The Effect Of Interface Type On Visual Error Checking Behavior

Sarah Wiseman, Judith Borghouts, Dora Grgic, Duncan P. Brumby, Anna L. Cox
UCL Interaction Centre, University College London, London, UK
{sarah.wiseman.10, judith.borghouts.14, d.brumby, anna.cox}@ucl.ac.uk,
dorijana.grgic@gmail.com

During data entry tasks, small errors can result in catastrophe, for instance adding an extra zero to a drug dose when programming an infusion in a hospital. For this reason understanding users' error checking behavior is highly important. One aspect that can affect error checking is the interface that a user must interact with to enter data. Often user interaction with interfaces is evaluated based on speed or error rate. In this paper, in addition to this, we also explore how different types of interface can affect a user's error checking behavior in a multitasking environment. We show that a fast to use and familiar interface discourages users from carrying out thorough visual checking in a number transcription task. We also found that having participants perform an additional secondary task while doing the number entry task made participants less likely to check the inputted numbers for errors.

INTRODUCTION

Tasks involving data entry can be found in a variety of safety-critical systems, from the medical domain (Vicente, Kada-Bekhaled, Hillel, Cassano, & Orser, 2003) to aviation (Network, 1992). The dangers of errors during such tasks are well known, and can result in serious harm or death (Institute of Safe Medication Practices, 2007). Improving the accuracy of data entry tasks is therefore a high priority for HCI research. Consequentially, much of the research at present has involved investigation of factors that affect the causes of errors, such as poorly formatted numbers (Thimbleby & Cairns, 2010) or situations that require high cognitive load (Wiseman, Cairns, & Cox, 2011). However, this is not the only perspective from which to view the issue of improving the accuracy of data entry tasks.

As Reason observes in his "Swiss Cheese Model", the more layers of defense against hazards, the less likely an error is to occur (Reason, 1990). For this reason, research has looked beyond the causes of error to investigate error checking behavior. An interface that can encourage error checking by the user, or make the task easier, has the potential to lower error rate further.

In one previous experiment investigating number entry error checking, users were specifically asked to double-check the numbers they had entered. Using this system, participants were only able to catch 36.18% of errors made (Wiseman, Cox, Brumby, Gould, & O'Carroll, 2013). In this work, Wiseman et al. asked participants to enter more data values to allow a natural "checksum" to be used in the medical task under investigation (two of the numbers presented could be used to calculate the third). This approach resulted in all errors being caught. However, not all data entry tasks can rely upon checksums. In such situations, entering numbers twice has been shown to significantly reduce the errors made (Barchard & Pace, 2011). However, this may not be an optimal solution as it requires the user to enter values twice, and this takes time.

Other studies into error checking in data entry have focused on the effect of the interface on error checking strategies. In one such experiment, two different number entry interfaces were compared: incremental (where the number displayed was gradually incremented by the interface, as with a volume control dial) and serial (where the number was added to, one digit at a time as with a telephone). During the experiment, errors were injected into the numbers that the participants had entered. When users were asked to transcribe numbers using both interfaces, it was found that incremental number entry interfaces resulted in participants noticing more of the artificially inserted errors (Oladimeji, Thimbleby, & Cox, 2011). Users of the incremental interface also spent more time looking at the device display compared to the serial interface. However, it is not clear from this analysis of eye-tracking data which area of the display participants were viewing, therefore making it difficult to know for certain whether participants were actually checking the numbers they had entered.

Previous studies have investigated error checking behavior at a more detailed level by analyzing at what points during a data entry task users choose to check what they have entered. Research has shown that a large number of different error checking strategies are used when transcribing multi-digit numbers (Smith, Lewis, Howes, Chu, Green, & Vera, 2008). This has the potential to improve our understanding of number entry error checking because some of the strategies appear more thorough than others, and may therefore be more likely to result in a number entry error being noticed. However, Smith et al. only investigated the serial interface.

