Making a Task Difficult: Evidence that Device-Oriented Steps are Effortful and Error-Prone

Maartje G. A. Ament, Anna L. Cox, Ann Blandford, and Duncan P. Brumby

University College London

Author Note

Maartje G. A. Ament, Anna L. Cox, Ann Blandford and Duncan P. Brumby, UCL Interaction Centre, University College London.

Maartje G. A. Ament is now at Imperial College School of Medicine.

This research was supported by an EPSRC DTA studentship and EPSRC Grant EP/G059063/1. We thank Jonathan Back, Paul Cairns & Sandy Gould for useful comments on this work. This paper has been much improved by the suggestions from Wendy Rogers, Mike Byrne and three anonymous reviewers.

Correspondence concerning this article should be addressed to Anna Cox, UCL Interaction Centre, MPEB 8th Floor, UCL, Gower Street, London WC1E 6BT. Phone: +44 (0)20 7679 0687 Fax: +44 20 7387 1397 Email: anna.cox@ucl.ac.uk

A preliminary report of Experiment 1 was presented as Student Research Competition paper at the ACM SIGCHI conference, CHI (Ament, 2011a).

Abstract

Errors in the execution of procedural tasks can have severe consequences. Attempts to ameliorate these slip errors through increased training and motivation have been shown to be ineffective. Instead, we identify the steps in a task procedure on which errors are most likely to occur, so that these might be designed out of the task procedure in the first place. Specifically, we consider whether device-oriented steps (i.e., steps in the task procedure that do not directly contribute to the achievement of the task goal) are more error-prone than task-oriented steps (i.e., steps that do directly contribute to the task goal). Two experiments are reported in which participants were trained to perform a novel procedural task. Across conditions, we manipulated the extent to which each step in the task procedure appeared to contribute to the achievement of the task goal (i.e., alternating the assignment of a task step between device- and task-oriented), while keeping the interface and underlying task procedure the same. Results show that participants made more errors and took longer to complete a task step when it played a device-oriented role rather than a task-orientated role. These effects were exacerbated by the introduction of a secondary task designed to increase working memory load, suggesting that when a task step plays a device-oriented role it is more weakly represented in memory. We conclude that device-oriented task steps are inherently problematic and should be avoided where possible in the design of task procedures.

Keywords: error, memory, task performance

Making a Task Difficult: Evidence that Device-Oriented Steps are Effortful and Error-Prone

Many interactive tasks require the user to execute a sequential procedure to achieve their goal. In a typical day, a person will perform many of these tasks. For example, an alarm clock might be set to wake the user in the morning, a train ticket might be purchased at an interactive ticket machine, and a coffee may be paid for via a chip-and-pin machine. Most of the time, these tasks are performed easily and without issue, but on rare occasions errors are made. Such errors can consist of the execution of an incorrect step, such as selecting the incorrect ticket from the machine, or the omission of a step, such as forgetting to select the correct mode after setting the wake time on the alarm clock. In many settings these errors are annoying and disruptive, but can be easily fixed. However, in safety-critical situations, such as healthcare or nuclear power plant operation, errors can have far more severe consequences (Casey, 1998, 2006).

Given the potentially severe consequences of error in the execution of procedural tasks, an important goal for researchers within the areas of Human-Computer Interaction (HCI) and Human Factors has been to understand when people are most likely to make errors, so that these situations can be avoided or designed out (Reason, 1990). It is known that some errors, known as mistakes, are caused by a lack of relevant task knowledge, and can be mitigated by improved training (Reason, 1990). In contrast, slip errors cannot easily be mitigated by improved training and occur even when a user is well practiced at performing a task. Slips are therefore not caused by a lack of relevant task knowledge, but instead reflect unintentional failures in the motor system to execute an action (Norman, 1981) or failures in the working memory system to accurately keep track of progress in the task (Byrne & Bovair, 1997; Gray, 2000).

A number of recent studies have tried to reduce the occurrence of slip errors in the execution of procedural tasks (e.g. Hiltz, Back & Blandford, 2010; Ratwani et al., 2008). Several attempts to mitigate slips have been shown to be ineffective, such as retraining (Byrne & Davis, 2006) or increasing motivation (Back, Cheng, Dann, Curzon & Blandford, 2006), while others, such as visual cues, must be highly aggressive to have an effect (Chung & Byrne, 2008; Jones, Gould & Cox, 2012). Rather than designing an intervention with the aim of mitigating error, we identify on which steps in a task procedure errors are most likely to occur so that these might be designed out of the task procedure in the first place. An example of the potential success for this approach comes from Zimmerman and Bridger (2000), who found that redesigning ATM machines to return the bank card *before* the cash is dispensed drastically reduced the number of bank cards that were accidently left behind at ATM machines.

Are some steps in routine procedural tasks more prone to slip errors?

Previous research has shown that people are systematically more likely to make errors on certain steps in a task procedure than others (e.g., postcompletion errors, initialisation errors, and mode errors). We review these commonly observed errors and consider whether there is a common underlying reason for their occurrence.

A postcompletion error (Byrne & Bovair, 1997), sometimes referred to as a termination error (Thimbleby, 1990) or an omitted secondary sub-goal (Young, 1994), is an error that occurs at the end of a task procedure. Typical examples include forgetting to collect the master copy from a photocopier machine after making a set of copies, or forgetting to collect change after using a vending machine. They occur on a step that is required by the device *after* the user has completed the main task goal. It is generally assumed that the completion of the goal (e.g., obtaining the copied document or receiving the drink from the vending machine) acts as a false completion signal to the user, suggesting that there are no more actions to complete.

The unselected window problem (USW) (e.g., Reichman 1986; Rieman, Byrne & Polson 1994), in which computer users forget to select a window before attempting to interact with it, is also a well-recognised problem in the HCI literature on human error. Young (1994, p.10) argued that "[…] this USW scenario is very similar to, or indeed is simply a specific case of, a class of errors known as "omitted secondary subgoals", or what Byrne [M. D. Byrne, personal communication] calls “post-completion [sic] errors”.”

In fact, the USW error can, perhaps more convincingly, be conceived of as an initialisation error (i.e. an error that occurs when a user fails to perform a step required at the *beginning* of a procedure), as reselecting the original window can be considered to be the *first* step required for achieving the subgoal of entering the information into the window. In a similar way, mode errors (e.g., Sellen, Kurtenback & Buxton 1990), in which users forget to change a system to the correct mode before interacting with it, can also be thought of as a form of initialisation error.

A number of studies have been designed to examine these errors. For instance, Gray (2000) had participants program a VCR and found that almost 50% of the errors made by participants were either mode or postcompletion errors. Such errors are also found in experimental tasks used in human error research. Li, Blandford, Cairns and Young (2008) and Hiltz, Back and Blandford (2010) commented that participants working on Li’s (2006) Doughnut task (a form of routine data-entry task devised for the purposes of studying errors in the laboratory) were more likely to make errors on initialisation and postcompletion steps in the task procedure. So while errors have been documented on a number of steps there is, as yet, little empirical evidence to explain the underlying causes of these errors.

