Working Memory Load Affects Device-Specific but Not Task-Specific Error Rates

Maartje G. A. Ament (M.Ament@ucl.ac.uk)

Anna L. Cox (Anna.Cox@ucl.ac.uk)

Ann Blandford (A. Blandford@ucl.ac.uk)

Duncan Brumby (Brumby@cs.ucl.ac.uk)

UCL Interaction Centre
University College London
Gower Street London, WC1E 6BT

Abstract

Human error in routine procedural tasks is often attributed to momentary failures to remember what step to perform. We argue that task-specific steps, which can be defined as actions required to achieve a particular goal across a variety of different devices, are far less prone to error than device-specific steps, which can be defined as actions that are required for the operation of the device but do not directly contribute to the goal. An experiment is reported that supports this distinction, showing that device-specific steps are more error prone than task-specific steps. Moreover, we argue that these errors reflect a failure of memory because the error rate for device-specific steps was sensitive to increased working memory load, while the error rate for task-specific steps was not. The current work demonstrates that a distinction between device- and task-specific steps can be effective in explaining error patterns observed on a specific task.

Keywords: human error; device-specific error; working memory load.

Introduction

While routine procedural errors occur only occasionally, they are persistent. A growing body of empirical work has studied these errors in the laboratory. Most of them have focussed on the post-completion error (PCE) (e.g. Byrne & Bovair, 1997; Chung & Byrne, 2008; Li, Blandford, Cairns, & Young, 2008), a cognitive slip that occurs when the final step in a task is omitted after the main goal has already been completed.

The PCE is theoretically well understood. An influential account is the *memory-for-goals* model developed by Altmann and Trafton (2002). This account assumes that goals are declarative memory representations (chunks) with an associated activation level. The interference level is defined as the 'collective effect of distractor goals'. In order to direct behaviour, the relevant goal needs to be above the interference level. In order to overcome the interference level, the activation of goals must be strengthened. A goal that is retrieved more often or the most recently retrieved subgoal will have a higher activation value than others with less history. Associative links between goals allow activation to spread to other goals. The PC step is usually remembered because it receives associative activation from the step preceding it. Moreover, Byrne and Bovair (1997) have argued that upon completion of the main goal, the sources of activation for the PC subgoal are reduced, leading to lower activation on the PC subgoal, often to a point where it cannot be retrieved.

Another step that is associated with a relatively high error rate is the device-initialisation (DI) step. A device initialisation step is an action that must be executed before the main task steps can be completed (e.g. pressing a 'mode' key before setting the alarm on a digital watch). Li et al. (2008) and Hiltz, Back & Blandford (2010) found relatively high error rates on both the post-completion and the device-initialisation steps. However, this error is less well understood, and it is not clear how the *memory-for-goals* model would account for it. For this error, the main goal has not yet been completed, so should still provide activation for the device-initialisation step.

A common factor that the PC step and the DI step share is that they are both device-specific (Cox & Young, 2000). This means that they do not make a direct contribution towards the main goal, but are only required for the correct operation of the device. Task-specific steps, on the other hand, do make a direct contribution towards the main goal and are required regardless of the type of device they are carried out on. Consider the example of using a state-of-the-art induction hob. A typical task-specific step may be to increase or decrease the power output by pressing the '+' or '-' button, whereas a device-specific step may be to press the selector button to cycle through the different hobs until you have selected the one for which you want to adjust the power. While a number of previous studies have discussed concepts similar to device- and task-specific steps (e.g. Cox & Young, 2000; Kirschenbaum, Gray, Ehret, & Miller, 1996; Gray, 2000), this is a novel approach to explaining routine procedural errors.

In this paper, we propose that the distinction between task-specific and device-specific steps can explain why some steps in a procedure appear to be more error prone than others. Our account relies on the user having a task model (how to do the task) and a device model (how to do the task using a particular device), two concepts widely used

in the field of human-computer interaction research (Young, 1983). Device-specific steps are only represented in the device model, whereas task-specific steps are represented in both. Using an activation-based approach, the current work hypothesises that device-specific steps have lower activation levels, because they have only one source of activation (the device model), whereas task-specific steps receive activation from two sources (the device model and the task model). These lower activation levels make it more likely that device-specific steps fall below the interference level, resulting in a slip. Ament, Blandford & Cox (2009) describe an experiment in which device-specific error rates on the 'Spy task' were significantly higher than those on task-specific steps, as predicted.

