Post-completion Errors in Problem Solving

Simon Y. W. Li, Ann Blandford, Paul Cairns & Richard M. Young
{simon.li, a.blandford, p.cairns, r.m.young}@ucl.ac.uk
UCLIC, UCL Interaction Centre
31/32 Alfred Place, London WC1E 7DP UK

Abstract

A post-completion error is a specific form of omission error occurring after some task goal has been accomplished – for example, forgetting to collect your original after photocopying. Using a novel experimental paradigm, this study demonstrates the cognitive robustness of post-completion error: that it occurs not only in routine procedural tasks but also in problem-solving situations. The effect of a simple static visual cue to mitigate the error was also assessed by testing two different interfaces through which users engaged in problem-solving. The findings suggest that even a simple static cue can reduce the error rate, though not eliminate it.

Introduction

In the operation of high-reliability and safety-critical equipment, the consequences of human error can be severe. In the study reported here, we focus on some factors that influence the error rate for a particular class of errors known as post-completion errors. These errors are interesting because they appear to persist over time (and hence cannot be readily attributed to lack of user knowledge). Their persistence suggests that the underlying cognitive causes are directly related to the human cognitive architecture. The work reported here adds to the body of empirical evidence that is needed to construct a validated model to account for occurrences of post-completion error.

Background

There are some empirical studies from cognitive psychology investigating, for example, errors in statistical problem solving (Allwood, 1984), rule-based errors in arithmetic thinking (e.g. Payne & Squibb, 1990; Ben-Zeev, 1995) and slips in speech (Baars, 1992). However, there has been little experimental work investigating human error in the context of Human Computer Interaction (HCI), particularly on capturing error behaviours under controlled laboratory settings. Two of the few examples are the work reported by Byrne and Bovair (1997) and Gray (2000).

One particular type of omission error that is highly relevant to HCI, and has been subject to some early empirical investigations, is the *post-completion error* (Byrne & Bovair, 1997) – sometimes called a 'termination error' (Thimbleby, 1990) or 'omitted secondary sub-goal' (Young, 1994). All these terms refer to broadly the same error behaviour, exemplified by forgetting to collect your change after buying a chocolate bar from a chocolate machine, or to take the original document back after photocopying. In general, a post-completion (PC) error is associated with tasks of a particular structure in which a final 'clean up' step is required *after* completing the main goal. The 'clean up' step is peripheral to the main goal and needs to be carried out before moving on to the next goal; it is usually imposed by the task structure or the artefact in use.

There are only a few studies of PC and similar errors. An early study by Lee (1992) investigated a particular type of action slip known as the unselected window (USW) error, which has very similar characteristics to the PC error. The USW error is best described by the following scenario, which involves using a graphical computer interface: a user has opened several windows at once; s/he is interacting with window A and then needs to switch to window B to extract some data; when this has been done, the user carries on with what s/he was doing with window A but forgets to reselect it; consequently, actions are executed in the wrong window (in this case, window B). Young (1994, p.10) commented that:

> "…this USW scenario is very similar to, or indeed is simply a specific case of, a class of errors known as 'omitted secondary sub-goals', or what Byrne calls 'post-completion errors'… In the present case the primary purpose of the sub-goal, finding the missing information, has been achieved, and the secondary activity of reselecting the original window is forgotten."

Lee (1992) explored the USW error in relation to skill development and the role of visual feedback over a prolonged period. Participants were asked to use a multi-window computer interface to find information in order to buy and sell shares. The effects of static and dynamic visual feedback were investigated. It was found that the number of USW errors did not increase as participants gained more experience with the task (and hence might be expected to become less attentive); neither did the error rate decrease with practice. Dynamic feedback was found to be more effective than static in reducing the error rate; it was suggested that participants became easily habituated to static feedback and its effect then diminished.

Focusing on the role of working memory rather than visual cueing, Byrne and Bovair (1997) carried out a study on PC errors in a controlled laboratory setting; they proposed a working memory capacity account for the error phenomenon. Their experiments involved participants executing various procedures in a computer game in which they had to learn to operate a space ship in order to shoot enemies down. In the high working memory load condition, participants were asked to perform a concurrent secondary task in addition to the primary computer game task. Post-completion errors were found to account for a higher proportion of errors than would be predicted by chance. This is consistent with Lee's finding that the occurrence of the error is persistent even when participants were familiar with the task. In general, Byrne and Bovair found that post-completion errors are rarely made by individuals under low working memory loading; however, the errors become more frequent when low-capacity individual executed a complex task under high working memory load. Although a high working memory demand is one important factor influencing the occurrence of PC errors, others have also been identified. For example, Reason (2002), considering features of the photocopying task, attributes the error to the diversion of attention imposed by a false completion signal (output of the copies) together with the absence of visual cues or other reminders for the PC step.

