our results suggest lockouts might be worth investigating further for routine human performance tasks too.

Generalizability

Our experiment used random sets of digits and errors were not costly. In safety-critical settings numbers often have meaning and errors can have serious consequences. This limits the generalizability of our findings. That said, scenarios involving recognizable numbers and serious consequences may actually be amenable to lockouts. Where people are motivated to spot errors and the majority of their input is easily recognizable, lockouts can be even shorter and still be sufficient for people to check their input.

Both experiments explore whether lockouts make people more likely to check their input. We do not consider whether other designs might improve checking further. One thing to be said for lockouts is that they are easy to add to existing systems – they can be implemented using standard interface widgets, or on devices with limited seven-segment displays.

ACKNOWLEDGEMENTS

We would like to thank Dr J M Saxton for his kind permission to use Figure 1. We also thank the reviewers for taking the time to review this manuscript. This work was supported by the UK Engineering and Physical Sciences Research Council grants EP/G059063/1 and EP/L504889/1.

REFERENCES

1. Erik M. Altmann, Gregory J. Trafton, and David Z. Hambrick. 2013. Momentary Interruptions Can Derail the Train of Thought. *Journal of Experimental Psychology: General* 143, 1: 215–226. http://doi.org/10.1037/a0030986
2. Gerry Armitage. 2008. Double checking medicines: defence against error or contributory factor? *Journal of*

Evaluation in Clinical Practice 14, 4: 513–519. http://doi.org/10.1111/j.1365-2753.2007.00907.x

3. Jonathan Back, Duncan P. Brumby, and Anna L Cox. 2010. Locked-out: investigating the effectiveness of system lockouts to reduce errors in routine tasks. *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, ACM, 3775–3780. http://doi.org/10.1145/1753846.1754054
4. Jelmer P. Borst, Trudy A. Buwalda, Hedderik van Rijn, and Niels A. Taatgen. 2013. Avoiding the problem state bottleneck by strategic use of the environment. *Acta Psychologica* 144, 2: 373–379. http://doi.org/10.1016/j.actpsy.2013.07.016
5. Jelmer P. Borst, Niels A. Taatgen, and Hedderik van Rijn. 2015. What Makes Interruptions Disruptive?: A Process-Model Account of the Effects of the Problem State Bottleneck on Task Interruption and Resumption. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 2971–2980. http://doi.org/10.1145/2702123.2702156
6. Juliana J. Brixey, Zhihua Tang, David J. Robinson, et al. 2008. Interruptions in a level one trauma center: A case study. *International Journal of Medical Informatics* 77, 4: 235–241. http://doi.org/10.1016/j.ijmedinf.2007.04.006
7. Duncan P. Brumby, Anna L. Cox, Jonathan Back, and Sandy J. J. Gould. 2013. Recovering from an interruption: Investigating speed-accuracy trade-offs in task resumption behavior. *Journal of Experimental Psychology: Applied* 19, 2: 95–107. http://doi.org/10.1037/a0032696
8. Michael D. Byrne and Susan Bovair. 1997. A Working Memory Model of a Common Procedural Error. *Cognitive Science* 21, 1: 31–61. http://doi.org/10.1207/s15516709cog2101_2
9. Paul Cairns, Pratyush Pandab, and Christopher Power. 2014. The Influence of Emotion on Number Entry Errors. *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 2293–2296. http://doi.org/10.1145/2556288.2557065
10. Jesse Chandler, Pam Mueller, and Gabriele Paolacci. 2014. Nonnaïveté among Amazon Mechanical Turk workers: Consequences and solutions for behavioral researchers. *Behavior Research Methods* 46, 1: 112–130. http://doi.org/10.3758/s13428-013-0365-7
11. Suzanne C. Charman and Andrew Howes. 2003. The Adaptive User: An Investigation Into the Cognitive and Task Constraints on the Generation of New Methods. *Journal of Experimental Psychology: Applied* 9, 4: 236–248. http://doi.org/10.1037/1076-898X.9.4.236
12. Carey D Chisholm, Edgar K Collison, David R Nelson, and William H Cordell. 2000. Emergency Department Workplace Interruptions Are Emergency Physicians "Interrupt-driven" and "Multitasking"? *Academic Emergency Medicine* 7, 11: 1239–1243. http://doi.org/10.1111/j.1553-2712.2000.tb00469.x
13. Laura Dabbish, Gloria Mark, and Víctor M González. 2011. Why do I keep interrupting myself?: environment, habit and self-interruption. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 3127–3130. http://doi.org/10.1145/1978942.1979405
14. Serge Egelman, Lorrie Faith Cranor, and Jason Hong. 2008. You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1065–1074. http://doi.org/10.1145/1357054.1357219
15. William Gardner, Edward P. Mulvey, and Esther C. Shaw. 1995. Regression analyses of counts and rates: Poisson, overdispersed Poisson, and negative binomial models. *Psychological Bulletin* 118, 3: 392–404. http://doi.org/10.1037/0033-2909.118.3.392
16. Laura Germine, Ken Nakayama, Bradley C. Duchaine, Christopher F. Chabris, Garga Chatterjee, and Jeremy B. Wilmer. 2012. Is the Web as good as the lab? Comparable performance from Web and lab in cognitive/perceptual experiments. *Psychonomic Bulletin & Review* 19, 5: 847–857. http://doi.org/10.3758/s13423-012-0296-9
17. Victor M. González and Gloria Mark. 2004. "Constant, constant, multi-tasking craziness": managing multiple working spheres. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 113–120. http://doi.org/10.1145/985692.985707
18. Joseph K. Goodman, Cynthia E. Cryder, and Amar Cheema. 2013. Data Collection in a Flat World: The Strengths and Weaknesses of Mechanical Turk Samples. *Journal of Behavioral Decision Making* 26, 3: 213–224. http://doi.org/10.1002/bdm.1753
19. Sandy J. J. Gould, Anna L. Cox, and Duncan P. Brumby. 2013. Frequency and Duration of Self-initiated Task-switching in an Online Investigation of Interrupted Performance. *Human Computation and Crowdsourcing: Works in Progress and Demonstration Abstracts AAAI Technical Report CR-13-01*, AAAI, 22–23.
20. Sandy J. J. Gould, Anna L. Cox, Duncan P. Brumby, and Sarah Wiseman. 2015. Home is Where the Lab is: A Comparison of Online and Lab Data From a Time-sensitive Study of Interruption. *Human Computation* 2, 1: 45–67. http://doi.org/10.15346/hc.v2i1.4
21. Wayne D. Gray and Deborah A. Boehm-Davis. 2000. Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology:*

