

The Effect of Interruptions on Postcompletion and Other Procedural Errors: An Account Based on the Activation-Based Goal Memory Model

Simon Y. W. Li, Ann Blandford, Paul Cairns, and Richard M. Young
University College London

A postcompletion error (PCE) is a specific kind of cognitive slip that involves omitting a final task step after the main goal of the task is accomplished. It is notoriously difficult to provoke (and hence study) slips under experimental conditions. In this paper, the authors present an experimental task paradigm that has been shown to be effective for studying PCEs in routine procedural tasks. Two studies were carried out to examine the effect of interruption position and task structure on the prevalence of PCEs. It was found that significantly more PCEs were obtained when an interruption occurred just before the PC step than when an interruption occurred at any other position in the task. The authors account for this effect in terms of Altmann and Trafton's activation-based goal memory model. The same interruption effect was obtained for some, but not all, other procedural errors; the authors discuss the nature of these errors and likely explanations for the differences.

Keywords: human error, postcompletion error, interruption, memory, HCI

A postcompletion error (PCE) is a specific kind of omission error, which occurs after the completion of the main goal of a task. Examples include leaving your change behind after purchasing from a vending machine and forgetting to retrieve the original from a photocopier. PCEs are interesting because they are not the result of incomplete or incorrect knowledge but of memory lapses in executing the task procedure. In relation to Norman's (1981) slip/mistake distinction of human error, Byrne and Bovair (1997) describe PCEs as a kind of slip. Furthermore, PCEs are infrequent but persistent: People do not always make the error but they make it from time to time.

Experimental studies of human error have investigated, for example, errors in statistical problem solving (Allwood, 1984), rule-based errors in arithmetic thinking (e.g., Ben-Zeev, 1995; Payne & Squibb, 1990) and slips in speech (Baars, 1992). However, little experimental work has investigated human error in the context of HCI, particularly under controlled laboratory conditions. Some researchers pointed out that this is because there is no research tradition in studying human error experimentally (Gray,

2004) and the inherently low frequency of error data in general (Sellen & Norman, 1992).

This paper has several objectives. The first objective is a substantive one: to investigate the effect of interruption position on PCEs; this draws on the activation-based goal memory model of Altmann and Trafton (2002) as described below. The second objective is a methodological one: to design a new task environment for studying PCEs in a laboratory setting; this has been informed by the work of Byrne and Bovair (1997), whose "Phaser task" has been the main one used in studying PCEs to date. The third objective is an analytical one: to investigate the effect of interruptions on other errors as well as PCEs. Previous studies on PCEs have not analyzed the effect of their manipulations on other errors in the same task environment.

Laboratory Studies of PCEs

The most extensive systematic studies of PCEs to date have been conducted by Byrne and colleagues (Byrne & Bovair, 1997; Byrne & Davis, 2006; Chung & Byrne, 2008). These studies have considered different factors that might influence the prevalence of PCEs: working memory load, just-in-time cueing, and various motivational factors. All these studies have used experimental materials based on the same basic task and interface setup, referred to as the Phaser task.

The Phaser task is a complex computer-based game. The objective of the game is to destroy enemy ships, and operation of the Phaser involves following a predefined procedure with various subgoals. To fire the Phaser, a component called "Tracker" needs to be switched on, and when the enemy ship is destroyed the "Tracker" needs to be switched off. Forgetting to switch off the "Tracker" is classified as a PCE.

Using the PCE-inducing Phaser task, Byrne and Bovair (1997) investigated the effect of working memory on PCEs. In a condition where working memory was taxed, participants were required to

Simon Y. W. Li, Ann Blandford, Paul Cairns, and Richard M. Young,
UCL Interaction Centre, University College London.

Simon Y. W. Li is now at the Centre for Health Informatics, University of New South Wales. Paul Cairns is now at the Department of Computer Science, University of York.

We thank Frank Durso, Erik Altmann, and three anonymous reviewers for their constructive feedback to improve the manuscript. We would like to acknowledge the support of UCL Psychology Department. The first author would also like to thank Magda Osman and Anna Cox for constant support and encouragement.

The series of experiments described in this paper was supported by a studentship from UCL Psychology Department.

Correspondence concerning this article should be addressed to Simon Y. W. Li, Centre for Health Informatics, Coogee Campus, University of New South Wales, Sydney NSW 2052. E-mail: s.li@unsw.edu.au

carry out a concurrent letter recall task in addition to the Phaser task. In the control condition, participants only had to perform the primary task. The working memory capacity of each participant was also measured, and participants were divided into two groups: low and high capacity. It was found that PCEs were rarely made by high-capacity participants with low working memory load (about 15% of the time), but low-capacity participants with high working memory load made reliably more PCEs (about 75% of the time). PCE rate did not differ significantly between high-capacity participants with high load and low-capacity participants with low load. Byrne and Bovair proposed a working memory capacity account to explain the effect, suggesting that under high working memory load the PC subgoal is more likely to be displaced from working memory when the main task goal is accomplished.

Using the same experimental paradigm, Chung and Byrne (2008) examined the effect of visual cues on PCEs. Specifically, performance was compared when participants were given a dynamic just-in-time cue (occurring just before the PC step of switching off the "Tracker") and a static asynchronous cue (occurring earlier in the task). It was found that the dynamic just-in-time cue was effective in eliminating PCEs; however, the static asynchronous cue had no reliable effect on PCEs when compared to the control condition.

Similarly, Byrne and Davis (2006) adopted the Phaser task paradigm to examine the effect of various motivational strategies, such as praising good performance, reporting poor performance, retraining and practice, on the occurrence of PCEs. A task redesign condition was also tested in which the Phaser task's structure was changed so that the PC step had to be carried out *before* task completion; in other words, the task redesign eliminated the PC structure. It was found that occurrences of PCEs decreased in all conditions and across testing sessions, and Byrne and Davis concluded that the motivational factors that were tested did not have any added effect over practice alone in combating PCEs. The task redesign condition obtained a complete elimination of PCEs but none of the other interventions differentiated statistically in error reduction. Results from this study suggest that PCEs are persistent and the complete elimination of the error might be best achieved by avoiding a PC task structure, where possible.

Although PCEs have received more attention recently, there are still many open questions about what other factors might provoke or mitigate the error. One such question this paper is going to address is whether or not the occurrence of PCEs is affected by interruptions. A further point to mention about the studies of PCEs to date (e.g., Byrne & Bovair, 1997; Byrne & Davis, 2006; Chung & Byrne, 2008) is that they examined PCEs in isolation without any reference to other errors that might have been affected by the same experimental manipulations. As a result, it is difficult for readers to judge whether PCEs are different from other errors or not. Therefore, to establish the effect of interruption on PCEs, it is necessary to study the effect in relation to other errors in the same task environment.

Research on interruptions have suggested several factors that affect primary task performance, for example, the complexity of the interruption and its similarity to the primary task (Edwards & Gronlund, 1998; Gillie & Broadbent, 1989), the role of retrieval cues after an interruption (e.g., Cutrell, Czerwinski, & Horvitz, 2000), control over the interruption (McFarlane & Latorella, 2002), and preparation before engaging with the interruption

(Trafton, Altmann, Brock, & Mintz, 2003). Although most studies have shown that interruptions disrupt primary task performance, for example, increasing levels of error (Latorella, 1999), there are some suggesting that interruptions can facilitate performance on simple tasks (e.g., Ratwani, Trafton, & Myers, 2006; Speier, Vessey, & Valacich, 2003). Despite the general disruptive effect of interruptions on primary task performance, most studies have not specified what *kinds* of errors are more prone to the disruptive effect of interruptions, or whether the kinds of interruptions affect error rates.

The AGM Model

A particularly useful theoretical framework for the current study is Altmann and Trafton's (2002) activation-based goal memory (AGM) model. The AGM model has its origin in explaining goal suspensions and resumptions in problem-solving and has been applied to investigate the disruptiveness of interruption on primary task performance in a range of tasks, for example, tactical decision-making (Trafton et al., 2003), problem-solving (Hodgetts & Jones, 2006a; 2006b), and VCR programming (Monk, Boehm-Davis, & Trafton, 2002).

Using the construct of activation, the AGM model suggests that, just like other memory elements in the cognitive system, goals have associated activation levels and cognition is directed by the most active goal retrieved at any given time. The amount of activation associated with a memory item is subject to decay, and this decay process is time-based and gradual.

If the cognitive system needs to refocus attention to (or resume) an old goal then this old goal needs to undergo a priming process to become active again. The priming process is possible through associative links between retrieval cues and the to-be-resumed goal. A retrieval cue can be internal, residing in the cognitive system; a procedural task step can act as a cue for the subsequent task step. On the other hand, a retrieval cue can also be external, residing in the environment; for example, a loud beeping signal in a ticket machine when it returns the change can prime the action of collecting the change provided the relationship between the cue (the beep) and the action (collection of change) is learned.

The AGM model suggests that execution of task steps in procedural task performance can be viewed as a sequence of associative links, each action step acting as a retrieval cue for the next. This associative cueing mechanism explains how PCEs are usually avoided.