There are two aims to this paper. First, we seek to provide empirical evidence to support the idea that error rates are higher on device-specific steps than on task-specific steps. Second, we investigate the effect that varying working memory load has on these two classes of steps. We argue there is good reason to believe that device-specific steps are more susceptible to the deleterious effects of increased working memory load than task-specific steps.

Byrne and Bovair (1997) argued that post-completion errors are memory-based failures. Therefore, they investigated how working memory load affects the PCE. They found that the frequency of the PCE increased under a high working memory load. Byrne and Bovair (1997) argued that a higher working memory load leads to the scaling back of activation on all items in memory. This means that the decay rate is higher, and items are displaced from memory faster. If the source of activation for an item is lost, such as on the post-completion step, it is more likely that that step will not reach the threshold necessary to be executed and a post-completion error will be likely.

However, this account does not explain how working memory load would affect other device-specific errors, since their source of activation is not lost like that of the PC step. In the *memory-for-goals* model (Altmann & Trafton, 2002), higher working memory load is represented by an increased interference level. While no direct predictions about the effect of this are made, it seems clear that an increased interference level makes it more likely that the activation level for a given action falls below it, leading to an error. We therefore hypothesise that device-specific errors should be particularly affected by an increase in working memory load, because a higher interference level makes it even more difficult for device-specific steps to overcome this. Conversely, task-specific steps are expected to be affected less, because their higher activation levels make them more robust to increases in the interference level.

We investigate the effect of working memory load on device-specific and task-specific error rates, by means of a secondary load task. It is expected that in low memory load conditions, participants will make fewer errors overall compared to high load conditions. Critically, it is expected that, under high load, there will be proportionally more errors on device- specific steps than on task-specific steps.

Method

Participants

Forty participants were recruited from a dedicated psychology subject database. They were aged between 18

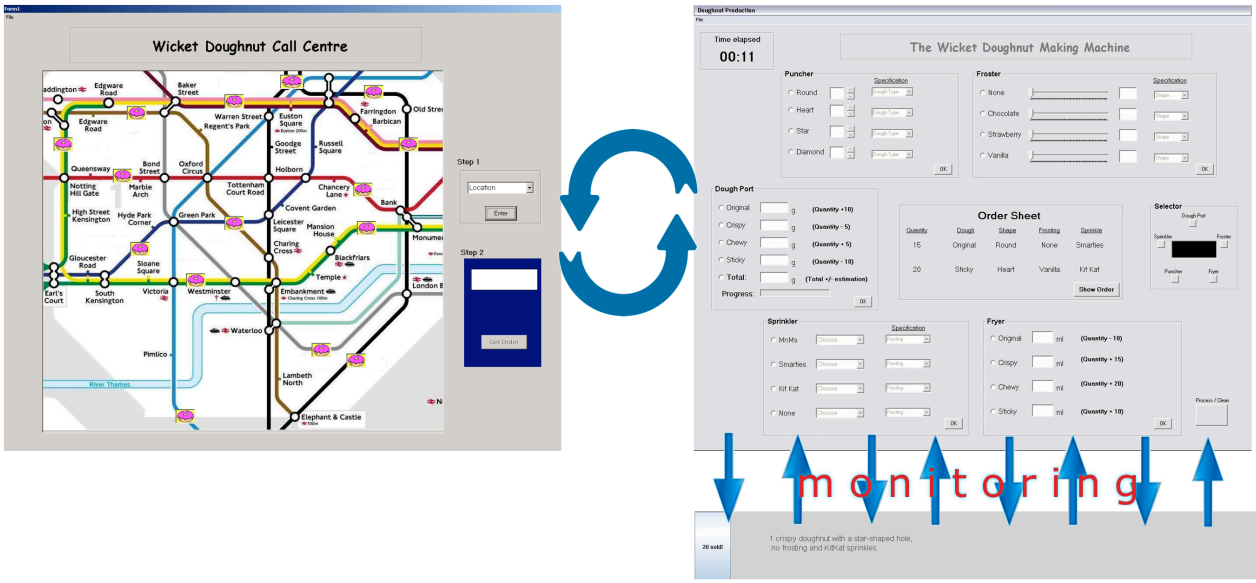

Figure 1: Diagrammatic representation of the Doughnut task. On the top right is the main Doughnut task interface. While making the doughnuts, participants monitor the Doughnut Live Feed, displayed directly underneath the main Doughnut task interface. In between doughnut making trials, participants answer a call at the Call Centre, displayed on the left.