We contend that a contributory factor for some steps in a task procedure being more error-prone than others is that they appear to be less relevant to the achievement of the main goal than others. We define those steps that are more concerned with the operation of the device rather than with achieving the task goal as device-oriented steps. These are steps that are generally specific to a particular device and may not be necessary on all systems. Examples of device-oriented steps include activating a text entry box before entering text, or selecting a window before interacting with it. This contrasts with task-oriented steps, which are steps more concerned with the achievement of the main goal and directly contribute to the completion of the task. Examples of task-oriented steps include entering text into a text box, or placing an original on a copy machine’s glass plate before making copies.

The distinction between device- and task-oriented steps in routine procedural tasks is not new. For instance, Young (1981, 1983) discusses several different types of mental models, including device and task models. Device models describe knowledge about how a device works (see also Cox & Young, 2000; Kieras & Bovair, 1984). Young argues if the mapping between the device and task model is simple and straightforward, then the device will be easy to use, because its operation fits with the user’s basic understanding of the steps involved in completing the task. In contrast, steps required by the device that do not correspond to the user’s task model will be more problematic and difficult to learn. In a similar fashion Kirschenbaum, Gray, Ehret and Miller (1996) make a distinction between ‘tool-only’ and ‘task-tool’ steps in which tool-only steps are defined as “operations that have no direct relationship to accomplishing the task” but that are more concerned with “fiddling with the tool” (Kirschenbaum et al., 1996, p203). These tool-only steps can be thoughts of as device-oriented steps. In contrast, task-tool steps work towards the goal of the task, and as such can be seen as task-oriented steps. Kirschenbaum et al. (1996) hypothesise that the more tool-only steps in a task, the worse the usability of the system, as grappling with using the tool takes the focus of the user’s attention and effort away from completing the task. However, neither the account given by Young, nor that of Kirschenbaum et al, is based on empirical evidence.

In summary, a distinction can be made between task-oriented steps, which are steps in a task procedure that contribute directly to the achievement of the user’s goal, and device-oriented steps, which are steps in a task procedure that do not directly contribute to the achievement of the user’s goal. This distinction has been widely discussed in the literature and there is a common assumption that device-oriented steps should be more problematic because they are idiosyncratic to the particular device being used. There is however limited empirical evidence to directly support this assumption.

Overview of the Experiments

As outlined above there has been a focus in the literature on errors made on initialisation and also postcompletion steps. It can be argued that these steps are instances of device-oriented steps because neither of them contributes directly to the user's main goal (i.e., the postcompletion step is by definition a task step executed after the achievement of the main goal). One of the hypotheses that we sought to test in this paper is that device-oriented steps may be the primary cause of many systematic slip errors in the execution of routine procedural tasks of the type discussed above.

To the best of our knowledge, only two studies have aimed to directly investigate whether device-oriented steps are more error-prone than task-oriented steps (Ament, Blandford & Cox, 2009; Ament, Cox, Blandford & Brumby, 2010). In experiments using Li's (2006) Doughnut task, Ament et al. demonstrate that there are higher error-rates on device-oriented steps. However, these studies are limited in that they did not control for features of the interface or the prior experience that participants had with the task domain. It is possible that the design of the interface elements of the Doughnut task may influence its difficulty. Furthermore, these studies did not control for the prior experience of participants in the domain of the task, which could influence error rates on particular steps. It is likely that participants had a prior mental model of the steps required to perform the Doughnut task. That is, even though the participants had almost certainly never used the specific doughnut machine before, they may well have a basic understanding of how doughnuts are made and the steps involved in the task procedure.

We report two experiments that investigate whether there are performance differences between device- and task-oriented steps. To overcome some of the limitations of previous work described above, we developed a novel procedural task, which we refer to as the Frankenstein task. The Frankenstein task required participants to either make or destroy monsters (see Figure 1 for a screenshot). Regardless of the cover story given to participants, the same task procedure was used to achieve both of these tasks (see Table 1 for a description). However, the relevance of a given step in the task procedure varied depending on whether the participant was making or destroying monsters. In this way, we were able to have the same step in the task procedure play either a device- or a task-oriented role. For instance, a particular button serves to activate a widget in one condition (not relevant to the task goal, so device-oriented) while in the other condition the same button removes a body part from the monster (directly relevant to the goal, so task-oriented). There are 10 such target steps in the task procedure, along with fourteen additional filler steps (see table 1). The aim of this manipulation was to find out whether device-oriented steps would be more effortful and error-prone than task-oriented steps, even though the task and interface remain essentially unchanged.

The Frankenstein task has a number of benefits. First, it allows us to study whether device-oriented steps are more prone to errors than task-oriented ones, while controlling for confounding factors such as differences in the task interface and task sequence. This is achieved by using the same interaction element for both device- and task-oriented steps. In addition, the same task sequence is used across conditions and only the meaning ascribed to each of these task steps is varied between conditions. Second, this task allows us to determine whether errors are due to a difference in goal relevance, or to some steps being more practised than others. It is thought that very few participants will have prior experience in making or destroying monsters (because it is not modelled on a naturalistic task, such as making doughnuts or programming a VCR). As such, we assume that participants will be naïve as to how to perform the Frankenstein task and so will have to learn the steps involved, and that the rate of learning will be equal across these steps. Moreover, training for both conditions is the same, ensuring that both steps in a pair are equally well learnt between conditions. Third, a general issue across a number of studies that have investigated errors in the execution of procedural tasks is that timing data was not reported. For instance, Li's (2006) Doughnut Task did not record any time event data, meaning that we do not know whether participants took more time to execute error-prone steps in the task procedure. In developing the

Frankenstein task we wanted to allow for the investigation of step completion time by controlling for the position of steps on the interface as an influencing factor in the time taken to complete steps.

Given the above task setup, we first consider whether error rates on device-oriented steps are higher than those on task-oriented steps when novelty of task and interface features are held stable. In addition, we also explore the cause of these errors. Our hypothesis is that the difference between device- and task-oriented steps is a distinction in the way that tasks are organised in memory. In particular, we argue that because device-oriented steps do not directly contribute to the user's primary goal, they are more weakly represented in memory (either through weaker encoding or through less frequent rehearsal in memory). Building on Altmann and Trafton (2002) memory for goals theory, we might therefore assume that if this is the case, these weaker memory representations are more difficult to retrieve, and so people forget to execute these task steps. Moreover, the time taken to execute a step is related to the strength of representation in memory: the weaker the representation of the step in memory, the longer it takes for a step to be retrieved (see Altmann & Trafton, 2002). It therefore follows that it should take longer to execute a device-oriented step than a task-oriented step assuming they are equated for motor execution difficulty.

Given the expected role of memory in mediating the likelihood of error, we expect that a manipulation of working memory load should differentially impact performance on the two types of step, as found by Ament et al. (2010). High working memory load conditions make temporary memory failure more likely, potentially increasing the error rate on both step types (as in Byrne & Bovair, 1997). However, if the memory representations of device-oriented steps are weaker than those of task-oriented steps, then we would expect that participants will make more device-oriented errors when they are under high working memory load. In contrast, we expect that the level of working memory demand imposed by the secondary task will not affect the ability to retrieve and execute task-oriented steps.

In summary, the first experiment reported here investigates whether error rates and step times differ on task-oriented and device-oriented steps. The second experiment extends this by considering whether working memory load differentially impacts performance on these different types of steps. We conclude by discussing the implications of these results and whether it is wise for designers to strive to remove device-oriented steps from task procedures to prevent errors.