Consideration of the AGM model enables us to predict that different interruption positions occurring during procedural tasks will result in different effects on the rate of PCEs. An interruption occurring just before the PC step is more likely to lead to a PCE than an interruption occurring at any other point within a task structure. An interruption just before the PC step will have disrupted the associative priming from the preceding step, whereas other interruption points will still allow the pre-PC step to be carried out (after the correct goal is resumed successfully), and once it is carried out, it cues the PC step. It is the prediction that different interruption positions would have differential effects on the rate of PCEs that is tested in the following studies.

Different Kinds of Procedural Errors

One of the objectives of the current study is to examine PCEs in relation to other errors; therefore, it is worth a preliminary description of two main kinds of procedural errors obtained in the current study.

To describe the two kinds of procedural errors, it is necessary to describe briefly the experimental task (details will be presented later). The current experimental task is a computer-based game simulating a procedural task of producing doughnuts. This doughnut-making task consists of a number of sub-tasks, such as specifying the quantity of dough, shapes of doughnuts, sprinkles on doughnuts and so forth, which are carried out in a predefined order. To carry out a sub-task, the sub-task has to be initialized by using a "Selector" to activate it.

Two different kinds of errors could happen at the beginning of each sub-task: a *sub-task sequence error* and a *sub-task initialization error*. A sub-task sequence error is an error in which a sub-task is activated by an action on the "Selector" but the initialized sub-task is an incorrect one in the procedure. A sub-task initialization error is one in which the correct sub-task is performed but not initialized by an action on the "Selector."

In terms of generality of the two kinds of errors, the sub-task sequence error is caused by "misordering the components of an action sequence" (Norman, 1981, p. 6) or is referred to as a program counter failure (Reason & Mycielska, 1982). The sub-task initialization error is very similar to the unselected window error (Lee, 1992), in which during operations in window-based computer applications, a desired window has not been selected and input has gone into an "incorrect" window. The sub-task initialization error also resembles a mode error, for example, forgetting to activate a required mode in programming a video recorder (Gray, 2000).

Programming a VCR to record a desired show provides an example of a procedural task that could give rise to sub-task sequence and sub-task initialization errors. The task of programming a VCR usually consists of the following sub-tasks: (a) Set Programme Channel, (b) Set Record Date, (c) Set Start Record Time, and (d) Set End Record Time. A sub-task sequence error could occur if a sub-task was not executed in the specified sequence. In some VCR models, setting the time to start and end a recording requires the user to specify the mode the machine should be in (Gray, 2000). In this example, before programming Steps c and d one needs to switch the VCR to a "Start" mode and an "End" mode correspondingly. The mode-switching steps can be viewed as initialization steps and omissions of these steps results in sub-task initialization errors.

In relation to the AGM model, we postulate that the procedural representations of sub-task sequence steps and sub-task initialization steps have different degrees of dependency on the associative cueing mechanism. This is because the sub-task sequence steps are integral to a predefined task sequence, whereas the sub-task initialization steps are "extra" steps imposed by the device. The correct execution of sub-task sequence steps and sub-task initialization steps rely on the correct knowledge of the task and the device respectively (Cox & Young, 2000). The associative cueing mechanism might not apply so well to sub-task initialization steps, because they are device specific. The execution of sub-task ini-

tialization steps might depend less on associative cueing and more on a deliberate and less automatic mechanism.

It should be pointed out that the prediction of the effect of interruption position made by the AGM model is not restricted to PCEs. In other words, any task step in a procedural task that is preceded immediately by an interruption is likely to result in an error upon resumption. However, this prediction only applies to task steps that are dependent on the associative cueing mechanism, that is, sub-task sequence errors. Therefore, it is also predicted that executions of task steps that are not dependent on associative cueing, that is, sub-task initialization errors, will not be affected by the disruptive effect of an immediately preceding interruption.

Overview of the Current Experiments

Two experiments were carried out to examine the effect of interruption position on PCEs. For the purpose of these studies, an interruption is taken to mean the abrupt onset of a different task activity during the execution of a primary task. The experimental methodology adopts a procedural task paradigm in which the primary task involves participants executing a set of predefined instructions in a task environment.

The first experiment, as described below, required participants to follow a specified procedure for making doughnuts using a simulated doughnut-making machine. Each trial was delineated by the participant having to turn to a second computer to complete a short interleaving task; an additional interrupting task was introduced at selected points within the main task to study the effects of interruptions. The postcompletion step, which was to clean the machine, had to be performed before turning to the second computer.

The second experiment was similar, but rather than performing the "clean" step at the end of every trial, it had to be completed on one third of the trials; this change was introduced to investigate the effect of making the PC step less practiced.

Experiment 1: The Effect of Interruption Position

Method

Tasks

The primary task: Doughnut task and Call Centre. The Doughnut task is a procedural task in which participants are required to carry out a set of predefined procedures to produce a specified number of doughnuts by operating a doughnut machine. Figure 1 is a screen shot of the Doughnut task.

The Order Sheet in the center contains information about how many and what kinds of doughnuts are to be made in each order. Two sets of doughnuts are to be made every time in one order. To produce the doughnuts as specified on the Order Sheet participants have to operate the five different compartments in the machine in a specific sequence: (a) Dough Port, (b) Puncher, (c) Froster, (d) Sprinkler, and (e) Fryer. Each compartment needs to be activated, using the "Selector" on the right, before any parameters can be entered. Table 1 summarizes all the steps required to operate the machine in a trial, including the possible interruption positions.

With each of the compartments, participants need to click OK when they finish entering in various parameters. Once the OK button is clicked, information just entered in that compartment is

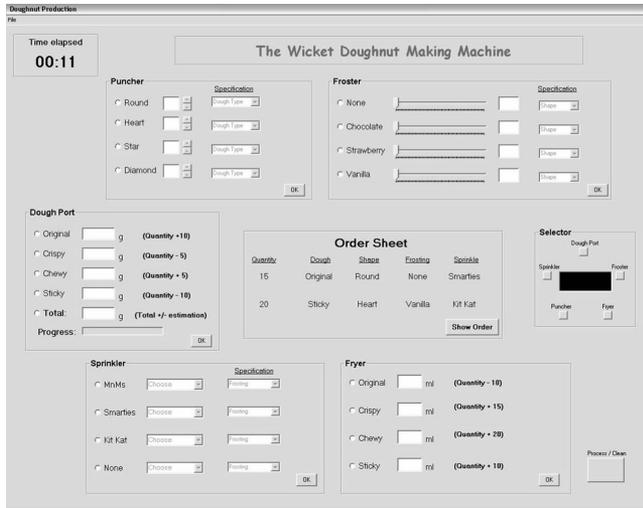


Figure 1. The Doughnut task.

erased, returning its appearance to its initial state. This ensures that there are no cues in the task environment to distinguish where one is in the doughnut-making process. After entering parameters into the five compartments, participants need to click “Process/Clean”

(bottom right) to obtain a report indicating how many doughnuts are made. When they have finished reading the completion report, participants have to dismiss it by clicking OK, and the screen returns to its initial state (see Figure 1). The final PC step is then to clean the machine by clicking the “Process/Clean” button again.

The Call Centre serves as the follow-on task. It is a simple search task where participants have to find a specified location from the London Underground Map to get an order to make doughnuts. Figure 2 depicts the transition between the Doughnut task and the Call Centre (arrows 1 and 2). The Doughnut task and the Call Centre were implemented on separate computer terminals, placed at 90 degrees, to simulate an environment where one has to move away physically from the main Doughnut task when it is completed.

At the beginning of a trial, the center of the Doughnut task indicates a location to collect an order (St. Paul’s in this case) and the participant is required to turn to the Call Centre computer terminal to find the location, then drag and drop the location into a text box and press a button to get the order. The participant then returns to the Doughnut task terminal and begins the doughnut-making process. The PC step (clicking “Process/Clean” at the bottom right corner; see Figure 1) in the Doughnut task needs to be executed before responding to the next specified location in the Call Centre, otherwise it is classified as a PCE.

Interrupting Task: The Packaging Task. The interruption is a mental arithmetic task called the Packaging task. The task is to

Table 1
Descriptions of the Steps in the Doughnut Task and the Interruption Positions

Step number	Step name	Actions involved
1	Next Order	Click on the Next Order button to show order details.
2	Select_DP opt_DP(x)	Click on the Dough Port Selector button.
3	opt_DP(y) opt_DP (Total)	Enter parameters in the Dough Port compartment.
4	DP_OK	Click on the OK button in the Dough Port compartment.
5	DP_ProgBar_Filling	Wait for the progress bar in the Dough Port compartment to fill up.
Interruption P		
6	Select_Puncher	Click on the Puncher Selector button.
7	opt_Puncher(x) opt_Puncher(y)	Enter parameter in the Puncher compartment.
8	Puncher_OK	Click on the OK button in the Puncher compartment.
Interruption Q		
9	Select_Froster	Click on the Froster Selector button.
10	opt_Froster(x) opt_Froster(y)	Enter parameter in the Froster compartment.
11	Froster_OK	Click on the OK button in the Froster compartment.
Interruption R		
12	Select_Sprinkler	Click on the Sprinkler Selector button.
13	Opt_Sprinkler(x) Opt_Sprinkler(y)	Enter parameter in the Sprinkler compartment.
14	Sprinkler_OK	Click on the OK button in the Sprinkler compartment.
Interruption S		
15	Select_Fryer	Click on the Fryer Selector button.
16	opt_Fryer(x) opt_Fryer(y)	Enter parameter in the Fryer compartment.
17	Fryer_OK	Click on the OK button in the Fryer compartment.
Interruption T		
18	Process	Click on the Process button.
19	Report_OK	Wait for the doughnut report to appear, read it and click on the OK button to dismiss it.
Interruption Z		
20	Clean	Click on the Clean button (this is the PC step).