In looking at ways of combating PC errors, Serig (2001) investigated the effects of organisational interventions such as reprimand, but found no significant reduction in the error rate. This is consistent with earlier findings by Lee (1992): that the error is persistent in nature and that one cannot avoid the error simply by being reminded not to make it again.

In a recent study, Chung and Byrne (2004) examined the effect of visual cues on reducing PC errors. The main PC task used in the experiment was very similar to the procedural task used by Byrne and Bovair (1997). The experiment tested two types of reminders; a mode indicator relying on static contextual information and a dynamic cue occurring just before the PC step. It was found that the visual cue with dynamic movement (flashing arrows in this case) was effective in mitigating against the error, but the static indicator was not; this is consistent with Lee's (1992) findings. Furthermore, it was found that, to be effective, a cue should be meaningful in relation to the PC action, and occur just before the PC step. However, there are examples of situations where the artefact in use might not necessarily be able to track one's state in a task; for example, in the USW scenario mentioned earlier. There is no way the system could tell when one has finished the task with window B. In such situations, the implementation of a just-in-time cue will not be possible and other means of cueing should be adopted.

Aims

The first aim of the current study was to investigate whether or not it was possible to generate the error phenomenon reliably under controlled experimental conditions, using a novel problem-solving paradigm. If this were possible, it would demonstrate that the PC error is a robust cognitive phenomenon that occurs not only in routine procedural tasks but also in problem-solving situations. The second aim of the study was to establish whether the error rate is sensitive to a simple change of interface (from text-based to menu-based) for the problem-solving tasks. A menu-based interface was used to investigate whether the implementation of a static reminder in a simple interface would have any effect in reducing the error; this is motivated by the case when the system is not able to track which step has, or has not, been executed in a task.

Method

The problem setting

Logic problems such as Missionaries and Cannibals (Ernst & Newell, 1969), including some adapted to include a post-completion step, were used in this study. Solving such logic problems should place considerable demands on working memory resources when one is not allowed to write down the intermediate steps of the solution. Since PC errors are prone to high working memory demand (Byrne & Bovair, 1997), solving logic problems "in the head" should be an effective way of producing PC errors without using a concurrent secondary task.

In order to have participants solve the problems "in the head", two interfaces were designed: the Text (Txt) condition and the Pop-up (Pop) condition. With each interface, the user enters each step of the solution into a window, then that step is stored; the user can explicitly look back at previous step(s), but they are not continually displayed.

In the Txt condition, participants had to type their answers into two text boxes and click the `Enter` button (Figure. 1). Object(s) to be moved across the river (e.g. Mrs. Jones and Hen) are typed into the left-hand text box and the sides of the river (e.g. Home or Market) into the text box on the right. Clicking the `Enter` button cleared the text for entry of the next step. In the Pop condition, participants entered their answers into the text boxes by selecting the appropriate item(s) from two pop-up menus (Figure 2). The pop-up menu on the left contains a list of object(s) to be moved across the river, including the transport vessel used in the story (e.g. "seat" in the Dog, Hen & Corn problem, as shown in Figure 3a). The pop-up menu on the right contains the sides of the river (Figure 3b).

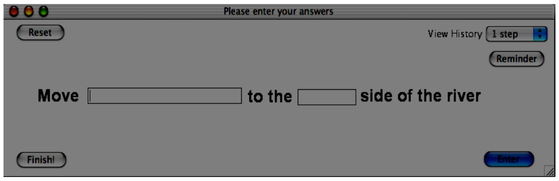

Figure 1: The Txt interface

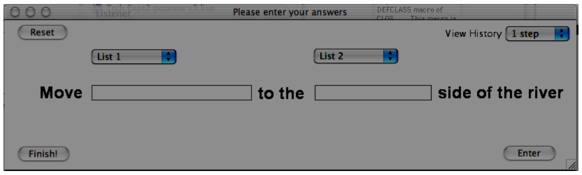

Figure 2: The Pop interface

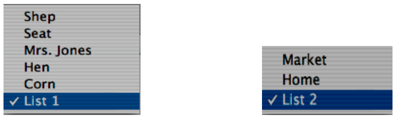

Figure 3: Pop-up menu items (a) left; (b) right (DHC example)

