

An exploration of internal cues to reduce omission errors in a procedural task

Kimberley Deanne Kelley

Project report submitted in part fulfilment of the requirements for the degree of Master of Science (Human-Computer Interaction with Ergonomics) in the Faculty of Life Sciences, University College London, 2007.

NOTE BY THE UNIVERSITY

This project report is submitted as an examination paper. No responsibility can be held by London University for the accuracy or completeness of the material therein.

Abstract

Systematic errors that occur at the end of a procedure, called post completion errors, have been extensively reproduced and researched in lab settings. A related systematic error was recently reported by Li (2006) and occurs at the beginning of a procedure; this error will be referred to hereafter as the *device initialization* error. Li (2006) suggested that the device initialization error might occur when the initial procedural step does not directly move one toward their main task-based goal, resulting in it having lower relevance to the main goal and being attributed lower *cognitive salience*. This thesis investigates two different approaches to increasing the salience of the device initialization step without making physical changes to the device interface. In the first approach participants were supplied with a new task-based goal for the procedure, for which execution of the device initialization step was critical to successful accomplishment. It was hypothesized that the salience of the step would be increased due to its central role in fulfilling the new goal. A significant reduction in the error rate was observed only if the new task-based goal was the only goal acted on. This suggests that the salience of the device initialization step can be influenced by factors internal to the individual, but is sensitive to competing internal factors. In the second approach participants were supplied with a device model (Kieras & Bovair, 1984) describing the role of the device initialization step in terms of the device's internal mechanisms and its significance to successful operation of the device. It was hypothesized that training participants to operate the device based on such a device model would support development of a more complete mental model, and would lead to improved performance. While evidence of a change in participants' resulting mental representations was found, this did not translate to a reduction in the device initialization error rate. This indicates that the cognitive salience of the step was not enhanced as a result of the improved mental model. Implications for training and device design are discussed.

Table of contents

<u>ABSTRACT</u>	<u>2</u>
<u>TABLE OF CONTENTS</u>	<u>3</u>
<u>LIST OF FIGURES</u>	<u>5</u>
<u>LIST OF TABLES</u>	<u>6</u>
<u>1 INTRODUCTION</u>	<u>7</u>
<u>2 BACKGROUND ON THE DEVICE INITIALIZATION ERROR</u>	<u>9</u>
2.1 OVERVIEW OF THE TASK ENVIRONMENT AND TASK PROCEDURE	9
2.2 THE DEVICE INITIALIZATION ERROR: OMISSION OF THE DOUGH PORT SELECTOR STEP ..	11
2.3 UNDERSTANDING THE DEVICE INITIALIZATION ERROR IN TERMS OF DEVICE’S AND USER’S TASK STRUCTURES.....	12
<u>3 MOTIVATION FOR THE STUDY OF ERRORS</u>	<u>13</u>
<u>4 COMMON APPROACHES TO THE STUDY OF ERRORS</u>	<u>15</u>
4.1 NATURALISTIC STUDIES	15
4.1.1 NORMAN’S CLASSIFICATION SCHEME.....	15
4.1.2 REASON’S CLASSIFICATION SCHEME	16
4.1.3 SUMMARY OF NATURALISTIC STUDIES OF ERROR	17
4.2 EMPIRICAL STUDIES OF ERROR	18
<u>5 EXISTING STUDIES OF OMISSION ERRORS</u>	<u>19</u>
5.1 POSTCOMPLETION ERRORS	19
5.1.1 THE ROLE OF TASK STRUCTURE IN POSTCOMPLETION ERRORS	20
5.1.2 STUDIES OF POSTCOMPLETION ERRORS	21
5.1.3 A MEMORY THEORY APPROACH TO GOAL ACTIVATION: THE AGM MODEL	22
5.1.4 OBSERVED DIFFERENCES BETWEEN POSTCOMPLETION AND DEVICE INITIALIZATION ERRORS 26	
<u>6 THIS EXPERIMENT</u>	<u>27</u>
6.1 APPROACH 1: IMPROVING COGNITIVE SALIENCE BY MODIFYING THE TASK-BASED GOAL 27	
6.1.1 RELATED STUDIES ON GOALS AND PERFORMANCE.....	28
6.1.2 THE TASK-BASED GOALS USED IN THIS EXPERIMENT	29
6.2 APPROACH 2: IMPROVING COGNITIVE SALIENCE THROUGH MENTAL MODELS.....	30
6.2.1 RELATED STUDIES ON MENTAL MODELS AND PERFORMANCE.....	32