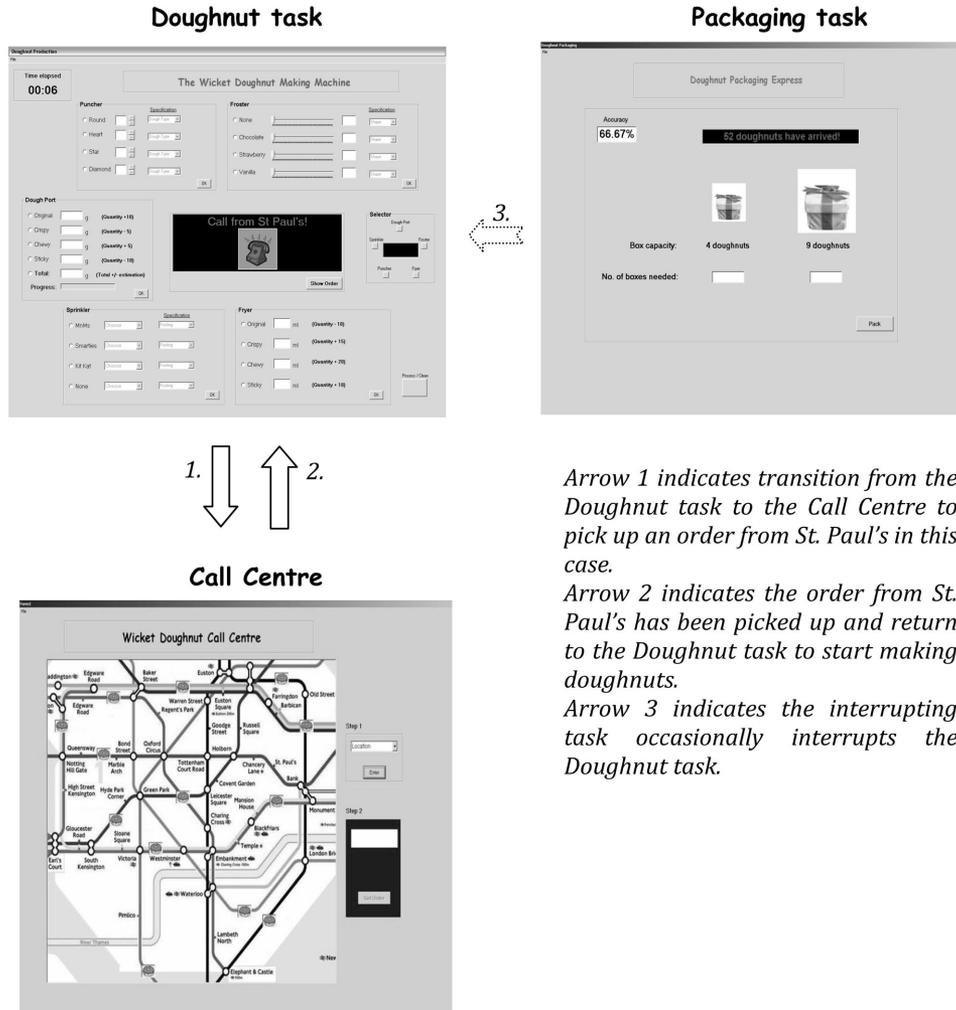


Figure 2. Transitions between the Doughnut task and the Call Centre. This figure has been previously published as: Li, S. Y. W., Cox, A. L., Blandford, A., Cairns, P., Young, R. M., & Abeles, A. (2006). Further investigations into postcompletion error: The effects of interruption position and duration. *In Proceedings of the 28th Annual Meeting of the Cognitive Science Conference*, pp. 471–476.

pack a certain number of doughnuts following some simple rules. When an interruption occurs during the Doughnut task, the computer screen switches to the Packaging task (see Figure 2, arrow 3). Participants then have to perform as many packaging trials as possible for 75 seconds until the screen switches back to the Doughnut task. After the interruption, participants have to continue the Doughnut task from where they left off.

Design

This was a within-subject design with one independent variable—interruption position—which had three levels: Z (just before the PC step), Other (positions P, Q, R, S, or T in Table 1), and Nil (no interruption). Each experimental session consisted of 11 trials in the testing phase: 4 trials with interruption position Z, 4 with interruption position Other, and 3 with no interruption.

Materials

The programs were written in Visual Basic 6. The Doughnut task program and the Packaging task program were run on a HP desktop and the Call Centre was run on a Compaq desktop.

Participants

Thirty-five participants, undergraduate and postgraduate students, took part in this study. Ages ranged from 19 to 37 with a mean of 24.8. There were 20 females and 15 males. Participants were paid £6 for their participation.

Procedure

In the demonstration phase, participants observed the experimenter performing both the Doughnut task and the interrupting

task separately. When performing the Doughnut task, the experimenter explained the need to respond to the Call Centre when the location signal flashes in the doughnut machine, demonstrated how to respond to a call using the Call Centre, and reminded participants that the doughnut machine needed to be cleaned after each completed order, before responding to the Call Centre.

In the training phase, the participant was given two training trials on the Doughnut task: one with and one without the interrupting task. Any errors in this training phase resulted in on-screen warning messages and beeps; participants were required to identify and correct the error to continue. Participants were encouraged to perform as quickly and accurately as possible.

In the testing phase, any error made resulted in a beep, but no on-screen message. The entire experiment took approximately 50 minutes.

Measures

Two dependent measures were of primary interest: first, the number of PCEs made; second, errors at other steps in the doughnut task. The error-counting scheme is the same as in Byrne and Davies (2006): When multiple incorrect mouse clicks were made consecutively on the same task step they were counted as one error.

Results

Data from four participants were excluded from the analysis. Two of them were making the PCE on every trial, suggesting that the PCEs made might have been because of incorrect knowledge of the task rather than procedural errors. One did not follow the task instructions properly and for one the data file was lost.

Distribution of All Errors

There were a total of 330 errors across the 31 participants. Over half of the participants (20 out of 31) made at least one PCE and a total of 56 PCEs were made. Figure 3 shows the distribution of all error rates across all task steps.

There were 274 errors at the non-PC steps in the task; about 77% (212 out of 274) of them occurred at one of the five “Selector” steps (i.e., Step 2, 6, 9, 12, and 15). Errors at Step 2 (“Selector DP”) are excluded from the following analyses (but discussed separately later) because there was no interruption manipulation before this step. Step 5 involves waiting for a progress bar to fill up and an error can occur when an action is made during the wait. This error suggests that participants were being impatient rather than making a slip, so it is not pursued further in the analysis.

Systematic Errors

One way to interpret the data is assessing the error against its systematicity level (e.g., Byrne & Bovair, 1997; Payne & Squibb, 1990). Systematicity, also referred to as “error rate,” is defined as a ratio of number of occurrences to the number of opportunities for that error. We chose to analyze only those errors that yielded a systematicity score of at least 5%, which is indicated by the dash line in Figure 3 (see also Byrne & Bovair, 1997). There were 11 trials for each of the 31 participants in the current study, giving a total of 341 opportunities for an error at each task step.

According to the systematicity definition, PCEs are highly systematic, with an error rate of 16%. Other non-PCEs occurring at the four “Selector” steps (i.e., Step 6, 9, 12, and 15) are also systematic. An examination of the error patterns at the four “Selector” steps identified a number of error kinds, as discussed above. As well as sub-task sequence errors and sub-task initializa-