The main difference between the Pop and the Txt interface, in terms of the information available, is the presence of the name of the transport vessel from the problem story. In the Txt interface, clicking on a Reminder button brings up a window containing all the necessary items from the problem story except the transport vessel, whereas in the Pop interface all the items including the transport vessel are present in the pop-up menus.

Tasks

Variants of well-known river-crossing problems (Ernst & Newell, 1969) were used in this experiment. These were called the Father & Son problem (FS), the Three Guests problem (3G), the Lion, Monkey & Banana problem (LMB), the Dog, Hen & Corn problem (DHC), the Torch problem (Trch) and the Itchy & Scratchy problem (IS). The latter three problems contained PC steps which required the participant to send the transport vessel back to the other side of the river as the very last step of the solution. Each problem was adapted into the form of a story so that the post-completion step was an integral part of the story.

The instructions for three problems that included a PC step all contained a sentence indicating the presence of a PC step, as follows:

DHC □ *Whoever finishes using the seat always returns it to the market side, where Mr. Edison lives, so that it is easy for him to take it in each evening.*

Trch □ *When they reach the other side of the river, they have to send the boat back to the forest so that no one will notice it has been used.*

IS □ *They realise that they have to send the crate back to the initial side when they reach the other side so no one would suspect that it had been used.*

Participants

41 undergraduate and postgraduate students from UCL participated in the study. There were 19 males and 22 females, with ages ranging from 21 to 50.

Materials

The two interfaces used in the Txt and Pop conditions were implemented in MCL 5.0 on a Macintosh iMac. All problem stories were printed on paper.

Design

The independent variable was the interface condition (Txt or Pop) which was manipulated between subjects. The dependent variable was the number of PC errors made in the PC tasks. All participants did all six problems using one of the interfaces. The post-completion problems were placed in positions 2, 4 and 6 in the sequence of problem presentation and the sequence was counterbalanced using a 3□3 Latin Square. Positions 1, 3 and 5 in the problem presentation sequence were filled with non-post-completion problems, and were again counterbalanced using a 3□3 Latin Square.

Procedure

Participants were randomly assigned to one of the two conditions and went through the following procedures:

Instruction phase First, each participant was presented with a set of instructions, in PowerPoint animation, about solving a simple river-crossing problem via the specially designed interface. The instructions were identical in the Txt and the Pop conditions except those relating to the computer interface for entering answers. The solution of the example problem was walked through during the instruction presentation; the first step of the solution was an explicit action of moving the transport vessel to one side of the river. The purpose of this was to ensure the participant knew s/he could enter the transport vehicle alone. The instructions also stated that the transport vessel only needed to be specified when it was moved on its own with no passenger(s). This was to eliminate the need to enter the transport vehicle into every step of a solution. For example, if one wants to move “Mrs. Jones”, “Hen” and “seat” across the river, only “Mrs. Jones” and “Hen” need to be entered into the long text box; however, when the “seat” needs to be sent back on its own after everything crosses the river, then “seat” needs to be entered into the text box.

The participants were also told that for every problem solved successfully they would receive a payment of 50p. This was intended to increase participants’ motivation to solve the problems.

Training trial After the instruction phase, participants were asked to perform a training trial. They had to read a simple river-crossing problem, presented on paper, then the paper was taken away so that the participant had to memorise the rules, names of the object(s) to be moved and the sides of the river in the problem. To ensure that participants were aware of the post-completion step in the problem, a reminder “*Please also send the boat back to the coffee shop when finish taking everything across*” was included at the end of the story in addition to the post-completion sentence “*the boat is always sent back to the coffee shop side of the river, as a convenience to the staff for the next order*”.

Before participants started entering their answers into the interface, they were asked to complete a checklist, consisting of 14 statements (6 true, 8 false) which they had to identify as being true or false. They were also asked to rate how important the true statements were (Not at all, Moderate or Very) for solving the problem. The statement “The boat is always sent back to the coffee shop side of the river after use" was used to check their understanding of the post-completion step.