In this paper, we investigate how using either a serial or an incremental number entry interface affects users' error checking strategy. Participants in our study performed a number entry task in a dual-task situation. We used a dual-task setup because in many applied settings number entry tasks are often interrupted by other tasks (Westbrook, Woods, Rob, Dunsmuir, & Day, 2010), and prior research has shown that such interruptions increase the likelihood of errors being made (e.g., Brumby, Cox, Back, & Gould, 2013).

METHOD

Participants

Twenty-two participants (16 male) were recruited from the UCL Psychology Subject Pool. The mean age of participants was 24 (range: 19-38). Participants were reimbursed £7 for their time. All participants were experienced computer users and had played Tetris before.

Design

A within-subjects design was used with two different independent variables: interface type and secondary task difficulty. Two interface types were used: serial and incremental. The secondary task was a Tetris game, which had three levels: no game, slow or fast. All participants took part in each of the six conditions.

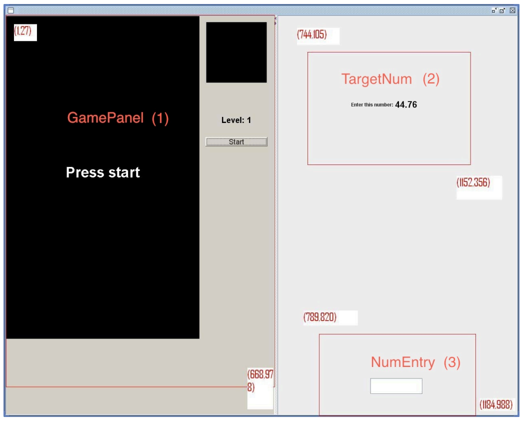
The dependent variables were the error rate, the speed of entry and the timing of eye fixations on the entry interface and the different areas of the display.

Materials

The number entry task and Tetris game were presented as a Java program running on a desktop PC. A Tobii eye-tracking system was used in conjunction with the PC to gather eye-tracking data. In order to study participants' checking behavior, their visual fixation on four components was measured: the Tetris game panel, the target number, the entry box and the input interface. The first three components were specific areas on the screen (a screenshot with the identified components can be seen in Figure 1). If no visual fixation on the screen could be measured, it was assumed that visual focus was on the input interface.

The software incorporated both the primary number entry task and the secondary Tetris task. If the participant was in a Tetris-playing condition, the left panel on the screen constantly displayed a Tetris game. On the right hand side of screen was the number entry task. This consisted of the number to be entered at the top, and the currently entered number at the bottom of the screen. This separation was necessary in order to determine the area of interest during the eye-tracking analysis. During the two conditions (one per interface type) with no Tetris the number entry task panel covered the entire window.

Participants were given two different interfaces to use when entering numbers. The serial number interface involves entering a number, one digit at a time in much the same way as a telephone number. In the serial number entry task, the keyboard number pad was used. The incremental interface asked users to add amounts to a value on screen. During incremental number entry, the number pad was adapted with chevron keys (see Figure 2). The single chevron keys either added or subtracted 1 from the number on screen whereas the double chevron keys added or subtracted 0.1 from the number on screen. Regardless of interface type, the arrow keys were used to control the Tetris game. The numbers participants were required to transcribe were randomly generated numbers ranging between 0 and 100 and had two decimal places (e.g., 97.96).



Enter this number: 44.76

Figure 1: Number entry screen showing the components. It shows the target number to be entered (44.76), the entry box where the typed number will be shown and the Tetris game panel on the left.

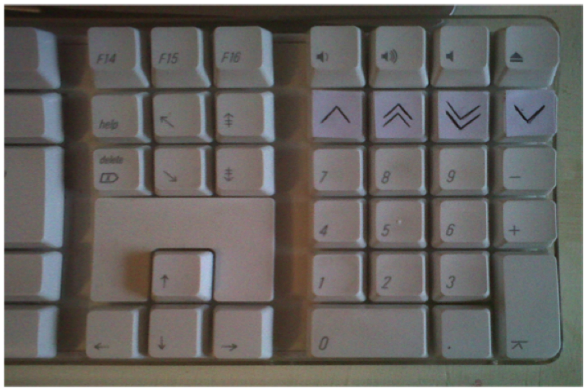


Figure 2: Keyboard used for number entry during the experiment. Serial number entry was completed on the number pad, whereas incremental number entry was completed using the chevron keys.