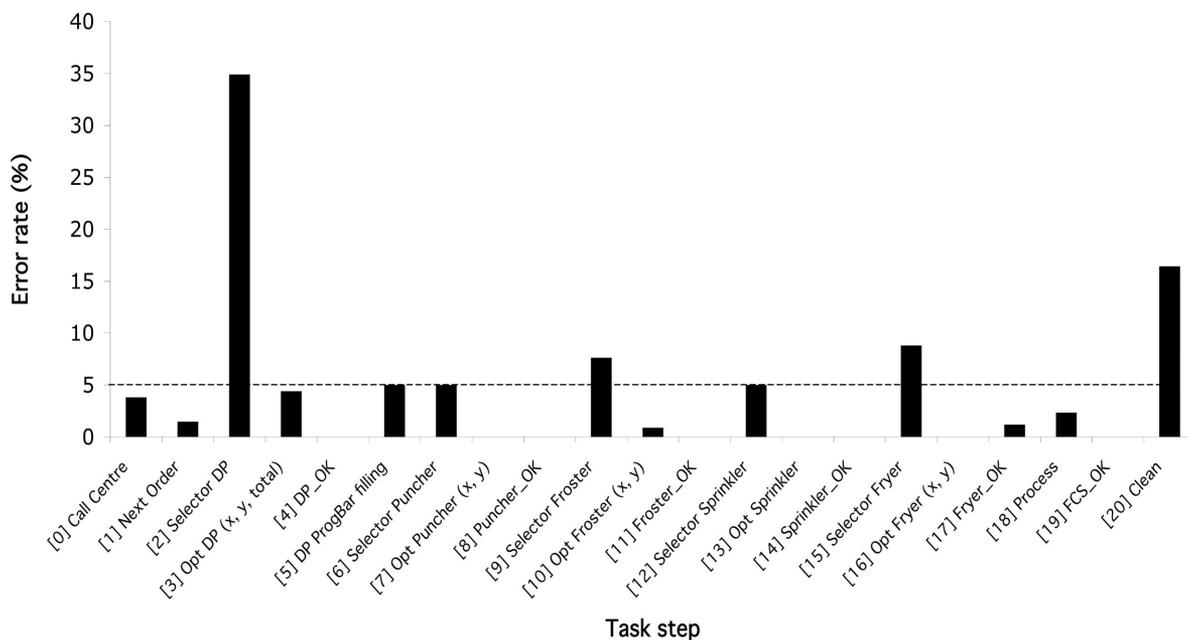


Figure 3. Error rate of each task step in Experiment 1. The dotted line indicates the 5% systematicity score.

tion errors, there were also errors that combined both of these features, such that participants both incorrectly remembered the main sequential task step and forgot the initial selecting step using the “Selector.”

There were a total of 172 sub-task initialization errors, 33 sub-task sequence errors and 7 combination errors. These identified systematic errors are analyzed in terms of the effect of interruption position in the following sections.

Interruption Position Effect on PCEs

The upper panel of Table 2 shows the distribution of PCE occurrences according to the different interruption positions. Error rates (%) of PCE occurrences were calculated for the three different interruption trials for each participant.

A one-way repeated measures ANOVA on the error rates showed a significant main effect of interruption position. The scores did not conform to the assumption of sphericity, so the Greenhouse-Geisser correction was used, $F(1.338, 40.149) = 9.921$, $p = .001$, $\eta_p^2 = 0.249$. Planned contrasts comparing position Z to position Other showed a reliable difference, $t(30) = 3.297$, $p = .003$, $d = .79$. Positions Z versus Nil also showed a significant difference, $t(30) = 3.339$, $p = .002$, $d = .79$. There was no reliable difference between positions Other and Nil, $t(30) = .09$, $p = .929$.

Categorizing Interruption Positions for the non-PCEs

To assess the effect of interruption positions for sub-task sequence errors and sub-task initialization errors, these errors were categorized into “immediately after interruption”, “later after interruption” and “no interruption” (including errors before an interruption). These are functionally equivalent to the PCE interruption positions Z, Other, and Nil, respectively. The description of each of the interruption categories is as follows.

Table 2
The Number and Mean Error Rates of PCEs and Sub-Task Sequence Errors According to the Different Interruption Positions (Experiment 1)

	Interruption position		
	Z (Immediately after)	Other (Later after)	Nil (No interruption)
Total no. of PCEs (Total no. of opportunities)	37 (124)	11 (124)	8 (93)
Mean error rate (SD)	29.8% (32.5)	8.9% (19.9)	8.6% (21.0)
Total no. of sub-task sequence errors (Total no. of opportunities)	28 (124)	2 (250)	8 (1331)
Mean error rate (SD)	22.6% (22.7)	1.1% (4.2)	0.6% (1.8)

Note. The PCEs data (upper panel) has been previously published as: Li, S. Y. W., Cox, A. L., Blandford, A., Cairns, P., Young, R. M., & Abeles, A. (2006). Further investigations into postcompletion error: The effects of interruption position and duration. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Conference*, pp. 471–476.

Immediately after interruption. An incorrect action performed immediately after interruption position P, Q, R, S, or T was classified as an error in this interruption category. Although after interruption position T the following “Process” step (i.e., Step 18) is not one of the “Selector” step, it was included in the analysis because a sub-task sequence error could still occur at this step.

Later after interruption. For interruption position P, Q, and R, any incorrect action performed at the *second* “Selector” step in relation to the interruption was classified as an error *later after* an interruption. For interruption position S, an incorrect action performed at the “Process” step was classified as an error in this interruption category.

No interruption. An incorrect action performed on any of the “Selector” steps or the “Process” step before an interruption, or in trials with no interruptions, was classified as an error in this interruption category.

Interruption Position Effect on Sub-Task Sequence Errors and PCEs

The first kind of non-PCE to examine is the sub-task sequence error. Combination errors were also included in the analysis. The occurrence of combination error was minimal (2.5%; 7 out of 274 non-PCEs) and, because they involved incorrect execution of the main sequential task step, they were included in the analysis of sub-task sequence errors.

There were a total of 40 sub-task sequence errors (including seven combination errors). Two of them occurred at the “Selector DP” step and were therefore excluded from the analysis. As a result, a total of 38 sub-task sequence errors were analyzed. An error rate for each interruption position category was computed for each participant. The lower panel of Table 2 shows the number and mean error rates for sub-task sequence errors in relation to different interruption positions.

Error rates of sub-task sequence errors and PCEs were subjected to a 2×3 (error type \times interruption position) repeated measures ANOVA yielding a significant main effect of interruption position, $F(1.253, 37.582) = 30.903$, $p < .001$, $\eta_p^2 = .507$, with Greenhouse-Geisser correction. The main effect of error type was also significant, $F(1, 30) = 5.391$, $p = .027$, $\eta_p^2 = .152$. The interaction was found not to be significant, $F(1.259, 37.756) = .007$, $p = .956$, with Greenhouse-Geisser correction. This suggests that there was no difference between PCEs and sub-task sequence errors with regards to the interruption effect.

A simple effects analysis on sub-task sequence errors yielded a significant effect of interruption position, $F(1.06, 31.789) = 26.951$, $p < .001$, $\eta_p^2 = .473$. Post hoc t test comparisons with Bonferroni corrections revealed a reliable difference between “immediately after” versus “later after,” $t(30) = 5.085$, $p < .01$, $d = 1.34$ and “immediately after” versus “no interruption,” $t(30) = 5.411$, $p < .01$, $d = 1.39$, but no reliable difference between “later after” versus “no interruption,” $t(30) = 0.559$, $p = .581$. The results suggest that interruption position has an effect on the occurrences of sub-task sequence errors.

Interruption Position Effect on Sub-Task Initialization Errors

The second kind of error to examine is the sub-task initialization error. Excluding those that occurred at the “Selector DP” step, as

Table 3
Number and Mean Error Rates of Sub-Task Initialisation Errors According to Different Interruption Position (Experiment 1)

	Interruption position		
	Immediately after	Later after	No interruption
Total no. of sub-task initialization error	3	9	45
(Total no. of opportunities)	(99)	(151)	(1114)
Mean error rate	3.2%	6.0%	4.0%
(SD)	(10.0)	(12.7)	(5.0)

discussed above, there were 57 such errors, which were analyzed in terms of the interruption categories. An error rate for each interruption position category was computed for each participant. Table 3 below shows the number and the mean error rates for sub-task initialization errors.

Interruption was not found to increase the error rate as it did for PCEs and sub-task sequence errors. A one-way repeated measures ANOVA yielded no significant effect of interruption position, with Greenhouse-Geisser correction, $F(1.676, 50.265) = 0.814$, $p = .429$.

Discussion

The current PCE rate obtained from trials without interruptions is 8.6% (8 out of 93 opportunities), which is comparable to the 9.3% (13 out of 140 opportunities) obtained in Byrne and Bovair's (1997) Experiment 1, which did not have working memory load manipulation. We can claim with reasonable confidence that the current paradigm has successfully generated PCEs at a level that allows investigation in a laboratory setting.

As predicted, the results show that a PCE is more likely to occur when the task is interrupted *just before* the execution of the PC step than any other position in the task. This finding can be explained in terms of Altmann and Trafton's AGM framework: When there was no interruption, or an interruption occurring earlier in the task, upon task resumption there are still remaining procedures to be carried out, and the step preceding the PC step provides associative priming to the PC action, so the PC error rate is lower than that with an interruption immediately before this step. The correct execution of the PC action depends on preserving the associative link between the pre-PC and PC steps.

On the other hand, an interruption just before the PC step disrupts the associative priming and upon task resumption the goal of executing the PC step may have decayed below retrieval threshold. Moreover, the task environment indicates that the task has been completed, with a signal prompting the participant to collect the next order from the Call Centre, so that the goal of moving on to the next task competes with the goal of executing the PC step.

Considering non-PCEs, it was found that sub-task initialization errors were not sensitive to interruption position, whereas sub-task sequence errors were. Like PCEs, sub-task sequence errors were more likely to occur after an immediately preceding interruption than after interruptions occurring earlier in the task. In terms of the AGM framework, the effect of interruption position on PCEs and sub-task sequence errors suggests that the correct executions of the

PC and main sequential task steps depend on whether the associative links between consecutive task steps in memory remain intact or not. When the associative links are intact, execution of a task step can then prime the execution of the next task step. The finding that PCEs and sub-task sequence errors are sensitive to interruptions suggests that these associative links can be disrupted by interruptions.