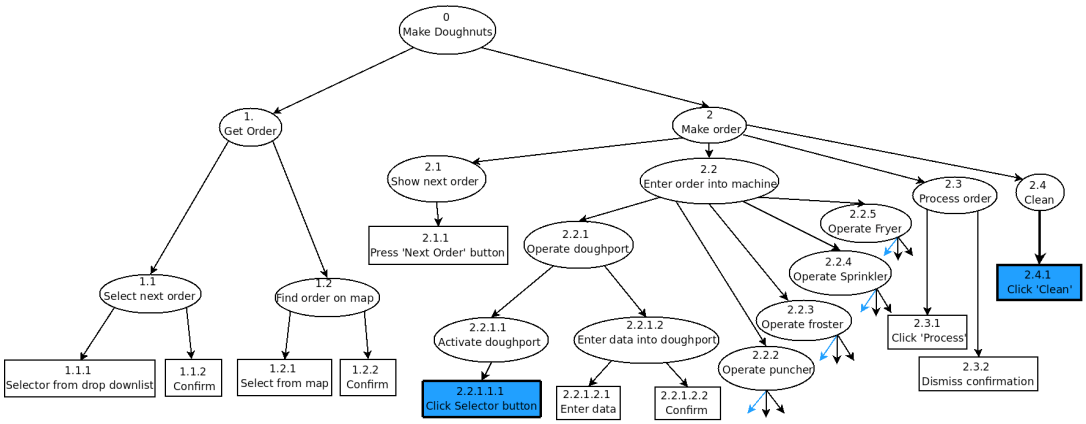

Figure 2: Hierarchical task analysis of the doughnut task. Step 2.2.1.1.1 is the device-initialisation/device-specific step, whereas step 2.4.1 is the post-completion step; both are shaded. Note that the 'Operate Puncher', 'Operate Froster', 'Operate Sprinkler' and 'Operate Fryer' subgoals are not defined further to save space; they are identical in structure to 'Operate Doughport' and as such also contain a device-specific step at the beginning.

and 33 with a mean age of 22.0, and 27 were female. The majority of participants were students, and they were paid £6 for their time.

Materials

The Wicket Doughnut task (Li, 2006), a routine procedural task in which participants have to follow a defined procedure to make virtual doughnuts, was used. Figure 1 shows the components of the doughnut task: the main doughnut interface, the call centre (both developed by Li (2006)), and the live feed (developed for the current study). Figure 2 shows a hierarchical task analysis of the doughnut and call centre tasks. The main task consists of two subtasks (represented as ovals), which are further subdivided into smaller subgoals. The square boxes represent the lowest-level goals and correspond to discrete actions. Device-specific steps are shaded. While only two are shown in the figure to save space, the task contained a total of 6 device-specific steps; the steps that are not shown are the initial selector steps on the Puncher, Froster, Sprinkler and Fryer subtasks.

A trial starts with taking a call at the call centre to get the next order, done on a separate computer terminal. It involves selecting the correct doughnut shop from a list, and finding it on a map. After confirming, the order is then 'transferred' to the Doughnut task interface on another computer terminal.

The main doughnut task consists of five compartments, or widgets, in which participants have to enter information from the order sheet. These need to be operated in the order: Dough Port → Puncher → Froster → Sprinkler → Fryer Before data can be entered, a widget needs to be activated by clicking the appropriate selector button on the selector panel on the right-hand side. Clicking the Ok button then confirms the entry for that widget. Once all widgets have been completed, the order needs to be processed by clicking the 'Process' button. A pop-up screen then indicates the completion of the trial, and the number of doughnuts made. At the end of the trial, the machine must be cleaned by clicking the 'Clean' button. While Li et al. (2008) used interruptions at certain points during the task, the current experiment did not.

To vary working memory load, a monitoring task was added in which participants had to count the number of doughnuts sold in the shops. The Doughnut Live Feed was shown at the bottom of the screen, where occasionally a description of a doughnut was shown. Participants had to attend to a specific characteristic of the doughnut (such as dough type, hole shape or frosting) and keep count of how many with that characteristic were sold. In the low working memory load condition, participants were asked to attend to and keep track of doughnuts with a specific dough type, for instance Crispy. In the high working memory load condition, participants were asked to attend to and separately keep track of doughnuts with a specific dough type and those with a specific hole shape. In both conditions, once a participant had counted 20 doughnuts of

Figure 3: the doughnut live feed. A cycle starts out completely white (a). The background then quickly fades to grey, while the item fades from white to black (b). Halfway through the cycle, the background and the item are at its darkest, and the item is clearly visible (c). At the end of the cycle, the background fades to white again while the item may either stay visible or fade as well (d).

the specified type, they had to click the button on the left of the live feed and start counting from zero again. This allowed the experimenter to assess whether a participant was successfully monitoring the live feed.