Experiment 1

Method

Participants. Thirty-two participants (twenty-one female) took part in the experiment i.e. sixteen participants in each of the two between-subjects groups. All were students at UCL, and they were paid £7 for their time and effort. They were aged between 18 and 30 (average 20.6 years).

Materials. The procedure for the Frankenstein task consists of 24 steps. Of these task steps, there are 12 target steps (two of these are repeated twice), which play either a device- or task-oriented role dependent on whether monsters are being made or destroyed. The remaining task steps are fillers and play a consistent role across both conditions. Table 1 shows an overview of the target steps for both conditions, indicating which are device- and which are task-oriented.

A pilot study was conducted to ensure the validity of the classifications of steps as device- and task-oriented. Two independent expert raters (researchers with a track record of publications in this area) and two novice raters (researchers with no prior experience in this area) assessed the classification of steps into device- and task-oriented. We used both novice and expert raters in order to control for prior familiarity with the concepts of device- and task-oriented steps. Raters were given the definition of device- and task-oriented steps, along with a number of examples and a list of common characteristics of the two types of steps. Each rater was tested on a number of 'classic' steps to determine whether they had accurately understood the definitions. Raters were 100% accurate in classifying these classic steps and were therefore allowed to begin the rating procedure. Inter-rater reliability using Cohen's κ statistic was used to measure their agreement. It was found that agreement was substantial on both conditions: κ values were 0.78 and 0.80 for the experts, and 0.59 and 0.60 for the novices (all significant at $p < 0.05$). This means that the classifications used in this experiment are reliable and can be used with confidence. For a more detailed discussion of this pilot study, see Ament (2011b).

Building on the work of Byrne and Bovair (1997), a monitoring task was added to ensure that error rates were at a measurable level. A variant of the well-known n-back task (Kirchner, 1958) was used for this purpose, for two reasons. First, this was highly similar to the working memory loading task used by Byrne and Bovair (1997), though using digits rather than letters. Second, the task is auditory and verbal, and therefore does not directly interfere with the primary task. Digits were read out to participants by the computer at a rate of one digit every three seconds, and they were asked to remember the last three digits at any point in time. A beep sounded after random intervals of between 3-10 digits, indicating that participants had to repeat the last three digits out loud. Participants carried out the Frankenstein task and the monitoring task concurrently.

The Frankenstein task was presented on two adjacent computer screens, attached to the same terminal. Both screens operated at a resolution of 1280x1024; the screen on the left showed the Locator (used for steps 1-5), while the one on the right showed the main Frankenstein task interface (used for steps 6-24). Participants used a mouse to interact with the task, and could move the cursor freely between the two screens. A Tobii X60 eye-tracker was used to record eye movement data; however, this is not reported in this paper.

Design. A mixed design was used. The first independent variable was the role that a step played in the task procedure: device-oriented or task-oriented. This was varied as a within-subjects factor. The second independent variable was the ‘create/destroy’ condition, which was varied between participants. It had two levels, ‘create monster’, and ‘destroy monster’, which are mirror images of one another. That is, if step X is device-oriented in one condition, it is task-oriented in the other, and vice versa. Note that the overall task condition is a dummy variable, as it is used purely for experimental design purposes and has no theoretical meaning. As such, the two levels will further be referred to as ‘condition 1’ and ‘condition 2’. Table 1 indicates which steps are device- and task-oriented in each condition and demonstrates that the target steps measured in the experiment were interspersed throughout the task procedure. In addition, trial number was included as a within-subjects variable, to test whether participants improved with practice.

Measures. Data from only the target steps was measured, as the remaining steps were not manipulated in terms of their relevance to the task goal. Two measures were included. First, the number of slip errors on the target steps was recorded. Errors were counted systematically according to the required steps. A slip 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 mouse-click, only one error could be made on each step. Second, the step time for each step was recorded. The step time is defined as the time from the successful execution of the previous step until the next meaningful mouse-click (i.e. a click on a button or other interaction element on the Frankenstein task, excluding 'repeat' clicks or drags on sliders or multiple clicks on a drop down menu).

Procedure. Each participant carried out the task individually. First, they read an information sheet explaining the monitoring task. Each participant then practised this task by itself, to ensure they had understood the instructions and were able to do the task. Each participant completed three practice runs.

Second, participants viewed a short computer animation that explained the story behind the Frankenstein task and what their task was. They then read the detailed on-screen instructions on how to use the interface to complete their task. Participants were asked to complete the task as quickly and accurately as possible.

After reading the instructions, participants observed the experimenter complete one trial of the task. They were then allowed to explore the task interface without the monitoring task for one trial. After this, participants practised the main task with the monitoring task. If an error was made during training, it was pointed out immediately with a sound and a pop-up screen that explained what had gone wrong and what the participant should do instead. All errors had to be corrected before the task could be resumed.

This level of training is similar to that of Back, Blandford and Curzon (2007) who investigated errors in routine performance and whose experimental participants only executed a small number of training trials. Similarly, Gray's (2000) participants only executed five training trials (plus two familiarization trials), though he claimed to test performance at the beginning of Anderson's (1995) third stage of expertise. To ensure that training had been effective, participants in our study had to complete 3 trials without errors before they were allowed to carry on with the experimental phase.

After training was finished, the eye tracker was calibrated and started before participants completed 11 experimental trials. Errors were pointed out with a sound, but no pop-ups with hints were shown. Again, all errors had to be corrected before participants could resume the task. After completing the experiment, participants were debriefed about the purpose of the experiment.

Results

Error Rates. A total of 218 errors were made on target steps during the experiment. Given a total opportunity for error of 32 participants x 11 trials x 12 target steps = 4224, the overall error rate is 5.16%. Table 2 shows the average overall error rates and standard deviations for task- and device-oriented steps. A 2 x 2 x 11 mixed-design ANOVA was done. The task condition (1 or 2) was the between-subjects variable, while type of step and trial number were within-subject variables. The latter was included to test whether participants were effectively trained before starting the experimental phase. The ANOVA revealed a significant main effect of the type of step, $F(1, 30) = 9.67$, $p = 0.004$, $\eta_p^2 = 0.25$, indicating that error rates were higher at device-oriented than at task-oriented steps. There was no significant main effect of task condition, $p = 0.90$, nor of trial number,

p = 0.60. Also, no interactions between any of the variables were found (type of step x condition: p = 0.43, trial number x condition: p = 0.94, type of step x trial number: p = 0.18 and type of step x condition x trial number: p = 0.09.

Figure 2 shows a further breakdown of the average error rates for each individual target step. As the figure illustrates, there is a trend towards high error rates on individual device-oriented steps; however, a one-way ANOVA comparing the device- and task-oriented versions of each step shows that this effect was significant only for step 3, $F(1, 31) = 6.59$, $p = 0.015$, $\eta_p^2 = 0.18$.

Step Times. Step times were also investigated. As outlined above, this was defined as the time from the successful execution of a previous step until the next meaningful mouse-click. We only include correct cases in this analysis because we assume that errors would reflect longer intervals.. Due to a programming error, the step time on the device-oriented version of step 4 (deactivating the slider) was not accurately recorded; data from this step are excluded (this error only affected the current experiment and was resolved for the second). The data were inspected for outliers, because extremely long step times may not reflect the time it takes for a goal to gain sufficient activation; it is likely that other processes are at work in these cases (e.g. distraction, daydreaming). The outliers were identified in the raw step time data (before averaging), to prevent having to exclude a participant for a single long step time. An outlier was identified as a data point that was more than 3 standard deviations removed from the grand mean. Four such outliers were found, each with a step time of more than 25 seconds; these were removed from the data set.