In contrast to PCEs and sub-task sequence errors, the occurrence of sub-task initialization error was found to be independent of the effect of interruption position. In terms of the AGM framework, the result suggests that the correct execution of sub-task initialization steps (i.e., activating the corresponding selector) may not depend on associative priming from their preceding step and that the procedural representations of these errors are not disrupted by interruptions. This further suggests that the procedural representations of the sub-task initialization steps might not be an integral part of the main sub-task sequence. As noted above, about two thirds of the sub-task initialization errors occur at the first compartment ("Selector DP") and only a very small portion of the sub-task initialization errors involves executing the incorrect sequential task step, that is, the combination errors. The pattern in the occurrence of sub-task initialization errors suggests that the participants remembered the correct main sub-task sequence even when they erroneously omitted the sub-task initialization step.

The finding of the dissociation between error types in terms of the effect of interruption provides support for the postulation that not all task step executions rely solely on the associative cueing mechanism. The distinction of sub-task sequence and sub-task initialization steps is similar to the distinction of task knowledge and device knowledge in HCI (e.g., Cox & Young, 2000; Gray, 2000). Cox and Young describe task knowledge as "how to complete a task using a particular device" and device knowledge as "a collection of facts about what the device as a whole (or parts of it) do." The correct execution of sub-task sequence steps then refers to task knowledge and the correct execution of sub-task initialization steps refers to device knowledge. The sub-task initialization steps are "extra" steps imposed by the device rather than steps that are integral to the "natural" task sequence. The executions of the PC step and the sub-task sequence steps depend on the associative cueing mechanism, whereas the execution of sub-task initialization steps might depend on a more deliberate and less automatic mechanism. This less automatic mechanism might involve the deliberate recall of facts about certain parts of a device: in this case, activate a compartment before it can be used.

The associative mechanism proposed by the AGM model accounts for the PCEs and sub-task sequence error results; however, it does not account for the sub-task initialization errors result. This is because the AGM model assumes each step is associatively linked to its neighboring steps but does not differentiate between different types of task steps. The current results suggest that different error types can respond differently to interruptions and this is because of different underlying procedural representations.

Experiment 2: A Change to the PC Step

The second experiment examined the effect of a task structure change of the Doughnut task on PCEs: making the PC step less practiced than other task steps. This manipulation is to mimic a

“cyclic” task structure present in Byrne and Bovair’s Phaser task, which is described below.

In the Phaser task used in Byrne and Bovair’s study, participants were required to repeat all the steps in the task, except the PC step, until the task is completed. We term this task structure “cyclic” because until a certain part of the task is completed, the participant remains in a “loop” cycling through a set of procedures repeatedly. The PC step is executed *only* when the task has been completed, which means that the execution of the PC step is relatively less well practiced compared to other steps in the task. Having a less-well-practiced PC step might be one of the contributing factors to the effectiveness of the Phaser task in provoking PCEs.

In the second experiment, the same primary task (the Doughnut task and Call Centre), interruption task (the Packaging task), and interruption duration (75 seconds) were used as in Experiment 1. Only one modification was made to the Doughnut task: The requirement of the PC action (cleaning the doughnut machine after each order) was *conditional* upon the presence of a notification signal at the beginning of each trial. The notification of the need to clean the doughnut machine appeared in one third of the total number of trials; as a consequence, the need to carry out the PC step was less frequent relative to other steps in the task.

It was expected that, with all else being equal, the effect of interruption position on PCEs and non-PCEs would be replicated. Furthermore, in the previous experiment, the interaction (position \times error type) between PCEs and sub-task sequence errors was not significant. The manipulation of a conditional PC step is to test if the relevant interaction becomes significant by weakening the associative link between the pre-PC step and the PC step through reduced practice. A significant interaction (position \times error type) between PCEs and sub-task sequence errors would provide further support that these two error types are dependent on an associative cueing mechanism.

Method

Tasks

As noted above, the tasks used in this experiment were identical to the ones used in Experiment 1, except that the PC step (clicking the Clean button) was only required when specified at the beginning of one third of the trials.

In a trial requiring the PC step (which we refer to as a “PC trial”), a notification saying “*There is some gunk in the machine, please clean it after this order.*” was displayed after the “Show Order” button was clicked. The notification is a pop-up dialog box that requires the participants to click on an “OK” button to acknowledge it.

Design

The basic design of the current experiment was the same as Experiment 1 in that it was a within-subject design with one independent variable—interruption position—which had three levels: Z (just before the PC step), Other (position P, Q, R, S, or T), and Nil (no interruption). However, the current experimental design differed in terms of its trial composition. Each experimental session consisted of 18 trials in the testing phase: 6 PC trials (2 trials with interruption position Z, 2 trials with position Other,

and 2 trials with no interruption) and 12 non-PC trials (2 trials with interruption position Z, 2 trials with position Other, and 8 trials with no interruption). The order of these trials was randomized.

Participants

Thirty-seven participants, either university undergraduate or postgraduate students, took part in this study. Ages ranged from 21 to 48 with a mean of 26.2. There were 18 females and 19 males. None of the participants had taken part in the previous experiment. Participants were paid £6 for their participation.

Procedure

The procedures were very similar to Experiment 1 except that participants were given three training trials (rather than the two used previously). One of these was with and two without the interrupting Packaging task. The two training trials without interruption were one non-PC and one PC trial; the trial with interruption was a PC trial.

Materials and Measures

The materials used and the measures taken were the same as in Experiment 1.

Results

Three participants’ data were excluded from analysis. One participant did not perform the task according to the instructions, one participant terminated the experiment without completing the entire session and the data file for one participant was lost.

The results are presented and analyzed separately for PC and non-PC trials

Analysis of PC Trials

Distribution of all errors. A total of 245 errors were obtained in the PC trials across the 34 participants. As there were six PC trials for each participant, this gives 204 error opportunities for each task step. Figure 4 shows the error rate of each individual task step.

Systematic errors. One of the errors at the PC step, after interruption Z, involved a participant attempting to repeat one of the steps in the Doughnut task. This error pattern did not conform to the operational definition of a PCE and was excluded from the analysis.

The pattern of systematic errors occurring across the task steps is consistent with the pattern from Experiment 1. More than half of the participants (24 out of 34) made at least one PCE, yielding a total of 56 PCEs. This gives PCEs at the “Clean” step an error rate of about 27% (see Figure 4). There were a total of 188 non-PCEs; most of the errors that have a systematicity score above the 5% criterion were errors at one of the five “Selector” steps. The exception is step 13 (“Selector Sprinkler”), which had a systematicity score just below criterion. Step 19 (“Process”) also reached systematicity. As in the previous experiment, errors at Step 3 (“Selector DP”) and Step 6 (“DP ProgBar filling”) also yielded systematicity but are not further analyzed.

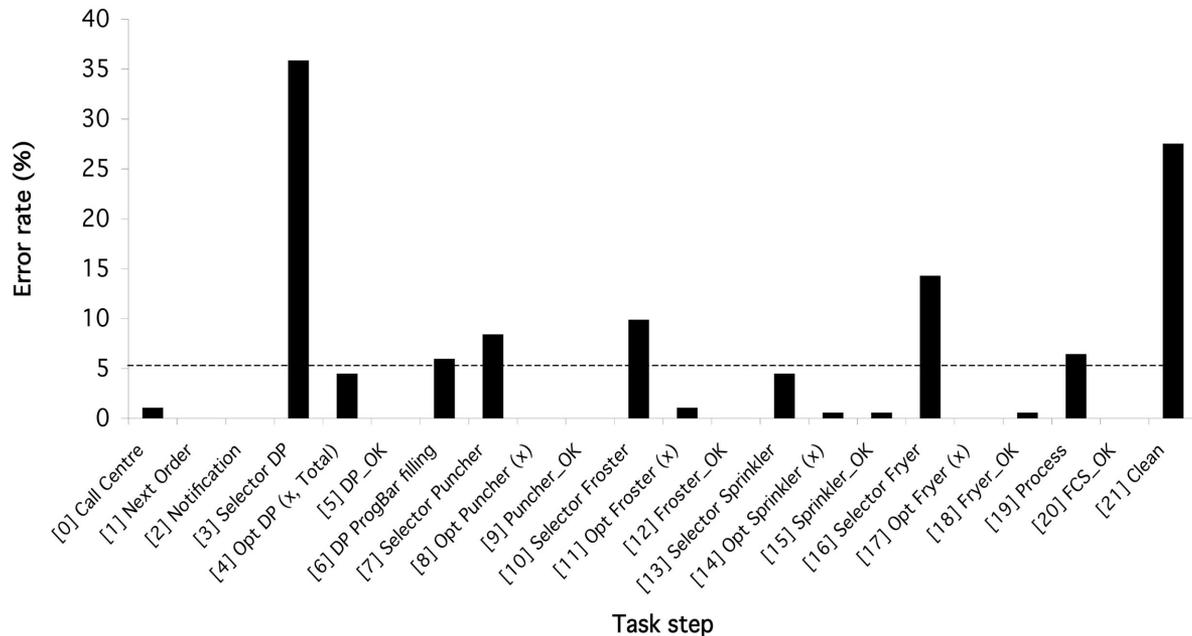


Figure 4. Error rate of each task step in PC trials in Experiment 2. The dotted line indicates the 5% systematicity score.