To ensure effective monitoring, new items on the live feed did not capture visual attention. This was achieved by using a background that changed from grey to white and back in continuous cycles. Each doughnut description faded in on top of that from white to black, and faded out again after a random number of cycles. Figure 3 shows the progression through one cycle. Each cycle took three seconds, and items remained visible for between 2 and 4 cycles. This randomness made it impossible for participants to predict when a new doughnut description would be shown. The monitoring task and primary tasks were carried out simultaneously.

A number of device-specific steps were present in the doughnut task. Selecting the first compartment, the dough port, was a device-initialisation step. The other selecting steps were counted as other device-specific steps. The last step in the procedure, cleaning the machine, was a post-completion step. A false completion signal was given in the form of a pop-up screen indicating that the doughnuts were ready. In addition, a flashing message notifying the participant of the next call provided a competing signal for the post-completion step. After dismissing this pop-up, the post-completion step took place.

Two separate computer terminals were used; one for the call centre and one for the doughnut making task and live feed. Both screens were operating at a resolution of 1280 x 1024 pixels.

Design

A mixed design was used, with two levels for each independent variable. The first independent variable was working memory load; this was varied between participants. This variable had two levels: low load and high load. The second independent variable was the type of step; this was varied within participant. This variable had two main levels, device-specific and task-specific.

The dependent variable was the error rate. Errors were counted systematically according to the required steps. An error is defined as any action that deviates from the required action at a certain step. To ensure only inappropriate actions are counted and not each individual inappropriate click, only one error could be made on each step.

Procedure

Participants carried out the experiment individually. During the training phase, participants were given an instruction sheet that explained in detail what their task was, and all the procedures necessary to complete the task. After reading the instruction sheet, they observed the experimenter doing the task once, after which they were allowed to practice it twice. Any errors made during the training trials were pointed out immediately using the default Windows XP notification sound and were required to be corrected before the participant was allowed to move on. After each practice trial, the experimenter asked the participant how many doughnuts they had counted on the live feed, and encouraged more accurate performance if necessary.

Participants were instructed to complete the doughnut task as quickly and as accurately as possible. A timer was displayed on the screen throughout the experiment to encourage swift performance; it was reset after each trial. After processing the doughnuts, a pop-up screen notified the participant of the number of doughnuts made. Participants were also told to count the doughnuts in the live feed as accurately as possible; this was further encouraged by the '20 doughnuts' button. Participants were not aware that errors were being studied.

During the experimental phase, the participants completed 11 trials, with the opportunity of a short break after 6 trials. Any errors were pointed out immediately and had to be corrected before the participant was allowed to carry on. The total duration of the experiment was approximately 60 minutes.

Results

Data from 12 participants was excluded from the analysis. The reasons for excluding participants varied. Three participants were excluded because they failed to follow the instructions to monitor the live feed correctly. One participant's data sheet was lost. Eight participants were excluded because they made omission errors at any step on more than 65% of trials. The reason for excluding these error-prone participants is that such high error rates likely

Type of Step	Error count (Opportunity)	Mean error rate (SD), in %
Total	292 (5852)	4.99 (2.51)
Task-specific	57 (4004)	1.42 (0.96)
Device-specific	235 (1848)	12.7 (7.44)
Device-initialisation	84 (308)	27.27 (20.55)
Post-completion	66 (308)	21.43 (21.60)
Other device-specific	85 (1232)	6.90 (6.47)

Table 1: Total error counts and mean error rates across all participants and conditions for the different types of steps.

indicate that the participant has not correctly learnt how to perform the task. We present analysis of error-rate for the remaining twenty-eight participants.

Due to the failure of so many participants to perform the task to criterion, we first examine whether error rate decreased as participants gained more experience at performing the task. There was no evidence of a learning effect over consecutive trials; that is, there was no relationship between number of errors per trial and trial number (τ = -0.26, p = 0.27). This suggests that those included in the analysis had been effectively trained before conducting the study.