The mean step times were then computed for each participant. Table 3 shows an overview of the mean step times for device- and task-oriented steps. A 2 x 2 x 11 mixed-design ANOVA was conducted, where task condition was the between-subjects variable and all others were within-subjects variables. There was a significant main effect of type of step, $F(1, 30) = 47.267$, p < 0.001, $\eta_p^2 = 0.63$, indicating that step times on device-oriented steps were longer than on task-oriented steps. There was also a significant main effect of trial number, $F(1, 30) = 3.75$, $p = 0.006$, $\eta_p^2 =$

0.66. No significant main effect was found of task condition, p = 0.30. Moreover, there were no significant interactions between type of step and condition (p = 0.79), type of step and trial number (p = 0.78), condition and trial number (p = 0.19), and type of step, condition and trial number (p = 0.67).

Discussion

The current study investigated whether device- and task-oriented steps differ in terms of their error rates and step times. An experiment that manipulated the type of steps, while controlling for other factors, provided data in support of the hypotheses.

First, it was found that the error rates were significantly higher when steps in the task procedure played a device-oriented role than when they played a task-oriented role. This indicates that the difference between these steps is robust. Moreover, it provides support for the hypothesis that this is due to differences in the relevance of steps to the task goal: the difference in error rates persisted despite all steps being equally well learnt.

Second, the results showed that step times were longer on device-oriented steps. This held even though the task interface was kept the same for each device- and task-oriented step pair. This finding indicates that device-oriented steps really are more difficult to remember. As Li et al. (2008) argue, they may rely on more deliberate mechanisms to be remembered.

While no significant effect of trial number on error rates was found, there was an effect of trial number on step times. This suggests that there was a small learning effect in the data. We interpret this as participants being effectively trained (i.e. the errors are not due to insufficient training), although they have not yet reached asymptote in relation to speed of performance. Overall, the behavioural findings support the hypothesis that device-oriented steps are more problematic than task-oriented steps, and that this is due to their relevance to the task goal rather than them being less well learnt.

This study addressed two of the three predictions set out in the introduction (i.e., that device-oriented steps will be more error prone and have longer step times than task-oriented steps). The third prediction, that working memory load has a disproportionate effect on device-oriented steps, is addressed in the following experiment.

Experiment 2

This study replicates experiment 1 and provides an extension by looking at the effect of working memory load on device- and task-oriented steps. In their study of postcompletion errors, Byrne and Bovair (1997) found that a high working memory load (accomplished by asking participants to keep track of the last three digits read out to them) significantly increased postcompletion error rates, though only in individuals with a relatively low working memory capacity. Byrne and Bovair (1997) explain their findings in terms of the competition between goals in memory. A high working memory load means that there are more goals in memory that compete with the target goal. This makes it more difficult for the target goal to be remembered. Postcompletion errors occur because, after completion of the main goal, the postcompletion subgoal is no longer being primed by the main goal (since it has been completed), and as such is quickly forgotten. This occasionally leads to an error. We have argued above that a postcompletion step is a special case of device-oriented step. If this is the case, we might therefore expect Byrne and Bovair’s findings that working memory load influences error rate to generalise beyond the postcompletion error to all device-oriented steps.

Ament et al. (2010) took a similar approach to Byrne and Bovair, by varying working memory load on the Doughnut task, finding that error rates on device-oriented steps were affected much more than those on task-oriented steps. However, this study did not investigate other important measures, such as step completion time. Moreover, direct comparison of device- and task-oriented steps is problematic in the Doughnut task because it does not control for step difficulty, visual salience of the interface items and task sequence. The current experiment sought to address these concerns by using the Frankenstein task to investigate the effect of working memory load. The experiment investigates the differential effect of working memory load on device- and task-oriented steps, by looking at error rates and step times. It is expected that working memory load will have a disproportionate effect on device-oriented steps, increasing the number of errors and step times.

Method

Participants. Fifty-two new participants (thirty-three female) took part in the study i.e. thirteen participants in each of the four between-subjects groups. They were aged between 18 and 25 years old, with a mean age of 20.3. All participants were students at UCL, and were paid £7 for their time.

Materials. The same Frankenstein task as in experiment 1 was used. The same secondary task was also used, with an additional manipulation that allowed for the variation of working memory load. In the high load condition, participants were asked to repeat the last three digits when the beep sounded, whereas in the low load condition, participants repeated only the last digit. These numbers were determined during a pilot study, which revealed that three digits was the maximum number that participants could reasonably remember without significant deterioration of performance. Note that the high load condition was the same as experiment 1.

Design. A mixed design was used. The first independent variable was working memory load. This was varied between participants, and had two levels: high and low. The second independent variable was the type of step, which was varied within participants, and also had two levels: device-oriented and task-oriented. In addition, the dummy variable of task condition was varied between participants, and had two levels: 1 (create monsters) and 2 (destroy monsters). As in the previous experiment, trial number was included as a within-participants variable. Moreover, participants' prior ability to perform a n-back task (described below) was included as a variable, for which participants were divided into a high and a low ability group, based on median split.

Measures. As in the previous study, the main measures were error rates and step times.

Procedure. Prior to starting the experiment, all participants completed a brief assessment of their ability to do an n-back task in order to rule out individual differences in ability to perform such a monitoring task as an explanation for the predicted findings. A visual variant on the n-back task was used for this. A sequence of virtual objects was shown to participants, and for each they were asked to indicate whether it was the same object as the one they saw n objects ago. Each sequence consisted of 30 + n objects, and participants completed 5 sequences, up to 5-back.

After completing the visual n-back task, the main part of the experiment started. The procedure was identical to that in the previous experiment.

Results

The main dependent variables were error rate and step times. It was expected that working memory load would have a greater influence on the performance of device-oriented steps than task-oriented steps. Data from 52 participants was recorded. Two participants failed to execute the secondary task accurately (one ignored the instructions and the other responded incorrectly on more than 25% of trials); these were excluded from analysis because the contribution to working memory load from the secondary task could not be guaranteed.

Performance on n-back task. The average score on the visual n-back test was 81% (SD 10%, range 51-96%). No differences were found between the high and low load groups (p = 0.102) or between the task conditions (p = 0.56).

Error Rates. A total of 385 errors were made during the experiment. Given a total opportunity for error of 50 participants x 11 trials x 12 target steps = 6600, the overall error rate is 5.83%. Figure 3 shows the error rates on device- and task-oriented steps under low and high working memory load. Comparing experiment 1 with the high-load condition of experiment 2, there was no significant difference, Z = 1.327, p = 0.059.

A mixed-design ANOVA was conducted, with the type of step and trial number as the within-subjects variables. Working memory load, task condition and prior performance on the working memory loadn-back task were the between-subjects variables. This resulted in a 2 x 11 x 2 x 2 x 2 design. This revealed a significant main effect of the type of step, with device-oriented steps giving rise to higher error rates than task-oriented steps, $F(1, 41) = 26.09$, $p < 0.001$, $\eta_p^2 = 0.39$. There was also a significant main effect of working memory load, $F(1, 41) = 12.56$, $p = 0.001$, $\eta_p^2 = 0.23$, with higher error rates under a high load. Moreover, there was a significant interaction between the two dimensions, $F(1, 41) = 6.77$, $p = 0.013$, $\eta_p^2 = 0.14$. Simple effects analysis showed that under low load, there was no difference in error rates between device- and task-oriented steps, $p = 0.073$. However, under high load, device-oriented steps gave rise to higher error rates than task-oriented steps, $F(1, 47) = 11.41$, $p = 0.002$, $\eta_p^2 = 0.22$.