The identified systematic errors were categorized into PCEs, sub-task sequence errors, sub-task initialization errors and combination errors as in the previous experiment, and analyzed in terms of the effect of interruption position in the following sections. There were a total of 113 sub-task initialization errors, 43 sub-task sequence errors and 2 combination errors.

Interruption Position Effect on PCEs

The upper panel of Table 4 shows the number and error rate of PCEs in relation to the different interruption positions. Each participant received two trials for each interruption position; the total number of opportunities is computed across all 34 participants.

A one-way repeated measures ANOVA yielded a significant main effect of interruption position, $F(2, 66) = 4.938, p = .01, \eta_p^2 = .13$. Planned comparisons between positions Z and Other showed a reliable difference, $t(33) = 2.61, p = .014, d = .64$, and the difference between positions Z and Nil was also significant, $t(33) = 2.244, p = .032, d = .48$. Comparison between positions Other and Nil did not show a reliable difference, $t(33) = -.941, p = .353$. The overall pattern of results is consistent with the previous experiments.

Interruption Position Effect on Sub-Task Sequence Errors and PCEs

As in the previous study, combination errors were included in the analysis of sub-task sequence errors, yielding a total of 45. Five sub-task sequence errors occurred at "Selector DP" but are excluded from the analysis because they could not be preceded by an interruption. The lower panel of Table 4 shows the number and mean error rates of sub-task sequence errors in relation to the interruption position.

A 2×3 (error type \times interruption position) repeated measures ANOVA was performed on the error rates of sub-task sequence errors and PCEs. A reliable main effect of interruption position was found, $F(1.439, 47.501) = 30.1, p < .001, \eta_p^2 = .477$ (with Greenhouse-Geisser correction). A significant main effect of error type was also found, $F(1, 33) = 5.588, p = .024, \eta_p^2 = .145$. The interaction between error type and interruption position was also found to be significant, $F(1.421, 46.905) = 5.381, p = .015, \eta_p^2 = .14$ (with Greenhouse-Geisser correction). The significant interaction reflects a difference between sub-task sequence errors and PCEs in terms of the effect of interruption. The data show that there were higher rates of PCEs than sub-task sequence errors in

Table 4
The Number and Mean Error Rates of PCEs and Sub-Task Sequence Errors According to the Different Interruption Positions (Experiment 2)

	Interruption position		
	Z (Immediately after)	Other (Later after)	Nil (No interruption)
Total no. of PCEs	28	12	16
(Total no. of opportunities)	(68)	(68)	(68)
Mean error rate	41.2%	17.6%	23.5%
(SD)	(43.5)	(30.0)	(30.7)
Total no. of sub-task sequence errors	33	0	7
(Total no. of opportunities)	(68)	(124)	(828)
Mean error rate	48.5%	0%	0.9%
(SD)	(45.2)	(0)	(2.2)

Table 5
Number and Mean Error Rates of Sub-Task Initialisation Errors According to Different Interruption Position (Experiment 2)

	Interruption position		
	Immediately after	Later after	No interruption
Total no. of sub-task initialization errors	1	7	28
(Total no. of opportunities)	(49)	(73)	(554)
Mean error rate	1.8%	11.5%	4.9%
(SD)	(9.4)	(24.1)	(8.2)

“Later after” and “No interruption.” This difference suggests that PCEs were more likely to occur than sub-task sequence errors when not influenced by an immediately preceding interruption.

A nonparametric test was used in analyzing the effect of interruption position on sub-task sequence errors because of the occurrence of zero variance in one of the interruption position cells. Post hoc comparisons using Wilcoxon signed-ranks test, with Bonferroni corrections, obtained a reliable difference between “immediately after” versus “no interruption,” $Z = 4.069, p < .001$.

Interruption Position Effect on Sub-Task Initialization Errors

More than half of the sub-task initialization errors (68 out of 113) occurred at “Selector DP” and these errors are excluded from the following analysis, for the same reason as before. Table 5 shows the number and mean error rate of sub-task initialization errors with respect to the interruption positions.

Six participants’ data were excluded for sub-task initialization error analysis as they had no opportunity to make an error in the “Later after” interruption category. This is because of the random sampling of interruption positions: These participants had interruptions at positions S and T. A one-way repeated measures ANOVA was performed and found a marginal effect of interruption position, $F(1.271, 34.307) = 3.349, p = .067, \eta_p^2 = .110$ (with Greenhouse-Geisser correction). However, the data in Table 5 show that sub-task initialization errors were *not* increased by immediately preceding interruptions.

Analysis of non-PC Trials

Distribution of all errors. Among the non-PC trials, there were 368 errors in total across the 34 participants. There were 12 non-PC trials for each participant; therefore, each task step has 408 opportunities for an error to occur. Figure 5 below shows the distribution of the error rates across all task steps.

Systematic errors. The pattern of error rates across all task steps has the same overall pattern as the PC trials; most errors that reached the 5% systematicity criterion occurred at the one of the five “Selector” steps. As before, Step 2 (“Selector DP”) and Step 5 (“DP ProgBar filling”) also have error rates above the criterion but they are not considered further.

Errors at the PC step also appear to be systematic; although they occur at the PC step, they are not PCEs according to the definition operationalized in the task. This error can be thought of as “extra Cleaning” (“xClean”) error as it was the execution of the “Clean” step that was not required in the first place, that is, false alarm; it is an error of commission. There were 45 xClean errors in total.

Seven of the errors at the PC step, after interruption Z, involved participants attempting to repeat one of the steps in the Doughnut

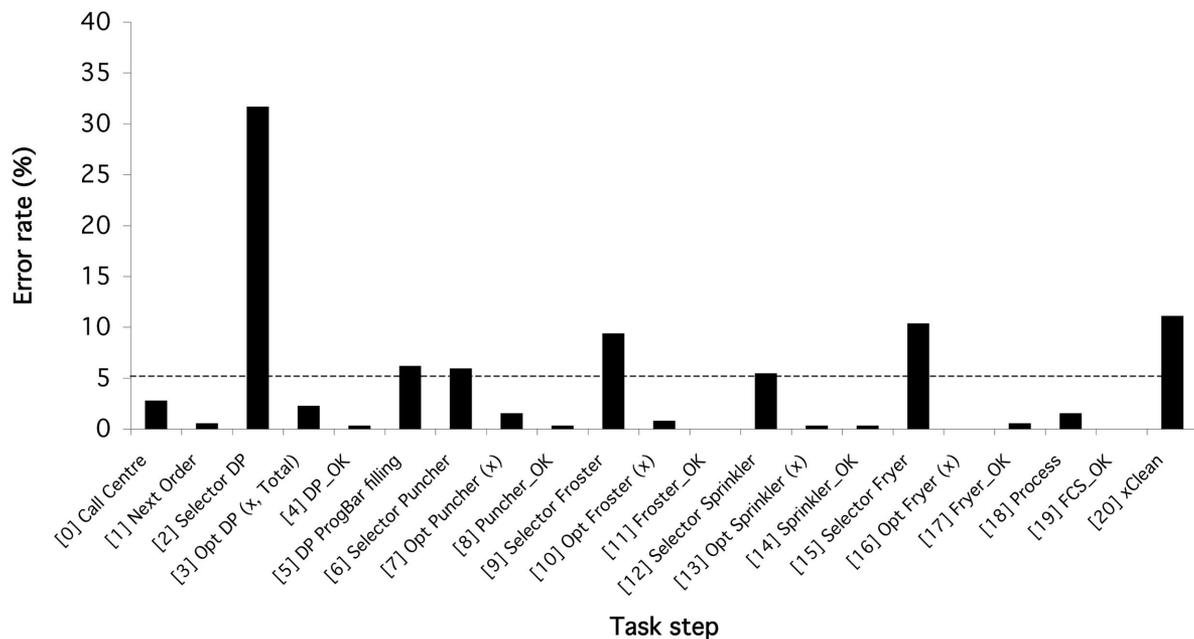


Figure 5. Error rate of each task step in non-PC trials in Experiment 2. The dotted line indicates the 5% systematicity score.

task. As before, these errors were excluded from the analysis because their error patterns did not conform to the operational criteria of xClean errors.

Interruption position effect on xClean errors. Table 6 shows the number and mean error rates in relation to the interruption positions. A one-way repeated measures ANOVA on the error rate did not find a significant effect of interruption position, $F(2, 66) = 1.129, p = .329$. These errors will be discussed in more detail later.