Feedback on any errors made (including PC errors) was given to the participants at the end of the training trial so that they knew whether they had completed the trial successfully. The solution to the training trial problem involved explicitly sending the transport vessel to the other side of the river on alternate steps to reinforce to participants that this was a permissible action.

Test trials Following the training trial, participants were asked to complete a series of six logic problems. As in the training trial, participants were asked to read a story and enter their answers via the same computer interface. A checklist was not used in the test trials phase.

Participants were told that there was no time pressure in solving the problems. However, they were told that the session would last for about an hour, and that the experimenter would invite them to move on to the next problem if they appeared to be unable to solve any particular problem. In this case, the experimenter would ask the participant to give a second best answer. The rationale was to have participants finishing each problem regardless of the correctness of the solution, to see whether they would still commit post-completion errors in the three problems.

No feedback to the solutions was given during the test trials, to minimise the effect of explicitly reminding participants about the PC steps in the three problems. Feedback were given at the end of the experiment. During each experimental session, the experimenter was present in the room with the participant. The experimenter answered clarification questions about the rules of the problems, but not questions regarding the solutions to the problems. At the end of the experiment, participants were debriefed about the objective of the experiment. Each session lasted approximately an hour.

Results

Data from 36 participants, 18 in each condition, were used for the analysis because 5 participants terminated the experimental session early.

Since each participant had three possible opportunities to make at least one PCE, the whole data set should have had 54 data points in each condition. However, 12 data points were excluded from the analysis because 5 of them (3 in the Txt condition and 2 in the Pop condition) were problems that were not completed, and 7 of them (2 in the Txt condition and 5 in the Pop condition) contained some ambiguity about the PCE[1] or had solutions that were unclear in relation to the problem space. Therefore, analysis was based on 49 data points from the Txt condition and 47 from the Pop condition.

Checklist response

All 36 participants, in both conditions, correctly acknowledged the PC step as part of the solution in the training trial. In the Txt condition, 16/18 participants indicated the importance of the PC step as "very" and only 2/18 participants indicated it as "moderate". In the Pop condition, 17/18 participants responded the PC step as "very" and only 1/18 responses was "moderate".

Overall PC error rate

Collapsing across Txt and Pop conditions, the data shows that 69% of participants (25/36) made at least one PC error. Excluding missing data, 75% of participants (18/24) made at least one PC error. A total of 44 PC errors occurred between the two conditions giving an overall error rate of 45% (out of 96 opportunities)

PC error rate in Txt and Pop condition

Across the three post-completion tasks, a total of 29 PC errors occurred in the Txt condition, which yielded an error rate of 59% (29 out of 49 opportunities). In the Pop condition, a total of 15 PC errors occurred, giving an error rate of 32% (15 out of 47 opportunities).

A Mann-Whitney test was applied to assess whether there was a significant difference in PC error rate between the two conditions. A score was calculated for each participant based on the number of PC errors committed, with one error counting for a score of 1 (so each participant could have a maximum score of 3 and a minimum of 0). Missing data were handled by list-wise exclusion. The effect between the two conditions was significant, U=37, p=.037. Thus, overall, there was a significant difference between the two conditions in terms of error rate.

$\square^2$ comparisons showed a significant difference between the two conditions in the IS task ($\square^2$=6.00, df=1, p=.014), the differences were not significant in the DHC task ($\square^2$=2.44, df=1, p>.10) or the Trch task ($\square^2$=.72, df=1, p>.35). However, in all three individual tasks, there were consistently a higher proportion of participants committing PC errors in the Txt condition than in the Pop condition (Figure 4).

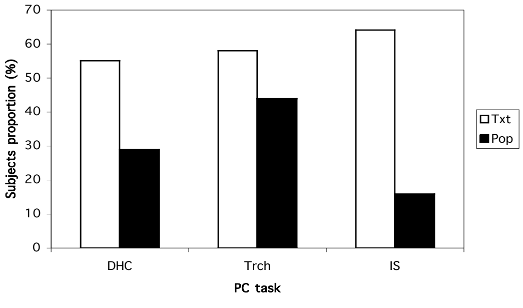

Figure 4: PC error rate in all three tasks

[1] A PC error was classified as ambiguous if participant reported that s/he was unclear about the PC step before finishing the trial.