Applied 6, 4: 322–335. http://doi.org/10.1037/1076-898X.6.4.322

22. Robert A. Green, George Hripcsak, Hojjat Salmasian, et al. 2015. Intercepting Wrong-Patient Orders in a Computerized Provider Order Entry System. *Annals of Emergency Medicine* 65, 6: 679–686.e1. http://doi.org/10.1016/j.annemergmed.2014.11.017
23. Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 203–212. http://doi.org/10.1145/1753326.1753357
24. Helen M. Hodgetts and Dylan M. Jones. 2006. Interruption of the Tower of London task: Support for a goal-activation approach. *Journal of Experimental Psychology: General* 135, 1: 103–115. http://doi.org/10.1037/0096-3445.135.1.103
25. Ling Jiang, C. Wagner, and B. Nardi. 2015. Not Just in it for the Money: A Qualitative Investigation of Workers' Perceived Benefits of Micro-task Crowdsourcing. *2015 48th Hawaii International Conference on System Sciences (HICSS)*, 773–782. http://doi.org/10.1109/HICSS.2015.98
26. Jing Jin and Laura A Dabbish. 2009. Self-interruption on the computer: a typology of discretionary task interleaving. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1799–1808. http://doi.org/10.1145/1518701.1518979
27. Adam Kapelner and Dana Chandler. 2010. Preventing Satisficing in online surveys. *CrowdConf*.
28. Ioanna Katidioti and Niels A. Taatgen. 2014. Choice in Multitasking How Delays in the Primary Task Turn a Rational Into an Irrational Multitasker. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 56, 4: 728–736. http://doi.org/10.1177/0018720813504216
29. Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with Mechanical Turk. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 453–456. http://doi.org/10.1145/1357054.1357127
30. Steven Komarov, Katharina Reinecke, and Krzysztof Z. Gajos. 2013. Crowdsourcing Performance Evaluations of User Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 207–216. http://doi.org/10.1145/2470654.2470684
31. Ross Koppel, Tosha Wetterneck, Joel Leon Telles, and Ben-Tzion Karsh. 2008. Workarounds to Barcode Medication Administration Systems: Their Occurrences, Causes, and Threats to Patient Safety. *Journal of the American Medical Informatics Association* 15, 4: 408–423. http://doi.org/10.1197/jamia.M2616
32. Paul T Lee, Frankie Thompson, and Harold Thimbleby. 2012. Analysis of infusion pump error logs and their significance for health care. *British Journal of Nursing* 21, Sup5: S12–S20. http://doi.org/10.12968/bjon.2012.21.Sup5.S12
33. Simon Y. W. Li, Ann Blandford, Paul Cairns, and Richard M. Young. 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, 4: 314–328. http://doi.org/10.1037/a0014397
34. Simon Y. W. Li, Anna L. Cox, Ann Blandford, Paul Cairns, and A. Abeles. 2006. Further investigations into post-completion error: the effects of interruption position and duration. *Proceedings of the 28th Annual Meeting of the Cognitive Science Conference*, Cognitive Science Society, 471–476.
35. Gloria Mark, Victor M González, and Justin Harris. 2005. No task left behind?: examining the nature of fragmented work. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 321–330. http://doi.org/10.1145/1054972.1055017
36. Winter Mason and Siddharth Suri. 2012. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods* 44, 1: 1–23. http://doi.org/10.3758/s13428-011-0124-6
37. Christopher A. Monk, J. Gregory Trafton, and Deborah A. Boehm-Davis. 2008. The Effect of Interruption Duration and Demand on Resuming Suspended Goals. *Journal of Experimental Psychology: Applied* 14, 4: 299–313. http://doi.org/10.1037/a0014402
38. Kenton P. O'Hara and Stephen J. Payne. 1998. The Effects of Operator Implementation Cost on Planfulness of Problem Solving and Learning. *Cognitive Psychology* 35, 1: 34–70. http://doi.org/10.1006/cogp.1997.0676
39. Kenton P. O'Hara and Stephen J. Payne. 1999. Planning and the user interface: the effects of lockout time and error recovery cost. *International Journal of Human-Computer Studies* 50, 1: 41–59. http://doi.org/10.1006/ijhc.1998.0234
40. Patrick Oladimeji, Harold Thimbleby, and Anna Cox. 2011. Number Entry Interfaces and Their Effects on Error Detection. In *Human-Computer Interaction – INTERACT 2011*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque and Marco Winckler (eds.). Springer Berlin Heidelberg, 178–185. http://doi.org/10.1007/978-3-642-23768-3_15