Interruption position effect on sub-task initialization and sub-task sequence errors. There were a total of 316 non-PCEs and the same procedures of categorization and analysis were carried out on these errors. The same patterns of results were obtained: A reliable effect of interruption position on sub-task initialization errors ($F(1.224, 37.94) = 4.045, p = .044, \eta_p^2 = .115$, with Greenhouse-Geisser correction) and significant interruption position effect on sub-task sequence errors ($F(1.001, 33.022) = 78.832, p < .001, \eta_p^2 = .705$, with Greenhouse-Geisser correction). The patterns of results are consistent with those of the PC trials, so details of the data are not presented here.

Discussion

The second experiment also successfully generated an overall level of PCEs (about 27%) that allows further investigations of the error with respect to the effect of interruption. The main finding of the effect of interruption position on PCEs has been replicated in the current experiment, as has the finding that sub-task sequence errors are sensitive to the disruptive effect of immediately preceding interruptions, whereas sub-task initialization errors are not. These results suggest that correct executions of the PC step and the main sub-task sequential steps depend on associative priming that takes place through intact associative links between consecutive task steps, whereas the correct execution of sub-task initialization steps may not depend on the same associative linkage mechanism from their preceding task step.

The effect of interruption position on sub-task initialization errors was found to be significant in both PC and non-PC trials; however, the data show consistent patterns with the previous experiment that sub-task initialization errors were *not* increased by immediately preceding interruptions. On the contrary, the data (see Table 5) show higher error rates in “Later after” and “No interruption” than “Immediately after.” It is not clear what contributes to the unexpected results; however, it is speculated that the conditional nature of the PC step might be a cause because of extra memory demand. Whatever the ultimate explanation is, the important result is that the occurrences of sub-task initialization

errors were not sensitive to immediately preceding interruptions, which is in contrast to sub-task sequence errors and PCEs.

In retrospect, the conditional nature of the PC step is quite different from the less practiced PC step in Byrne and Bovair’s Phaser task. The conditional PC step resembles a prospective memory task: Participants received a notification of whether to perform the PC step at the beginning of a trial, and had to maintain that goal in memory until they reached the particular task step (Kvavilashvili & Ellis, 1996). This is likely to have added memory load to performing the PC action, because the participants needed to remember whether the PC action is required to perform or not in a given trial.

The conditional nature of the PC step is likely to relate to two other findings in the current experiment. First, there is a significant interaction effect between error type (PCE or sub-task sequence error) and interruption position, which is contrary to the previous experiment’s finding. This suggests that PCEs and sub-task sequence errors are both increased by interruptions but PCEs are more likely to occur than sub-task sequence errors without immediately preceding interruptions. The average PCE rates that are not immediately preceded by interruptions are about 20% whereas the average sub-task sequence error rates that are not immediately preceded by interruptions are about 0.5% (see Table 4). This can be explained by the AGM in terms of different strengths in the associative links: The associative links formed between the PC step and its preceding step may be weaker than those formed between the main sequential task steps, because the conditional nature of the PC step only requires participants to perform the PC step in one third of the trials, whereas the main sequential task steps are performed in every trial. In other words, the weak associative link between the pre-PC and PC steps is a result of reduced practice.

Second, the conditional nature of the PC step gave rise to the error of commission—xClean. This error was found to be systematic and involved executing the PC step when it was not required. In terms of the AGM framework, the occurrence of xClean may be because of the formation of a weak associative link between the pre-PC and PC steps. It is important to note that priming can still occur through a weak associative link because of noise in memory retrieval. In other words, the actual activation value of a memory item can fluctuate around an expected value. Consequently, priming between the pre-PC and PC steps may still occur, despite a weak association, when the PC action is not required. The occasional occurrence of the error reflects the noisy nature of the associative priming process. However, the error was not found to be sensitive to the effect of interruption position, suggesting that associative priming may not be the sole mechanism responsible for this error. It is possible that the occurrence of xClean may also be because of other mechanisms, such as failure of self-reminding strategies or prospective memory failure.

General Discussion

Consistent results were produced in Experiments 1 and 2, showing that the rate of PCEs can be affected by manipulating where the interruption occurs in the primary task. The findings are consistent with the predictions made from Altmann and Trafton’s AGM model, that interruptions occurring just before the PC step in a task are more likely to result in a PCE than earlier interruptions,

Table 6
Number of xClean and Their Mean Error Rates With Respect to the Different Interruption Positions (Experiment 2)

	Interruption position		
	Z	Other	Nil
Total no. of xClean	8	11	26
(Total no. of opportunities)	(68)	(68)	(272)
Mean error rate	11.8%	16.2%	9.6%
(SD)	(24.8)	(26.7)	(16.9)

because of the disruption of associative priming from the previous task step. When attention is focused on the interrupting activity, the level of activation of the to-be-resumed goal undergoes a time-based decay process. When the interrupting activity is long enough for the to-be-resumed goal to decay below retrieval threshold, an incorrect goal is likely to be retrieved upon task resumption.

Although PCEs were found to be sensitive to the effect of interruption position, analyses of other errors in Experiments 1 and 2 suggest that the effect is not unique to PCEs. sub-task sequence errors are also susceptible to the disruptive effect of interruption. In terms of the AGM model, the execution of a sequence of proceduralized task steps depends on priming through intact associative links between task steps. Interruptions occurring between the proceduralized task steps can be thought of as disruptions that break the associative links. This suggests that priming through intact associative links is crucial for the correct executions of the PC step and the sub-task sequential steps. This priming mechanism in procedural task execution is similar to the traditional chaining approach in serial recall in basic memory research (Lewandowsky & Murdock, 1989), in which neighboring memory items in a list are associatively linked to each other.

However, this does not mean that PCEs and sub-task sequence errors are the same kind of procedural error: this is supported by the reliable main effect of error type obtained in Experiments 1 and 2. Furthermore, the two kinds of errors have a number of structural differences: First, PCEs occur at the end of a task *after* the overall main goal is completed. Second, there is often a false completion signal upon completion of the overall main goal (Reason, 2002); in the case of a photocopying task, a false completion signal would be the collection of the desired photocopies, which gives a sense of having completed the main task. This completion signal is false because there is a remaining “clean up” step to be done until the task is fully completed. Third, a subsequent task is usually present requiring one to move away from the main task. For example, a subsequent task following a photocopying task might be to disseminate the photocopies to students. In contrast, errors that involve executing the wrong sequential task step, that is, sub-task sequence errors, occur *before* the overall main goal is accomplished and there is neither a false completion signal nor a new subsequent task waiting to be executed.

The analyses of other procedural errors also reveal the presence of another kind of omission error—a sub-task initialization error—and a commission error—xClean. These errors were found to be unaffected by the disruptive effect of interruptions. This suggests that the procedural representations of these two kinds of errors may be different from associative priming between neighboring task steps.

In terms of error behavior, sub-task initialization errors suggest a very different manifestation to sub-task sequence errors. When a sub-task sequence error is committed, the participant is executing a wrong sequential task step in the procedure but the initial “selecting” step is not forgotten; whereas the occurrence of a sub-task initialization error suggests that the participant is remembering the main sequential task steps correctly but forgetting the initial “selecting” step. The dissociations between sub-task sequence errors and sub-task initialization errors in terms of error behavior and sensitivity to interruptions indicate that they may have different procedural representations. The representation of

sub-task initialization errors might be declarative in nature in that execution of a sub-task initialization step requires deliberate recall of facts about the device in hand, whereas other task knowledge appear to be procedural.

The error of commission—xClean—may depend on a combination of different procedural representations. In terms of the AGM model, the conditional nature of the PC step may help to form an associative link between the pre-PC and PC steps but the association is weak because the PC step is not always carried out. The occurrence of xClean may be because of the presence of the weak associative link that primes the PC step. However, the commission of this error is found to be independent of interruptions, and the nature of the manipulation in Experiment 2 resembles that of a prospective memory task. Therefore, the occurrence of this commission error may also be because of failures in other mechanisms such as maintaining a prospective memory element.

The high occurrence of sub-task initialization errors at the very first sub-task initialization step (“Selector DP”) is an unanticipated finding. One might argue that the operation of the follow-on task between trials, that is, the Call Centre task, could act as an interruption and that the sub-task initialization errors at the “Selector DP” step could be explained by the effect of interruption. However, this does not explain why the sub-task initialization errors at the other “Selector” steps are not increased by interruptions. Although it is out of the scope of the current study to provide an explanation, it could be speculated that higher working memory demand might be associated with the beginning of the task compared to later in the task because of more competing goals. Further experiments are being designed to examine whether there is higher working memory demand at the “Selector DP” step and whether the high sub-task initialization error rate is related to working memory demand. Work by Hiltz, Blandford, & Back (submitted) shows that this error is robust to interventions that involve adapted instructions to participants (e.g., changing the user task or providing a rich mental model of the purpose of the initial device step): The only intervention of the three tested that had a significant effect on error rates was to instruct participants to enter (specified) arbitrary values for all domain parameters. Anticipating future parameters is one contribution to working memory load, which was reduced through this particular intervention.