There were no significant effects of condition, $p = 0.11$, or prior performance on working n-back task, $p = 0.37$. In contrast, there was an effect of trial number, $F(10, 410) = 2.47$, $p = 0.007$, $\eta_p^2 = 0.06$, showing an overall decrease in error rates with increased trial number. No further interactions between any of the factors were found.

Step Times. Similar effects of working memory load and type of step were found on the step time data. Figure 4 shows an overview of these data. A mixed-design ANOVA revealed a significant effect of type of step, $F(1, 41) = 272.99$, $p < 0.001$, $\eta_p^2 = 0.87$, with longer step times on device-oriented steps. There was also a significant main effect of working memory load, with longer step times under a high load, $F(1, 41) = 18.30$, $p < 0.001$, $\eta_p^2 = 0.31$. There was also a significant interaction between the two dimensions, $F(1, 41) = 10.46$, $p = 0.002$, $\eta_p^2 = 0.20$. Simple effects analysis revealed that step times were significantly longer on device-oriented steps than on task-oriented steps, both in the low load condition, $F(1, 47) = 7.97$, $p = 0.007$, $\eta_p = 0.16$, and in the high load condition, $F(1, 47) = 24.33$, $p < 0.001$, $\eta_p^2 = 0.37$.

No effect of prior performance on working the n-back task was found, $p = 0.25$, although there were main effects of task condition, $F(1, 41) = 9.43$, $p = 0.004$, $\eta_p^2 = 0.19$, and of trial number, $F(10, 410) = 6.88$, $p < 0.000$, $\eta_p^2 = 0.14$. Figure 5 shows which pairwise comparisons are significant, and demonstrates the general direction of this finding.

Discussion

The current study investigated the effect of working memory load on device- and task-oriented steps. It was hypothesised that a high working memory load would have a disproportionate effect on device-oriented steps, leading to a larger increase in error rate and step time than on task-oriented steps. The results of the study support these predictions, in that, error rates were higher on device-oriented steps, and an interaction between the type of step and working memory load was also found. This indicates that the working memory load manipulation had a larger effect on device-oriented steps than on task-oriented steps. A highly similar pattern was found for step time data, with device-oriented steps taking longer than task-oriented ones. There was also an interaction between step type and working memory load. This interaction shows that while there was a difference between high and low working memory load for both device- and task-oriented steps, the effect of working memory load was more pronounced for device-oriented steps. Again, this indicates that working memory load had a stronger effect on device-oriented steps.

In contrast to experiment 1, a small learning effect was found on both error rates and step times. This means that a proportion of the errors made and longer step times might be explained by participants still learning the task. However, since the effect sizes were small, this is unlikely to play a major role in the current results. Nevertheless, in future experiments, it may be beneficial to increase the length of the training phase, or use a different indicator to determine whether participants have been effectively trained.

The monitoring task used to vary working memory load was highly similar to the one used by Byrne and Bovair (1997). They found that a high load (accomplished by asking participants to repeat the last three digits read out to them) significantly increased postcompletion error rates, though only in individuals with a relatively low working memory capacity. This is in line with the findings of the current study, which shows that these results also extend to slip errors beyond the postcompletion error. A difference is that the current study found an effect of working memory load in all participants, not only those who performed better on the initial test of ability to perform well in the secondary task.

General Discussion

Errors made in the completion of routine procedural tasks are infrequent but persistent. Some steps in a procedure, such as mode and postcompletion steps, are known to be more error prone than others. We argue that these steps are special cases of a broader category of steps we call device-oriented steps. Our results show that error rates were higher and step times were longer on device-oriented steps than on task-oriented ones. This occurred irrespective of interface factors, such as the size of, and distances between, interaction elements. In addition, working memory load was found to affect device-oriented steps more severely than task-oriented ones. As such, the current work demonstrates that when working memory load is increased those steps that do not contribute to the task goal suffer most.

An important issue is *why* device-oriented steps should be more error prone than task-oriented steps. We hypothesise that it is because they are more weakly represented in memory. One possible explanation for the weaker representations that we can immediately rule out is that device-oriented steps were in some way less well learnt than the task-oriented steps. This is because the Frankenstein task consists of two equally well-learnt procedures, while only the type of step was manipulated. Yet the difference in error rates persisted. While this does not rule out that a learning effect plays a role in tasks performed in other settings or contexts, it is not the main cause of the differences found between device- and task-oriented steps in the experiments reported here.

An alternative explanation then is that it is a consequence of how knowledge of the task and the device is represented in memory. In the introduction we outlined two types of mental model: task models that describe knowledge concerning how to do a task, and device models that describe knowledge about how to operate a particular system or interface. The discrepancy between the two mental models (as discussed by Young, 1981, for instance) forms the basis of the explanation proposed. The task model represents all the steps that provide a direct contribution towards the goal. The device model, on the other hand, represents all the steps necessary to operate the device. This means that those steps that are represented only in the device model will be more weakly represented in memory than those steps that are present in both models. As a result device-oriented steps are not adequately primed in memory and it is this that leads to the increase in errors and slower step times observed here.

Finally, we give some consideration to the generalizability of our findings to different settings and tasks, and their practical implications. First, the knowledge gained from the studies reported here can be used to mitigate errors in routine procedural tasks. Specifically, the findings suggest that steps that are less relevant to the task goal are inherently problematic, and should therefore be avoided as far as possible in the design of task procedures. Moreover, the results demonstrated that this is especially the case in environments in which operators are subject to a high working memory load. Devices that are operated in busy environments in which people must keep track of multiple plans and activities should, where possible, be free of device-oriented steps.

Future Research

Future research should investigate whether designing out steps is indeed effective in reducing errors. There is some experimental evidence that this approach can be extremely effective for the postcompletion error. Byrne and Davis (2006) found a complete elimination of postcompletion errors when the task structure was resolved.

An important consideration is when and how device-oriented steps can be designed out. Many different constraints affect the design of devices, and it is not feasible to eliminate all device-oriented steps. In this case, it is important to also investigate how errors on these steps can be mitigated. One approach would be to use just-in-time cues to remind users to perform a required step. Indeed, these have been shown to be effective in mitigating postcompletion errors in circumstances where it is possible to predict the next step reliably (Chung & Byrne, 2008; Ratwani et al., 2008). Another approach could be to use forcing functions on device-oriented steps in order to prevent the user from progressing within the task sequence until the device-oriented step had been completed. This approach can be used when the user has to make a choice between modes of operation of a device (if modes cannot be avoided). Rather than trying to minimise the chance of error, other approaches might attempt to mitigate the effects of an error. For example, measures that involve adjusting the task context (such as placing a system in a secure environment), or re-allocating device-oriented steps to the device rather than to the user (so that user is logged out automatically after a period of inactivity), could minimise the risk of unauthorised use in the case of a user forgetting to log out (see Blandford 2000 for further discussion). Future work should investigate whether these approaches can be effectively implemented across a range of device-oriented steps.