6.3	THE DEVICE MODEL USED IN THIS EXPERIMENT.....	33
<u>7</u>	<u>METHOD.....</u>	<u>34</u>
7.1	THE PROCEDURAL TASK ENVIRONMENT	34
7.2	PARTICIPANTS	38
7.3	MATERIALS.....	38
7.4	DESIGN	38
7.5	PROCEDURE.....	40
<u>8</u>	<u>RESULTS.....</u>	<u>42</u>
8.1	OVERALL ERRORS.....	42
8.2	DEVICE INITIALIZATION ERRORS.....	44
8.3	CATEGORIES OF ERRORS	45
8.4	QUALITATIVE RESULTS.....	47
<u>9</u>	<u>DISCUSSION.....</u>	<u>51</u>
9.1	APPROACH 1: IMPROVING COGNITIVE SALIENCE BY MODIFYING THE TASK-BASED GOAL	
	51	
9.1.1	INTERPRETATION OF TESTER RESULTS	52
9.1.2	INTERPRETATION OF THE TESTER-ENHANCED RESULTS	53
9.2	APPROACH 2: IMPROVING COGNITIVE SALIENCE THROUGH MENTAL MODELS.....	56
9.3	THE POTENTIAL ROLE OF MATH ANXIETY	57
9.4	IMPLICATIONS FOR TRAINING AND DEVICE DESIGN	59
<u>10</u>	<u>SUMMARY AND CONCLUSIONS.....</u>	<u>60</u>
	<u>REFERENCES.....</u>	<u>62</u>
	<u>APPENDIX 1.....</u>	<u>64</u>
	<u>APPENDIX 2.....</u>	<u>65</u>
	<u>APPENDIX 3.....</u>	<u>66</u>
	<u>APPENDIX 4 (I).....</u>	<u>68</u>
	<u>APPENDIX 4 (II).....</u>	<u>69</u>
	<u>APPENDIX 4 (III).....</u>	<u>71</u>
	<u>APPENDIX 5.....</u>	<u>73</u>

List of figures

- Figure 1. The Wicket Doughnut Call Centre interface used by Li (2006) to simulate a call centre for a doughnut-making operation. Participants used the interface to retrieve incoming orders from different customer locations..... 10
- Figure 2. The Wicket Doughnut Making Machine used by Li (2006) to simulate the production of doughnuts (the red letters have been added by this author for the purpose of identifying specific parts of the machine). The device initialization step is to select the “Dough Port” button (circled) prior to entering any data in the Dough Port machine component (far left)..... 11
- Figure 3. Task structure for the doughnut-making task as determined by the device design. As in Byrne and Bovair (1997), ovals represent goals and subgoals, rectangles represent actions required to fulfill subgoals, and subgoals are executed from left-to-right. A dashed border identifies the action for the device initialization subgoal. The red dotted line indicates a divergence between the path required by the device, and the execution path that users naturally attempted to follow..... 13
- Figure 4. A sample of the visual feedback displayed each time a new component was activated by pressing its selector button. 30
- Figure 5. System topology representation of the device model for the doughnut-making machine 33
- Figure 6. The modified Wicket Doughnut Making Machine interface..... 35
- Figure 7. Error rates for task steps in the main procedure. Error rates above .05 suggest a systematic error. 43
- Figure 8. Mean error rates for the device initialization step in each condition, with error bars identifying the corresponding standard deviations..... 45

List of tables

<u>Table 1.</u> List of steps for completing the preliminary task procedure	36
<u>Table 2.</u> List of steps for completing the main task procedure. The device initialization step is identified with bold text, and the postcompletion step with italic text.....	37
<u>Table 3.</u> Summary of the differences in the training material presented to the four groups.	40
Table 4. Mean error rates and standard deviations for the device initialization error, by condition.....	44
<u>Table 5.</u> Mean error rates for selector errors, postcompletion errors, and data-step errors, by condition (data not normally distributed).....	46
<u>Table 6.</u> Example segments of responses to question 1 that were included in the <i>explicitly mentioned</i> (EM) response category.....	47
<u>Table 7.</u> Example segments of responses to question 1 that were included in the <i>indirectly mentioned</i> (IM) response category.....	48
<u>Table 8.</u> Number of responses to question 1 that fell into the <i>explicitly mentioned</i> (EM), <i>implicitly mentioned</i> (IM), and <i>not mentioned</i> (NM) categories, per condition.	48
<u>Table 9.</u> The most frequent responses to question 3 in the post-study interview, which examined what participants emphasized during the experimental trials.....	49

1 Introduction

The term “human error” means many things to many people. For example, a student calculating an incorrect sum during a math test, a busy mom forgetting to confirm that she has her house keys before pulling the door closed behind her, and a pilot entering data into an airplane’s Flight Control Unit while it is in the wrong system mode might all be described as examples of human error.

While we apply the general term “human error” to each of these situations, researchers have identified a limited number of recurrent error forms, with the most common form being errors of *omission* (Reason, 2002). An omission error involves leaving out a necessary step in a task sequence, which generally prevents the task from being successfully completed. A critical feature of omission errors is that knowledge of the omitted step and when to execute it is intact; that is, they are not errors that arise due to a lack of knowledge. As such, omission errors generally occur during well-learned, automatic procedures.