Procedure

Prior to starting the task participants were given time to practice playing Tetris (2 minutes) and entering numbers (10 numbers on each of the two interfaces).

Participants were instructed to enter 90 numbers using each interface. Within these 90 numbers, 30 were entered in the no Tetris condition, 30 in the slow and 30 in the fast condition. Participants were told to keep the Tetris game running as long as possible. All participants began in the no Tetris condition, and then were presented with either the fast or slow Tetris game first.

RESULTS

The results of an ANOVA show that mean number entry time on the serial interface (M=4.95, SD=2.55 s) was significantly faster than on the chevron interface (M=25.45, SD=11.04 s), $F(1,21)$=260.60, p<.001. Participants were significantly faster entering numbers in the single-task condition (M=9.36, SD=6.45 s) than in the dual-task Tetris condition, $F(1,21)$=40.61, p<.001. There was no difference in speed of entry between the slow Tetris condition (M=17.12, SD=14.07) and the fast Tetris condition (M=19.12, SD=14.93).

The error rate for the serial interface (1.36%) was not significantly different to the error rate for the incremental interface (1.41%). The secondary task difficulty had no significant effect upon error rate.

During the experiment, three error checking behaviors were identified using eye-tracking data. After performing the last key press to enter the number, participants either made no visual check (None), visually checked the number they had just entered (Entered Number) or visually checked both the number they had entered, and the number they had been presented with to copy (Both) before pressing the confirmation button. Each strategy was determined by looking at eye-tracking data and checking for fixations in the Areas Of Interest that represented the number the user had entered and the number they were asked to copy. A final error check was considered to have been made if in the time between the last key press and enter key press visual fixations on only the entry box, or both entry box and target number were measured.

Most commonly, the eye-tracking data showed that participants made no final check after completing the last key press (74.95% of trials, *SD*=19.67%). In 4.22% of trials (*SD*= 6.51%) participants made an Entered Number check and in 20.83% of trials (*SD*=19.75%) participants visually checked both numbers on screen.

An ANOVA test of these results showed that interface type had a significant effect upon the number of times participants checked only the Entered Number. Participants using the serial interface used the Entered Number visual checking strategy significantly more often (6.46% of the time) than when using the chevron interface (1.97% of the time), *F*(1,21)=9.08, *p*=0.006.

Further ANOVA tests show that the Tetris condition had a significant effect upon the number of times a user checked Both the target number and the number entered. Participants checked both numbers 27.12% of the time when not playing Tetris, 17.73% of the time when playing slow Tetris and 17.65% of the time when playing fast Tetris, *F*(1,21)=8.16, *p*<0.001. These results can be seen in Figure 3. The Tetris condition also had a significant effect upon the number of times no visual check was performed. 67.88% of trials performed in the No Tetris condition resulted in participants performing no visual check. 78.18% of trials in the Slow Tetris condition, and 78.79% of trials in the Fast Tetris condition resulted in no visual checks from participants, *F*(1,21)=9.14, *p*<0.001. These results can be seen in Figure 4.

GENERAL DISCUSSION

As with many experiments investigating number entry error, the error rates found in this study were low and did not differ significantly between the two interfaces tested. Adding the secondary task slowed people down, but perhaps it was not difficult enough to increase the error rate.

The error checking behavior however was dependent upon the interface used. When performing a number transcription error check, it is possible for a user to compare the number they have entered to the representation in their short-term memory, or to compare it to the number on screen.