One of the limitations of the current study is that it is not possible to distinguish the effect of disruptiveness of those interruptions occurring early in the task in relation to PCEs or other procedural errors. It is possible that an interruption could have a graded effect on error occurrences, that is, an interruption occurring at Step n , before a particular task step (e.g., the PC step), may have a more disruptive effect and more likely to cause an error to occur than an interruption occurring at Step $n - 1$. The differentiation of error patterns because of relative positions has been observed in basic memory research of serial list memory (e.g., Lewandowsky & Murdock, 1989).

Connectionist models based in distributed associative memory have been proposed to account for cognitive sequential behaviors such as serial list memory recall. In the same spirit as the AGM model, these serial list memory models propose that memory items in a recall list are associated in a pairwise manner (e.g., item A – item B, item B – item C, etc.). More recent developments in serial list memory models have been proposed to account for a commonly observed error pattern called movement gradient (Brown,

Preece, & Hulme, 2000; Farrell & Lewandowsky, 2002), in which recall errors in a list are more likely to occur in nearby memory items than memory items that are further apart. For example, ABCD is more likely to be incorrectly recalled as ACBD than ADCB. This error pattern in serial recall is explained by a gradient of differentiating levels of activation among different memory item positions.

Altmann and Trafton (2007) extend the development of the AGM model and propose a gradient of differentiating strength among associative links between sequential memory items. Altmann and Trafton observed that interruptions could have a disruptive effect beyond the first postinterruption response. It was found that not only the first response takes time to resume immediately after an interruption but subsequent actions also need time to recover fully from the disruption. Altmann and Trafton propose a simple formal model to explain the timecourse of recovery after an interruption. It was suggested that associative links between sequential memory items follow a strength gradient, which assumes memory items have stronger associative links to other nearby items than more distant items. Altmann and Trafton (2007) suggest that "These assumptions imply that when the first step of such a plan is retrieved, it will prime retrieval of the second step strongly and retrieval of the third step weakly." (p. 1081). According to this extension of the AGM model, interruption position may have a graded effect on error occurrences predicting that nearby steps after an interruption may be more likely to result in errors than more distant steps. Future research will have to examine this possible graded effect on error occurrences with more precisely controlled interruption positions.

As discussed above, it is a challenge to study human error in a laboratory setting. The task paradigm designed and used in this study makes a significant methodological contribution to the task repertoire in studying PCEs. Most experimental studies on PCEs have used Byrne and Bovair's Phaser task; we have shown that the same error phenomenon can be generated using a different task paradigm.

The AGM model serves as a useful theoretical framework allowing predictions to be made about the effect of interruption. Traditional studies in interruption research have not looked at how an interruption might affect specific kinds of error. The current series of experiments has furthered the understanding of PCEs and other errors in a procedural task by investigating the effect of interruption within an appropriate theoretical framework.

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.
- Altmann, E. M., & Trafton, J. G. (2007). Timecourse of recovery from task interruption: Data and a model. *Psychonomic Bulletin & Review*, 14, 1079–1084.
- Baars, B. J. (1992). The many uses of error: Twelve steps to a unified framework. In B. J. Baars (Ed.), *Experimental slips and human error: Exploring the architecture of volition*. New York: Plenum Press.
- 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.
- Brown, G. D. A., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological Review*, 107, 127–181.
- Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science*, 21, 31–61.
- Byrne, M. D., & Davis, E. M. (2006). Task structure and postcompletion error in the execution of a routine procedure. *Human Factors*, 48, 627–638.
- 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 *Proceedings of the Third International Conference on Cognitive Modelling* (pp. 70–77). Veenendaal, The Netherlands: Universal Press.
- Cutrell, E. B., Czerwinski, M., & Horvitz, E. (2000). Effects of instant messaging interruptions on computing tasks. *Proceedings of CHI2000* (pp. 99–100). New York: ACM Press.
- Edwards, M. B., & Gronlund, S. D. (1998). Task interruption and its effects on memory. *Memory*, 6, 665–687.
- Farrell, S., & Lewandowsky, S. (2002). An endogenous distributed model of ordering in serial recall. *Psychonomic Bulletin & Review*, 9, 59–79.
- Gillie, T., & Broadbent, D. (1989). What makes interruption disruptive? A study of length, similarity, and complexity. *Psychological Research*, 50, 243–250.
- Gray, W. D. (2000). The nature and processing of errors in interactive behaviour. *Cognitive Science*, 24, 205–248.
- Gray, W. D. (2004). Errors in interactive behavior. In W. S. Bainbridge (Ed.), *Encyclopedia of human-computer interaction* (pp. 230–235): Berkshire Publishing Group. Retrieved September, 2006, from http://www.rpi.edu/~grayw/pubs/papers/wdg04_HCI-Encycl.html
- Hiltz, K., Blandford, A., & Back, J. (submitted). The roles of mental models and user goals in avoiding device initialization errors. *Submitted for journal publication*.
- Hodgetts, H., M., & Jones, D., M. (2006a). Interruption of the Tower of London task: Support for a goal-activation approach. *Journal of Experimental Psychology: General*, 135, 103–115.
- Hodgetts, H., M., & Jones, D., M. (2006b). Contextual cues aid recovery from interruption: The role of associative activation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32, 1120–1132.
- Kvavilashvili, L., & Ellis, J. A. (1996). Varieties of intention: Some distinctions and classifications. In M. Brandimonte, G. O. Einstein, & M. A. McDaniel (Eds.), *Prosepective memory: Theory and applications*. Erlbaum: Mahwah, N. J.
- Latorella, K. A. (1999). *Investigating interruptions: Implications for flight-deck performance*. NASA/TM-1999-209707, National Aeronautics and Space Administration, Hampton.
- Lee, W. O. (1992). The effects of skill development and feedback on action slips. In A. Monk, D. Diaper, & M. Harrison (Eds.), *People and computers VII*. Cambridge: Cambridge University Press.
- Lewandowsky, S., & Murdock, B. B., Jr. (1989). Memory for serial order. *Psychological Review*, 96, 25–57.
- McFarlane, D. C., & Latorella K. A. (2002). The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction*, 17, 1–61.
- Monk, C. A., Boehm-Davis, D. A., & Trafton, J. G. (2002). The attentional costs of interrupting task performance at various stages. *Proceedings of the 46th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 1824–1828). Santa Monica, CA: HFES.
- Norman, D. A. (1981). Categorization of action slips. *Psychological Review*, 88, 1–15.
- Payne, S. J., & Squibb, H. R. (1990). Algebra mal-rules and cognitive accounts of errors. *Cognitive Science*, 14, 445–481.

- Ratwani, R. M., Trafton, J. G., & Myers, C. (2006). Helpful or harmful? Examining the effects of interruptions on task performance. *Proceedings of the 50th Annual Meeting of the Human Factors and Ergonomics Society*.
- Reason, J. (2002). Combating omission errors through task analysis and good reminders. *Quality of Safety and Health Care, 11*, 40–44.
- Reason, J. T., & Mycielska, K. (1982). *Absent-minded? The psychology of mental lapses and everyday errors*. Englewood Cliffs, NJ: Prentice Hall.
- Sellen, A. J., & Norman, D. A. (1992). The psychology of slips. In B. J. Baars (Ed.), *Experimental slips and human error: Exploring the architecture of volition*. New York: Plenum Press.
- Speier, C., Vessey, I., & Valacich, J. S. (2003). The effects of interruptions, task complexity, and information presentation on computer-supported decision-making performance. *Decision Sciences, 34*, 771–797.
- Trafton, J. G., Altmann, E. M., Brock, D. P., & Mintz, F. E. (2003). Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human-Computer Studies, 58*, 583–603.

Received October 4, 2007

Revision received September 17, 2008

Accepted September 25, 2008 ■

Call for Nominations: *Psychology of Violence*

The Publications and Communications (P&C) Board of the American Psychological Association has opened nominations for the editorship of *Psychology of Violence*, for the years 2011–2016. The editor search committee is chaired by William Howell, PhD.

Psychology of Violence, to begin publishing in 2011, is a multidisciplinary research journal devoted to violence and extreme aggression, including identifying the causes and consequences of violence from a psychological framework, finding ways to prevent or reduce violence, and developing practical interventions and treatments.

As a multidisciplinary forum, *Psychology of Violence* recognizes that all forms of violence and aggression are interconnected and require cross-cutting work that incorporates research from psychology, public health, neuroscience, sociology, medicine, and other related behavioral and social sciences. Research areas of interest include murder, sexual violence, youth violence, inpatient aggression against staff, suicide, child maltreatment, bullying, intimate partner violence, international violence, and prevention efforts.

Editorial candidates should be members of APA and should be available to start receiving manuscripts in early 2010 to prepare for issues published in 2011. Please note that the P&C Board encourages participation by members of underrepresented groups in the publication process and would particularly welcome such nominees. Self-nominations are also encouraged.

Candidates should be nominated by accessing APA's EditorQuest site on the Web. Using your Web browser, go to <http://editorquest.apa.org>. On the Home menu on the left, find "Guests." Next, click on the link "Submit a Nomination," enter your nominee's information, and click "Submit."

Prepared statements of one page or less in support of a nominee can also be submitted by e-mail to Emmet Tesfaye, P&C Board Search Liaison, at Emnet@apa.org.

Deadline for accepting nominations is January 31, 2009, when reviews will begin.