Discussion

The study achieved its first aim: of demonstrating that PC errors can be generated reliably under laboratory conditions using a problem-solving task. Responses from the checklist suggest that participants had the correct knowledge about the PC step in a problem. More than half of the participants made at least one PC error and this suggests the observed error rate was not due to a few particularly error-prone participants. Furthermore, Byrne & Bovair (1997) obtained an error rate of about 50% in their high working memory load condition; compared to the 45% obtained in this study. This confirms the robustness of the error phenomenon: that it occurs not only in routine procedural tasks but also in problem-solving situations.

The problem-solving tasks used in this study had a couple of characteristics that made them particularly effective in provoking PC errors that are worth mentioning. First of all, participants were required to solve the problems "in the head", without the use of any form of external representation, which placed a high demand on working memory (for keeping track of the problem state). The use of the problem-solving tasks in generating the error is functionally similar to the use of a concurrent secondary task to tax working memory (Byrne & Bovair, 1997), except that the working memory demand is an integral part of the primary problem-solving task, removing the need for an artificial secondary task. Secondly, Altmann and Trafton (2002) suggest that the relatively low PC error rate in routine procedural tasks (e.g. using an ATM) is due to rote associative learning □ in which the temporal order of the steps in a procedural task helps to cue the next step in the sequence. The relatively low occurrence of PC errors is explained in terms of the successful cueing from the preceding step to the PC step. The problem-solving tasks adopted in the current study are not routine procedural tasks, so the extent of procedural cueing from one step to the next is minimal.

The second aim of this study was to investigate whether the error can be mitigated through a simple change of interface (from text-based to menu-based) for the problem-solving tasks. The PC error rate was found to be significantly lower in the Pop condition than in the Txt condition. The result suggests that the inclusion of a visual cue in a menu-driven interaction can significantly reduce the occurrence of PC errors when using the interface to solve the problem tasks. One interpretation of this result is that the presence of the name of a transport vessel (e.g. crate) in the pop-up menu acts as a reminder for the PC step of sending the vessel back to the other side of the river. According to memory theory, with the reminder which associatively links to the PC step, the suspended goal of carrying out the PC step may regain its activation, compensating for the loss in activation in the decay process which started at the point of goal suspension (Altmann & Trafton, 2002), so that it gets retrieved after the step preceding the PC step.

While the overall PC error rate was lower in the Pop condition, the results from individual tasks (Figure 4) are rather puzzling: the reminder effect in the pop-up menus was only found to be significant in reducing the error in the IS task and not the DHC or the Trch task. One possible explanation comes from looking at the menu lists of the three tasks (Figures 3a and 5). One feature of the DHC and Trch menus is that there were five items in each of the menu lists, whereas there were only three items in the IS menu list. In addition, the vessel is embedded within the list in the DHC and Trch cases, but appears at the end (right next to the initially highlighted item) in the IS case. When using these lists, participants' visual attention on the vessel item (i.e. Seat or Boat) might have been distracted by other items in the list for DHC and Trch, making the vessel less conspicuous and, consequently, less effective as a reminder to the PC step when compared to the IS list.

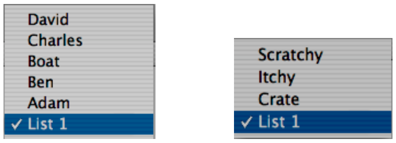

Figure 5: Menu items in (a) Trch (b) IS problems

In contrast to the study by Chung & Byrne (2004), in which PC errors were completely eliminated by their visual cue design, it is worth asking why PC errors still persist in the Pop condition in our study. Although having a static cue in the pop-up menus seems to have had a reminder effect in carrying out the PC step (at least in the IS task), a couple of characteristics of the cue itself might suggest why it did not manage to reduce the error rate substantially. In contrast to the visual cue used in Chung & Byrne's study, the cue adopted in the current study did not have dynamic-movement and just-in-time properties. The persistent error rate in the Pop condition could be explained by the current static cue not being enough to capture participants' attention; in addition, the fact that the cue did not occur *just before* the PC step reduced its role as a direct reminder about the execution of the PC step. However, this explanation does not account for the fact that the current static cue still had any effect in reducing the error; this is counter to one of the findings in Chung & Byrne's study – that their static cue did not have any effect on reducing the error. An alternative interpretation of the current finding is that the tasks used in Chung & Byrne and the current study differ in many ways, suggesting that the context of the visual cue also plays an important role in determining the cue's effectiveness in reducing the error. The task in Chung & Byrne's study involves an interface which is visually more dynamic and complex than the one used in this study; furthermore, their interface involves more interactions than simply typing or selecting menus, as in the current study. The static information and simple interaction imposed by the current task environment might help explain why a simple static cue had an effect in partially reducing the error rate in the Pop condition.