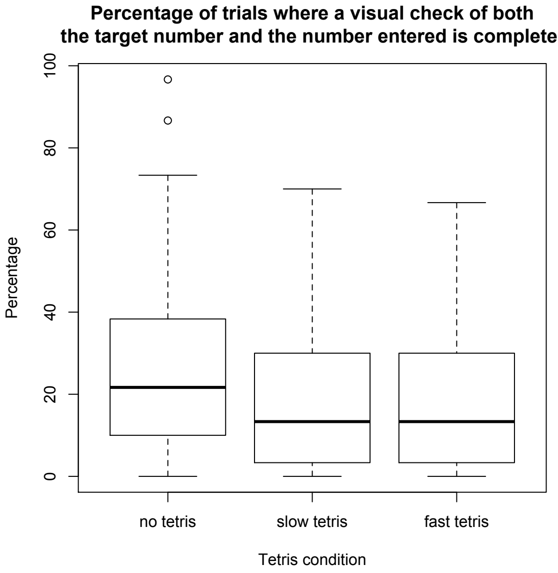


Figure 3: Graph showing percentage of trials where a visual check of both the target number and number entered was completed.

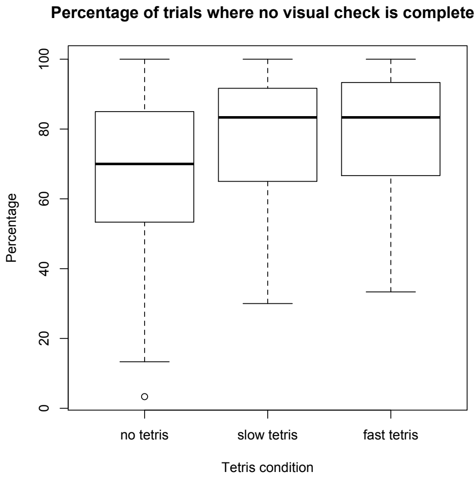


Figure 4: Graph showing percentage of trials where no visual check was completed.

The former of the two checking behaviors involves lower time costs (Gray, Sims, Fu, & Schoelles, 2006), but is potentially more risky, as the memorized number may be incorrect. This strategy was used significantly more often for users of the serial interface compared to the incremental interface.

This may be due to a number of factors. The first key factor is the time taken to enter the numbers on each interface. The incremental interface resulted in significantly longer completion times, and may therefore result in the original number fading in memory thus making a final comparison between the number entered, and the number stored in memory more difficult than with the relatively faster serial interface.

A second potential factor could have been the participants' familiarity with each of the two interfaces. It has been found that participants are more likely to be familiar with the serial interface compared to the incremental interface

(Oladimeji, Thimbleby, & Cox, 2013; Wiseman, Brumby, Cox, & Hennessy, 2013). This familiarity may have led participants to being more confident of their actions, and therefore not checking the number as thoroughly as possible. This result mirrors results from another study which showed that numbers presented in less clear font were more likely to be remembered accurately (Soboczenski, Cairns, & Cox, 2013): when the method of interaction is not familiar or easy, a user may put in more cognitive effort and therefore be more likely to be accurate.

The secondary task also had an effect upon the likelihood of visual checks being performed. When participants did not have the Tetris secondary task they were significantly less likely to enter the numbers without checking the display at all, and were more likely to check both the number entered and the target number. This result suggests that in situations when users are able to focus solely on the number entry task, they are more likely to double-check their work. However, despite the lack of a secondary task, in the majority of trials users still performed no visual check.

No difference was found between the two conditions of Tetris. It appears that the presence of a secondary task alone was enough to change the users' checking behavior. The difficulty of that task does not appear to alter the checking strategy. However, this may reflect more upon the difference in difficulty between the fast and slow Tetris conditions. It could be that the difference in difficulty was not great enough to cause a noticeable effect.

CONCLUSION

This experiment has shown that both data input method type and the presence of a secondary task can influence the thoroughness of error checking behavior during a data entry task. Familiar and fast interfaces can result in more risky error checking strategies. In addition, asking a user to multitask can reduce the likelihood of visual checking.