41. Kai A. Olsen. 2008. The $100,000 Keying Error. *Computer 41*, 106–108. http://doi.org/10.1109/MC.2008.135
42. Antti Oulasvirta and Pertti Saariluoma. 2006. Surviving task interruptions: Investigating the implications of long-term working memory theory. *International Journal of Human-Computer Studies* 64, 10: 941–961. http://doi.org/10.1016/j.ijhcs.2006.04.006
43. Gabriele Paolacci and Jesse Chandler. 2014. Inside the Turk Understanding Mechanical Turk as a Participant Pool. *Current Directions in Psychological Science* 23, 3: 184–188. http://doi.org/10.1177/0963721414531598
44. Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis. 2010. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making* 5, 5: 411–419.
45. Stephen J. Payne and Andrew Howes. 2013. Adaptive Interaction: A Utility Maximization Approach to Understanding Human Interaction with Technology. *Synthesis Lectures on Human-Centered Informatics* 6, 1: 1–111. http://doi.org/10.2200/S00479ED1V01Y201302HCI016
46. Eyal Peer, Joachim Vosgerau, and Alessandro Acquisti. 2014. Reputation as a sufficient condition for data quality on Amazon Mechanical Turk. *Behavior Research Methods* 46, 4: 1023–1031. http://doi.org/10.3758/s13428-013-0434-y
47. Raj M. Ratwani, J. Malcolm McCurry, and J. Gregory Trafton. 2008. Predicting postcompletion errors using eye movements. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 539–542. http://doi.org/10.1145/1357054.1357141
48. L. T. C Rolt. 2007. *Red for danger: the classic history of British railway disasters*. Sutton, Stroud.
49. Jeffrey M. Rzeszotarski and Aniket Kittur. 2012. CrowdScape: interactively visualizing user behavior and output. *Proceedings of the 25th annual ACM symposium on User interface software and technology*, ACM, 55–62. http://doi.org/10.1145/2380116.2380125
50. Dario D. Salvucci. 2010. On reconstruction of task context after interruption. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 89–92. http://doi.org/10.1145/1753326.1753341
51. Harold Thimbleby. 2008. Ignorance of interaction programming is killing people. *interactions* 15, 5: 52–57. http://doi.org/10.1145/1390085.1390098
52. Harold Thimbleby and Paul Cairns. 2010. Reducing number entry errors: solving a widespread, serious problem. *Journal of The Royal Society Interface* 7, 51: 1429–1439. http://doi.org/10.1098/rsif.2010.0112
53. Ricardo Mark Villamarín-Salomón and José Carlos Brustoloni. 2010. Using Reinforcement to Strengthen Users' Secure Behaviors. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 363–372. http://doi.org/10.1145/1753326.1753382
54. Johanna I Westbrook, Enrico Coiera, William T M Dunsmuir, et al. 2010. The impact of interruptions on clinical task completion. *Quality and Safety in Health Care* 19, 4: 284–289. http://doi.org/10.1136/qshc.2009.039255
55. Sarah Wiseman, Duncan P. Brumby, Anna L. Cox, and Orla Hennessy. 2013. Tailoring Number Entry Interfaces To The Task of Programming Medical Infusion Pumps. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 57, 1: 683–687. http://doi.org/10.1177/1541931213571148
56. Sarah Wiseman, Paul Cairns, and Anna Cox. 2011. A Taxonomy of Number Entry Error. *Proceedings of the 25th BCS Conference on Human-Computer Interaction*, British Computer Society, 187–196.
57. Sarah Wiseman, Anna L. Cox, Duncan P. Brumby, Sandy J.J. Gould, and Sarah O'Carroll. 2013. Using Checksums to Detect Number Entry Error. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2403–2406. http://doi.org/10.1145/2470654.2481332