We were primarily interested in error rates at device-specific and task-specific steps. Error rates were calculated for each participant for the relevant step types. Only one error was possible on each of the steps. Step 19 (dismissing the pop-up screen) was removed from further discussion, because no error was possible on this step, since the pop-up screen blocked action on the main screen. Thus, a total of 19 errors could be made on a single trial. Each participant did 11 trials, and data from 28 participants was analysed, giving a total opportunity for errors of 19 x 11 x 28 = 5852. Across all participants, a total of 292 errors were made, giving an overall error rate of 4.99%.

It was hypothesised that error rates were higher on device- than on task-specific steps. Table 1 shows the average error rates across all participants on the different types of steps. A repeated-measures ANOVA, comparing error rates on task-specific, device-initialisation, post-completion and other device-specific steps, showed a significant difference between the types of steps, F(3,81) = 19.46, p = 0.000, with Greenhouse-Geisser correction. A post-hoc comparison showed that task-specific steps had significantly lower error rates than all device-specific steps. Looking more specifically at the different types of device-specific steps, it becomes clear that the error rates on DI and PC steps are higher than on the other steps. Post-hoc tests confirm that PC and DI steps have significantly higher error rates than both task-specific steps and other device-specific steps, although there is no significant difference between PC and DI steps.

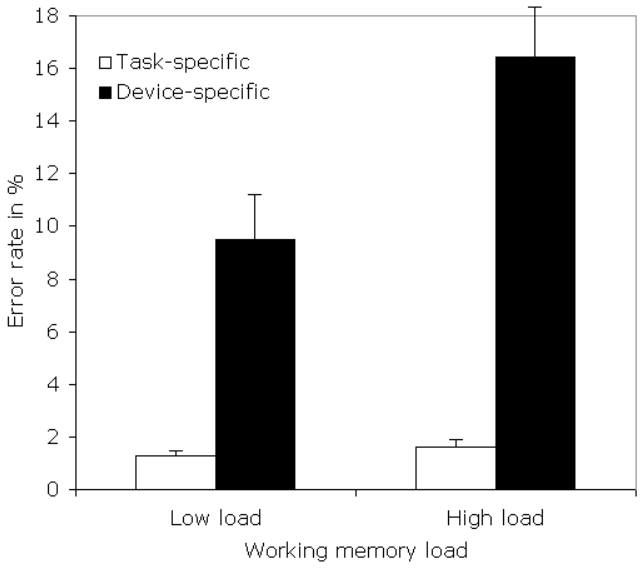

Figure 4: Error rates across working memory load and type of step conditions. Error bars represent the standard error of the mean.

Working memory load was also manipulated on two levels, low load and high load. Figure 4 shows the error rates on the different working memory load levels, for both device- and task-specific steps. Error rates on task-specific steps remained stable across all conditions, while error rates on device-specific steps increased under high working memory load. A 2 x 2 mixed-design ANOVA with type of step as the within-subjects variable and working memory load as the between-subjects variable revealed a main effect of working memory load, F(1,26) = 8.10, p = 0.009. An interaction effect was also found, F(1,26) = 6.68, p = 0.016. A main effect of type of step was also found to be significant, F(1,26) = 81.90, p = 0.000. Simple effects analysis showed that there was no simple effect of working memory load on task-specific steps, F(1,26) = 0.95, p = 0.339. There was a simple effect of working memory load on device-specific steps, F(1,26) = 7.53, p = 0.011.

Discussion

The current experiment investigated the hypothesis that error rates on device-specific steps are higher than on task-specific steps, and that working memory load has a differential influence on them. The results of this study show that the error rates observed at device-specific steps is greater than the error rates observed at task-specific steps. Also, a high working memory load resulted in higher error rates overall. In addition, an interaction effect of working memory load and type of step was found. This supports our predictions.

It can be argued that the finding that error rates are higher on device-specific than on task-specific steps is mainly due to the high error rates on device-initialisation and post-completion steps. However, it should be noted that the error rate on the 'other device-specific steps' was also found to be higher than that on task-specific steps. This indicates that device-specific steps are indeed associated with higher error rates than task-specific steps. Nevertheless, the relatively high error rates on the PC and DI steps may indicate that other factors play a role as well.

Byrne and Bovair (1997) found that only low-capacity individuals were affected by a high working memory load. Although we did not administer working memory capacity tests to participants, the fact that working memory load had a significant effect without dividing participants into low and high capacity groups suggests that this is unlikely to have adversely affected the results.