Conclusion

Findings from the current study have several implications in relation to the study of human error and interface design. This study contributes to the small body of experimental studies of human error in the context of HCI, and of PC errors in particular, by successfully provoking the occurrence of errors under controlled laboratory conditions. Previous studies have investigated this error phenomenon within the boundary of highly familiar or proceduralised tasks; the current study extends that boundary to problem-solving tasks which possess the PC characteristic but are not necessarily familiar to participants. The occurrence of PC errors in problem solving demonstrates the pervasiveness and cognitive robustness of the phenomenon.

This study also has some methodological implications with regard to provoking PC error. The problem-solving tasks adopted in the experiment have several characteristics that contrast with the routine proceduralised tasks used by previous studies. Firstly, the problem-solving tasks do not have a fixed repetitive pre-completion task structure. In Byrne & Bovair's task, a certain number of preceding steps to the PC step were carried out repeatedly until the main goal was achieved, then the PC step could be executed. Secondly, the problem-solving paradigm did not require extensive training on the participants' part. Although the participants had training to ensure they had knowledge of the PC step and action, they were not trained in doing the problems. Thirdly, the false completion signal was internally driven (when the participants thought they had solved a problem) rather than being located externally in the environment as with a photocopier (Reason, 2002) or having a completion signal on the computer screen. The successful generation of the error using a sufficiently different task paradigm further highlights the robust cognitive nature of the error phenomenon.

Post hoc explanations have been proposed to account for the partial reduction of errors by the static cue in the current task interface, and further experimentation needs to be carried out to clarify the effect of the number and ordering of menu items in the menu lists or the visual complexity of the task environment. Whatever the ultimate explanation, the current findings show that a simple static visual cue could reduce PC errors, and the context of the task environment could also be an important determinant of the effectiveness of the cue. In practice, providing a just-in-time cue might not be possible if a system cannot tell where the PC step lies in a particular task.

Further research could be directed to investigate cue characteristics that best mitigate PC errors when the artefact in use does not detect whether one has fulfilled one's main goal. In the current study, completion of the main goal could not be detected by the system, whereas Chung & Byrne's interface could use a just-in-time cue because the system could detect when the main goal had been achieved. Other factors that might influence post-completion rates, such as motivation, attention, interruptions, etc. remain to be studied before it will be possible to develop a comprehensive model that accounts for such errors.

Acknowledgments

We are grateful to all participants who took part in this study. The research is funded by a studentship from UCL.

References

Allwood, C. M. (1984). Error detection processes in statistical problem solving. *Cognitive Science*, *8*, 413-437.

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

Baars, B. J. (1992). *Experimental slips and human error: Exploring the architecture of volition*. New York: Plenum.

Ben-Zeev, T. (1995). The nature and origin of rational errors in arithmetic thinking: Induction from examples and prior knowledge. *Cognitive Science*, *19*, 341-376.

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

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

Ernst, G. W., & Newell, A. (1969). *GPS: A Case Study in Generality and Problem Solving. ACM Monograph Series.* New York: Academic Press

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

Lee, W. O. (1992). The Effects of Skill Development and Feedback on Action Slips. In A. Monk, D. Diaper and M. Harrison (Eds.)*, People and Computers VII* (pp. 73-86)*. Cambridge: Cambridge University Press.*

Payne, S. J. & Squibb, H. R. (1990). Algebra mal-rules and cognitive accounts of errors. *Cognitive Science*, *14*, 445-481.

Reason, J. (2002). Combating omission errors through task analysis and good reminders. *Quality of Safe and Health Care*, *11*, 40-44

Serig, E. M. (2001). Evaluating Organizational Response to a Cognitive Problem: A Human Factors Approach. Doctoral dissertation, Rice University, Houston, TX.

Thimbleby, H. (1990). *User Interface Design*. ACM Press.

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