Limitations

A final issue that should be discussed concerns the level of training that our participants had in working on the Frankenstein task. The current work is concerned with routine procedural tasks. This implies that the person doing the task is relatively well practiced at it and can perform it to a reasonable level of competence. In our experiments we aimed to train participants up to a level of competence at the task given the limited time for training that was available. To test whether this issue affected performance on the Frankenstein task, the current work investigated participants' performance over the course of the experiment. We might expect that with practice, participants would make fewer errors but at some point their performance would become stable. A similar approach was taken by Byrne and Bovair (1997), who compared error rates over consecutive trials to assess whether participants had acquired sufficient skill. This was also done in the current work, which revealed a small learning effect on errors and step times. Therefore, it cannot be ruled out that participants were still perfecting their skills while executing the experiment.

Monk (1986) has argued that it might not always be necessary or even desirable to train participants very extensively. Experimental tasks are generally far simpler than those involved in operating complex systems such as an airplane cockpit or nuclear power plant control room. Monk (1986) argued that, for his experimental task, much less training was needed for participants to reach a level of expertise comparable to that of, for instance, nuclear operators. While this argument may not apply to simpler safety-critical equipment such as infusion pumps used in hospitals, it highlights the importance of properly defining the required expertise level.

Conclusion

In conclusion, errors in routine procedures do not occur at random: they are more frequent on some steps in a task than on others. The results of the current work provide support for the argument that devices should allow users to focus on managing the mission, not the machine, by designing devices with as few device-oriented steps as possible, particularly in situations with a high working memory load.

References

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

Ament, M. G. A. (2011a). Frankenstein and human error: Device-oriented steps are more problematic than task-oriented ones. In *CHI '11 Extended Abstracts on Human Factors in Computing System.* ACM, New York, NY, USA, 905-910.

Ament, M. G. A. (2011b). *The role of goal relevance in the occurrence of systematic slip errors in routine procedural tasks.* PhD thesis, University College London, Department of Computer Science.

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

Ament, M. G. A., Cox, A. L., Blandford, A., & Brumby, D. (2010). Working memory load affects device-specific but not task-specific error rates. In S. Ohlsson and R. Catrambone (Eds.), *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, Portland, OR, pp. 91–96.

Anderson, J. R. (1995). *Cognitive psychology and its implications* (4th ed.). New York, NY: W. H. Freeman.

Back, J., Blandford, A., & Curzon, P. (2007). Slip errors and cue salience. In W. Brinkman, D. Ham & B. L. W. Wong (Eds.), *Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore!*, London, United Kingdom, pp. 221–224.

Back, J., Cheng, W. L., Dann, R., Curzon, P., & Blandford, A. (2006). Does being motivated to avoid procedural errors influence their systematicity? In *Proceedings of HCI '06,* Vol. 1, pp. 2–9.

Blandford, A. (2000) Designing to avoid post-completion errors. PUMA Working Paper WP33.

Available from http://www.eis.mdx.ac.uk/puma/wp33.pdf (accessed 19th May 2013).

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, E. M. (2006). Task structure and postcompletion error in the execution of a routine procedure. *Human Factors, 48(4),* 627–638.

Casey, S. M. (1998). *Set phasers on stun: and other true tales of design, technology, and human error.* Santa Barbara, CA: Aegean Publishing Company.

Casey, S. M. (2006). *The atomic chef: and other true tales of design, technology, and human error.* Santa Barbara, CA: Aegean Publishing Company.

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 and J. Aasman (Eds.), Proceedings of the Third International Conference on Cognitive Modelling,* Universal Press, Veenendaal, the Netherlands, pp.70–77.

Gray, W. D. (2000). The nature and processing of errors in interactive behaviour. *Cognitive Science, 24(2),* 205–248.

Hiltz, K., Back, J., & Blandford, A. (2010). The roles of conceptual device models and user goals in avoiding device initialization errors. *Interacting with Computers, 22(5),* 363–374.

Jones, S. A., Gould, S. J. J., & Cox, A. L. (2012). Snookered by an interruption?: Use a cue. In *Proceedings of the 26th Annual BCS Interaction Specialist Group Conference on People and Computers (BCS-HCI '12).* British Computer Society, Swinton, UK, 251-256.

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science, 8*, 255–273.

Kirchner, W. K. (1958). Age differences in short-term retention of rapidly changing information.

Journal of Experimental Psychology 55(4), 352–358.

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, pp.203–204.

Li, S. Y.-W. (2006). *An empirical investigation of post-completion error: a cognitive perspective.* PhD thesis, University College London, Department of Psychology.

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.

Monk, A. (1986). Mode errors: A user-centred analysis and some preventative measures using keying-contingent sound. *International Journal of Man-Machine Studies, 24,* 313–327.

Norman, D. A. (1981). Categorization of action slips. *Psychological Review, 88(1),* 1-15.

Ratwani, R. M., McCurry, J. M., & Trafton, J. G. (2008). Predicting postcompletion errors using eye movements. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08).* ACM, New York, NY, USA, pp. 539–542.

Reason, J. (1990). *Human error*. Cambridge, UK: Cambridge University Press.

Reichman, R. (1986). Communication paradigms for a window system. In D. A. Norman & S. W. Draper (Eds.): *User Centred Systems Design: New Perspectives on Human-Computer Interaction* (pp. 285-313), Lawrence Erlbaum Associates.

Rieman, J., Byrne, M. D., & Polson, P. G. (1994). Goal formation and the unselected window problem. In *CHI' 94 Research Symposium*, Boston, MA.

Sellen, A.J., Kurtenback, G.P., & Buxton, W.A.S. (1990). The role of visual feedback and kinesthetic feedback in the prevention of mode errors. In *D. Diaper, D. Gilmore, G. Cockton & B. Shackel (Eds.), Human-Computer Interaction – INTERACT '90,* Elsevier Science, the Netherlands, pp. 667-673.

Thimbleby, H. (1990). *User interface design*. ACM Press Frontier Series, Addison-Wesley.

Young, R. M. (1981). The machine inside the machine: Users' models of pocket calculators. *International Journal of Man-Machine Studies, 15(1),* 51–85.

Young, R. M. (1983). Surrogates and mappings: Two kinds of conceptual models for inter- active devices. In D. Gentner & A. L. Stevens (Eds.), *Mental Models*, Lawrence Erlbaum Associates, pp. 37–52.

Young, R. M. (1994). The unselected window scenario: Analysis based on the Soar cognitive architecture. In CHI '94 Research Symposium, 9-11.

Zimmerman, C. M., & Bridger, R. S. (2000). Effects of dialogue design on automatic teller machine (ATM) usability: Transaction times and card loss. *Behaviour & Information Technology, 19(6),* 441–449.