As expected, working memory load increases the overall error rates. The significant interaction indicated that this

effect is much stronger on device-specific than on task-specific steps. This confirms our predictions.

The current work has implications for theoretical models of error. We hypothesised that device-specific steps have lower activation levels, and are therefore more likely to fall below the interference level. The higher error rates on device-specific steps are in line with this explanation. In addition, the differential influence of working memory load on the two types of steps further supports our theory. It is not clear how the *memory-for-goals* model would account for the lower activation on device-specific steps, highlighting a possible limitation of the model.

Apart from higher error rates and a greater influence of working memory load, these lower activation levels make a number of further predictions. First, reaction times should be longer on device-specific steps. A lower activation level on such steps means that more time is needed for the activation level to increase above the interference level, in order to execute the associated step. Due to the nature of the steps within the doughnut task, it is not appropriate to conduct this analysis on the data from the experiment reported in this paper. Future studies should use a more suitable task to investigate the differences in reaction times on device- and task-specific steps.

Second, device-specific errors should be qualitatively different from task-specific errors. It is more difficult for device-specific steps to overcome the interference level, making it more likely that the step's activation inadvertently falls below the interference level. When this happens, it is likely that the next step has the highest activation level and directs behaviour: an omission error occurs. On the other hand, the higher activation levels on task-specific steps make it less likely that the step accidentally falls below the interference level. Instead, other errors such as incorrect sequence errors (i.e. performing a different task-specific step that is out of sequence) may be more common.

The current work also has implications for the design of interactive systems by going beyond the well-studied PCE. While PC steps are relatively rare, device-specific steps occur on many devices. The current results have demonstrated that device-specific steps are more prone to errors than their task-specific counterparts, and therefore these steps should be avoided in task design where possible.

Conclusion

The current study demonstrated that people are more likely to make errors on device-specific steps than on task-specific steps, providing support for the claim that this distinction can be effective in explaining observed error patterns. Moreover, working memory load was found to have a greater effect on device-specific error rates than on task-specific ones, providing support for our hypothesis that device-specific steps have lower activation levels. Future studies can look more closely at the mechanisms underlying device- and task-specific steps, and investigate how these can lead to different activation levels.

Acknowledgements

This work is supported by an EPSRC DTA studentship. We thank Simon Li for the use of his Doughnut Task.

References

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

Ament, M. G. A., Blandford, A., & Cox, A. L. (2009). Different cognitive mechanisms account for different types of procedural steps. In N. A. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31th Annual Conference of the Cognitive Science Society* (p. 2170-2175).

Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science, 21(1)*, 31-61.

Byrne, M. D., & Davis, (2006) Task structure and postcompletion error in the execution of a routine procedure. *Human Factors, 48,* 627-638.

Chung, P. H., & Byrne, M. D. (2004). Visual cues to reduce errors in a routine procedural task. In *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society*.

Chung, P. H., & Byrne, M. D. (2008). Cue effectiveness in mitigating postcompletion errors in a routine procedural task. *International Journal of Human-Computer Studies, 66,* 217-232.

Cox, A. L., & Young, R. M. (2000). Device-oriented and task-oriented exploratory learning of interactive devices. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the Third International Conference on Cognitive Modelling* (p. 70-77). Veenendaal, The Netherlands: Universal Press.

Gray, W. D. (2000). The nature and processing of errors in interactive behaviour. *Cognitive Science, 24(2)*, 205248.

Hiltz, Back & Blandford (2010) The roles of conceptual device models and user goals in avoiding device initialization errors. *Interacting with Computers*. DOI http://dx.doi.org/10.1016/j.intcom.2010.01.001.

Kirschenbaum, S. S., Gray, W. D., Ehret, B. D., & Miller, S. L. (1996). When using the tool interferes with doing the task. In *Proceedings of CHI '96*. Vancouver, Canada.

Li, S. Y.-W. (2006). *An empirical investigation of post-completion error: A cognitive perspective*. Unpublished doctoral dissertation, Department of Psychology, UCL.

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(4)*, 314-328.

Young, R.M. (1983) Surrogates and Mappings: Two Kinds of Conceptual Models for Interactive Devices. In Gentner, D. and Stevens, A.L. (Eds.), *Mental Models*. Lawrence Erlbaum Associates Inc., pp 35-52.