Future research in this area might investigate the effects of familiarity and speed of use of interfaces to fully understand which of the two affects error checking. Further analysis might investigate other error checking strategies beyond those listed in this paper, for instance performing checks during the transcription of the number. This result could extend to situations beyond number transcription, perhaps to online form completion for example, where data entry accuracy can be improved.

Despite this experiment showing no difference in error rate between the two interfaces tested, in safety-critical systems, it is imperative that the transcription task is as error-free as possible. Interfaces that encourage thorough error checking may therefore be better suited to environments where the results of errors can be disastrous.

REFERENCES

Barchard, K.A. & Pace, L.A. (2011). Preventing human error: The impact of data entry methods on data accuracy and statistical results. *Computers in Human Behavior 27*(5), 1834–1839.

Brumby, D.P., Cox, A.L., Back, J., & Gould, S.J.J. (2013). Recovering from an interruption: Investigating speed-accuracy tradeoffs in task resumption strategy. *Journal of Experimental Psychology: Applied, 19,* 95-107.

Gray, W.D., Sims, C.R., Fu, W.-T., & Schoelles, M.J. (2006). The soft constraints hypothesis: a rational analysis approach to resource allocation for interactive behavior. *Psychological Review, 113*, 461–482.

Institute for Safe Medication Practices Canada. (2007). *Fluorouracil Incident Root Cause Analysis.* Retrieved from http://www.ismp-canada.org/download/reports/FluorouracilIncidentMay2007.pdf.

Network, A.S. (1992). *Air Inter Flight ITF148 Accident Description.* Retrieved from http://aviation-safety.net/database/record.php?id=19920120-0.

Oladimeji, P., Thimbleby, H., & Cox, A. (2011). Number entry interfaces and their effects on error detection. *Human-Computer Interaction–INTERACT 2011,* 178–185.

Oladimeji, P., Thimbleby, H., & Cox, A. (2013). A performance review of number entry interfaces. *Human-Computer Interaction–INTERACT 2013,* 365–382.

Reason, J. (1990). *Human Error*. Cambridge University Press.

Smith, M.R., Lewis, R.L., Howes, A., Chu, A., Green, C., & Vera, A. (2008). More than 8,192 Ways to Skin a Cat: Modeling Behavior in Multidimensional Strategy Spaces. *Proceedings of the 30th annual conference of the Cognitive Science Society,* 1441–1446.

Soboczenski, F., Cairns, P., & Cox, A. (2013). Increasing accuracy by decreasing presentation quality in transcription tasks. *Human-Computer Interaction–INTERACT 2013,* 380–394.

Thimbleby, H., & Cairns, P. (2010). Reducing number entry errors: solving a widespread, serious problem. *Journal of the Royal Society, Interface 7*(51), 1429–1439.

Vicente, K.J., Kada-Bekhaled, K., Hillel, G., Cassano, A., & Orser, B. (2003). Programming errors contribute to death from patient-controlled analgesia: case report and estimate of probability. *Canadian Journal of Anaesthesia 50*(4), 328–332.

Westbrook, J. I., Woods, A., Rob, M. I., Dunsmuir, W. T. M., & Day, R. O. (2010). Association of interruptions with an increased risk and severity of medication administration errors. *Archives of Internal Medicine, 170*, 683–690.

Wiseman, S., Cairns, P., & Cox, A.L. (2011). A taxonomy of number entry error. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction* (BCS-HCI '11). British Computer Society.

Wiseman, S., Cox, A.L., Brumby, D.P., Gould, S.J.J., & O'Carroll. S. (2013). Using checksums to detect number entry error. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13* (pp. 2403-2406). ACM.

Wiseman, S., Brumby, D.P., Cox, A.L., & Hennessy, O. (2013). Tailoring number entry interfaces to the task of programming medical infusion pumps. In *Proceedings of the Annual Meeting of the Human Factors and Ergonomics Society, HFES 2013* (pp 683-687). HFES.