Table 1

All Steps Required to Complete the Frankenstein Task

Step number	Step name	
	Condition 1: Create monsters	Condition 2: Destroy monsters
1	Pick up phone	
2	Get coordinates	
3	***Activate slider***	**Move slider**
4	**Move slider**	***Deactivate slider***
5	Get item at coordinates	
6	***Activate body widget***	**Remove head**
7	Specify body	Classify head
8	Confirm body widget	Confirm head widget
9	***Activate leg widget***	**Remove arms**
10	Specify legs	Classify arms
11	Confirm leg widget	Confirm arm widget
12	***Activate arm widget***	**Remove legs**
13	Specify arms	Classify arms
14	Confirm arm widget	Confirm body widget
15	***Activate head widget***	**Remove body**
16	Specify head	Classify body
17	Confirm head widget	Confirm body widget
18	**Paint**	***Activate wiper widget***
19	Select function	
20	Debug and program	Ok
21	***Deactivate widget***	**Wipe monster brain**
22	**Position laser**	***Activate laser***
23	Operate laser	
24	**Release monster**	***Deactivate***

Note. Target steps are marked in bold, italics denotes device-oriented and underlined denotes task-oriented. Note that steps 3 and 4 are repeated twice, once for each coordinate.

Table 2

Mean Error Rates on Device- and Task-Oriented Steps Across All Participants

Steps	Error Count (Opportunity)	Mean Error Rate (SD), in %
Total	218 (4224)	5.16 (3.57)
Device-Oriented	145 (2112)	6.96 (5.97)
Task-Oriented	73 (2112)	3.55 (3.42)

Note. The second column shows the total number of errors made for each step, and the opportunity for error. The third column shows the average error rate and standard deviation for each step.

Table 3

Mean Step Times on Device- and Task-Oriented Steps Across All Participants

Steps	Mean Step Time (SD), in msec
Total	3976 (3054)
Device-Oriented	4456 (3384)
Task-Oriented	3520 (2625)

Note. Standard deviation is given in brackets.

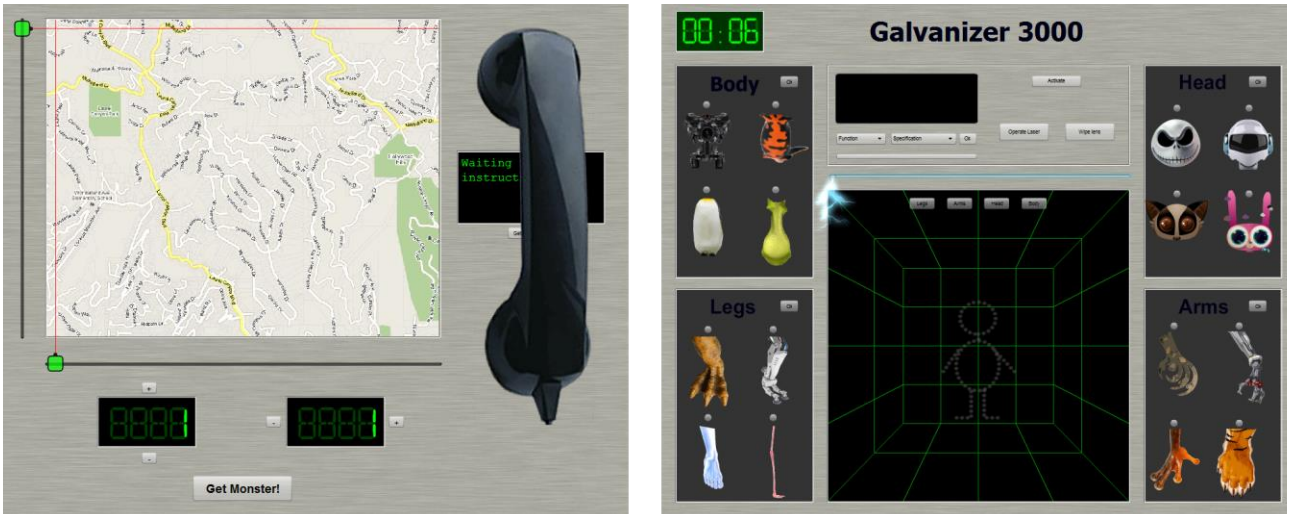

Figure 1. Screenshot of the Frankenstein task. The Locator (left) is shown on a separate screen to the left of the main Frankenstein interface (right). Note that the interface is exactly the same both when participants make monsters and when they destroy them.

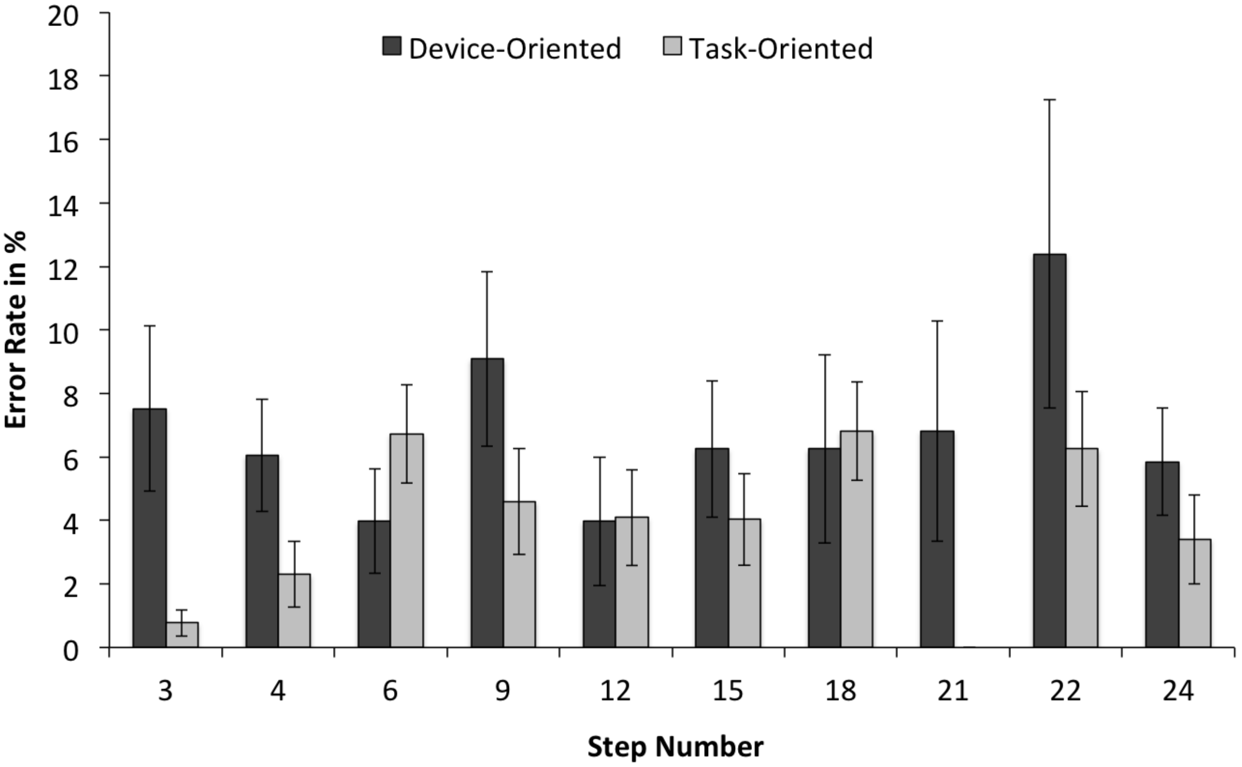

Figure 2. Error rates for each target step. Black bars indicate device-oriented steps, grey ones indicate task-oriented steps. Error bars represent the standard error of the mean.

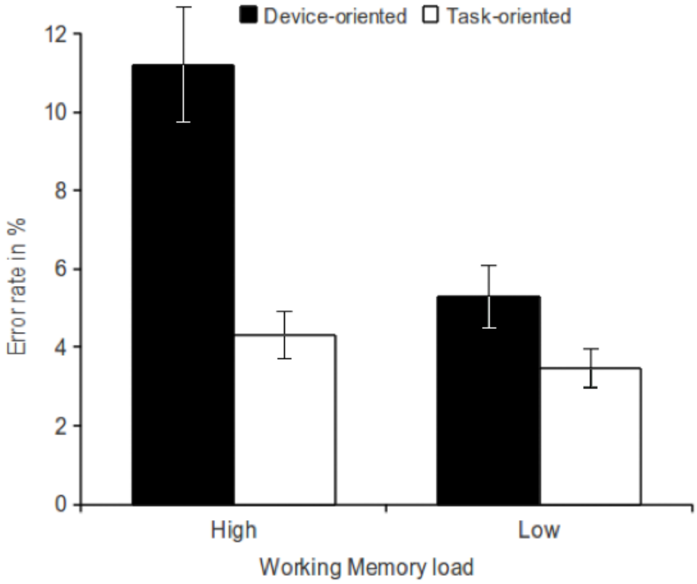

Figure 3. Error rates under high and low load. Black bars indicate device-oriented steps, white bars indicate task-oriented steps. Error bars represent the standard error of the mean.

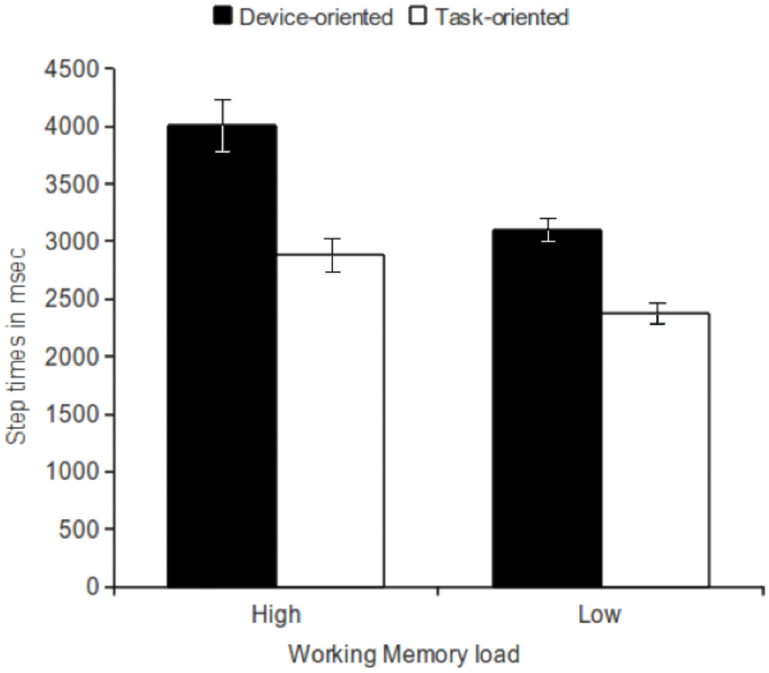

Figure 4. Step times under high and low load. Black bars indicate device-oriented steps, white bars indicate task-oriented steps. Error bars represent the standard error of the mean.

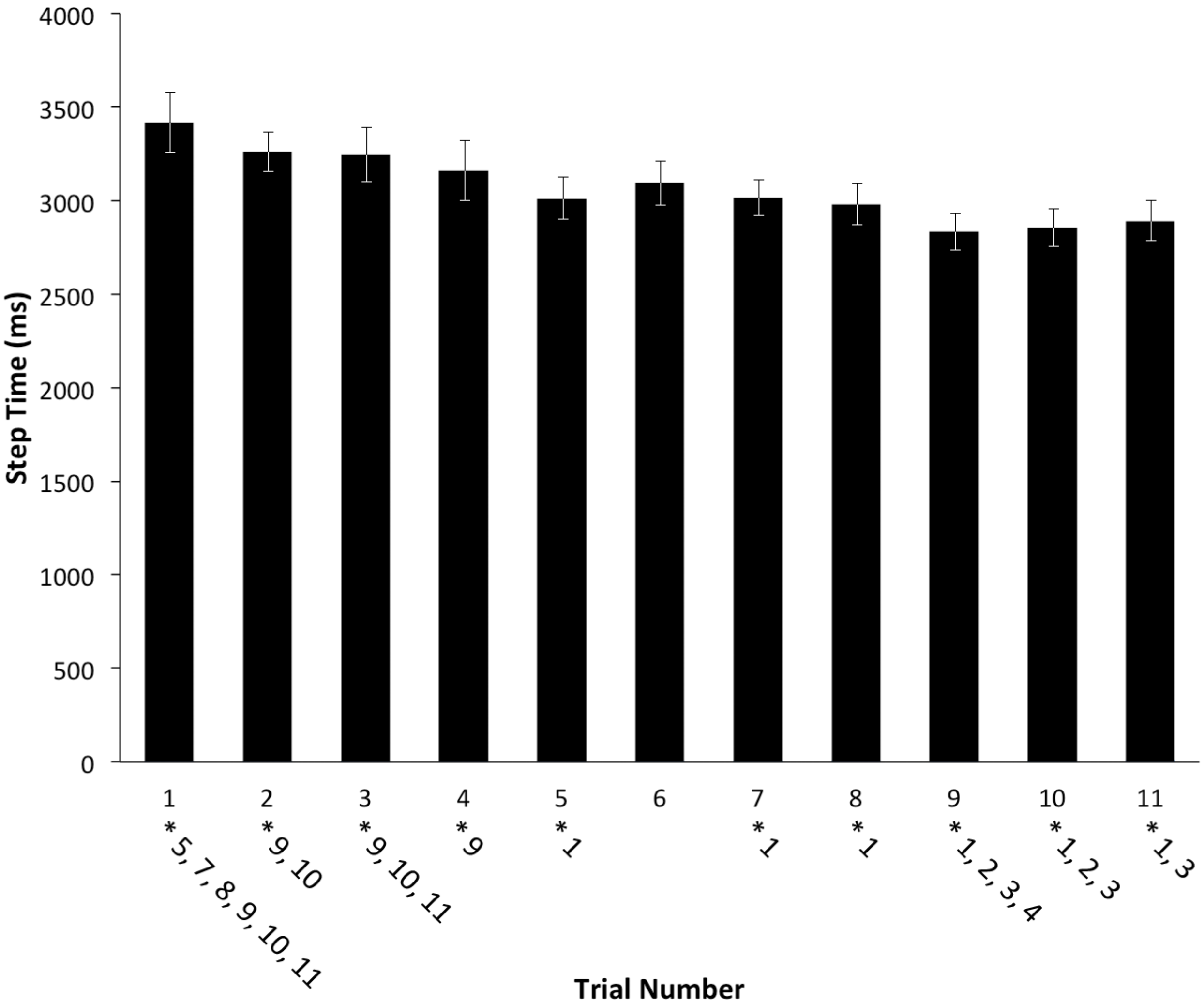

Figure 5. Step times for the different trial numbers. The diagonal numbers on the X axis denote the significant differences (e.g. trial 2 has a significantly longer step time than trials 9 and 10). Error bars represent the standard error of the mean.