This thesis investigates a particular type of omission error reported by Li (2006), which he calls an *incorrect interface usage* error. This error occurs during the execution of a procedural task, when the participant remembers the task sequence correctly but temporarily forgets how to operate the device interface correctly (Li, 2006, p. 238). In the particular task used by Li, the error occurs near the very beginning of the primary task sequence; before proceeding with the main task the user must select a specific button on the interface, and it is this selection step that is frequently omitted. Therefore, this error will be referred to as a *device initialization* error.

Forgetting a step at the beginning of a routine task is not unfamiliar to most of us; an example from everyday life is setting out to boil a pot of water but forgetting to turn the gas on to the stove before setting the temperature for the burner. Preliminary investigations by Li suggest that this error is remarkably stable, and may be related to the error-prone step having a secondary role in the task sequence (i.e., it may not represent a natural step in the task sequence).

As so many of our daily activities involve interactions with device interfaces, the possibility that certain types of steps at the beginning of a task are more prone to error is a very important issue to explore. While most day-to-day errors result in minor annoyances,

those that occur in safety critical situations can be catastrophic, and in such situations error prevention is crucial. For example, in an analysis of incidents in nuclear power plants Rasmussen (1980) determined that approximately 34% of errors arose from the omission of functionally isolated steps (i.e., steps that are not cued by the external environment or internal processes of the user). Through a better understanding of the conditions under which the device initialization errors reported by Li are likely to occur, as well as the precise characteristics of the device step that make it error prone, and factors associated with both the task and the individual that increase or decrease likelihood of the error occurring, preventative measures may be taken. Those measures may be in the form of the interface design itself, the design of the task structure, or even the training material provided to users of the system.

This thesis contributes to the body of knowledge about device initialization errors that occur at the beginning of task sequences by exploring factors internal to the individual that might influence the likelihood of committing such errors. Li suggested that a contributing factor to the omission rate on initial procedural steps might be the relevance of that step to the main task goal. That is, if the initial step doesn't directly move one closer to their main goal, but is a necessary step imposed by the device design, then it may be considered of low relevance to the main goal. Further, if that step occurs at the beginning of the task sequence then it may be more prone to omission errors than neighbouring steps, and than similar steps later in the procedure. Gray (2000) and Blandford et al (2006) refer to such actions that don't result in progress towards a goal state as *device-specific* actions, and refer to the main goal as a *task-based* goal; the same terminology will be used in this thesis.

Relevance to the main task-based goal has also been implicated as a factor in the omission of other procedural steps that occur at the *end* of the task sequence (e.g., Byrne & Bovair, 1997). However, there is no existing study that directly investigates the role that relevance plays in omission errors. As such, the first factor examined in this thesis relates to the effect of relevance to the task-based goal on the omission rate for an error-prone procedural step at the beginning of a task sequence.

The second factor explored in this thesis relates to participants' understanding of the role that error prone device-specific steps play in a procedure. Kieras & Bovair (1984) demonstrated that when participants are provided with a concrete "device model", or description of how a device works in terms of its internal mechanisms, they are able to learn and retain the operating procedures more effectively and execute correct procedures

more often than participants who are not provided with a device model. It is possible that when participants in Li's experiment learned the procedure they were required to execute, the role of the initial selection step in relation to the overall procedure was unclear. This lack of understanding of the step's purpose may have resulted in it being attributed lower importance or salience relative to other steps in the procedure. As such, this thesis also explores whether providing participants with a viable device model that explains the role of the initial device-specific step in the overall operation of the device is sufficient to increase its internal salience and result in a reduction in its omission rate.

An overarching goal of the reported work is to begin to identify whether systematic device initialization errors can be mediated without making physical changes to the device, through the emphasis of certain information prior to learning the device's operating procedures. If the rate of errors can be successfully reduced, this might help guide the development of training programs to employ in situations where similar errors have been identified. Alternatively, if the rate of errors cannot be reduced, this will suggest that investment in redesigning the device and task flow should be the preferred path as soon as such errors are identified.

2 Background on the device initialization error

Knowledge of the task environment used by Li (2006) provides useful context for understanding and discussing the device initialization error as observed in his studies and reproduced for this thesis. This section presents relevant background information and establishes further vocabulary related to the device interface and procedure that will be used throughout this thesis.

2.1 Overview of the task environment and task procedure

Li's (2006) experiments involved the use of two different interfaces that together simulated a doughnut-making operation. The primary task-based goal that participants worked towards was to produce the correct amount of doughnuts ordered by customers. The first interface, called the Wicket Doughnut Call Center, simulated a call center in which participants had to respond to incoming calls from different customer locations in London to retrieve doughnut orders. The interface is shown in Figure 1. The *customer location selector* (see item A in Figure 1) was used first to specify the location of an

incoming call, the *customer location tube map* (see item B in Figure 1) identified all customer locations in London with a doughnut symbol on the map, and the *customer order processor* (see item C in Figure 1) was used to send the corresponding doughnut order details to the second interface, the Wicket Doughnut Making Machine.

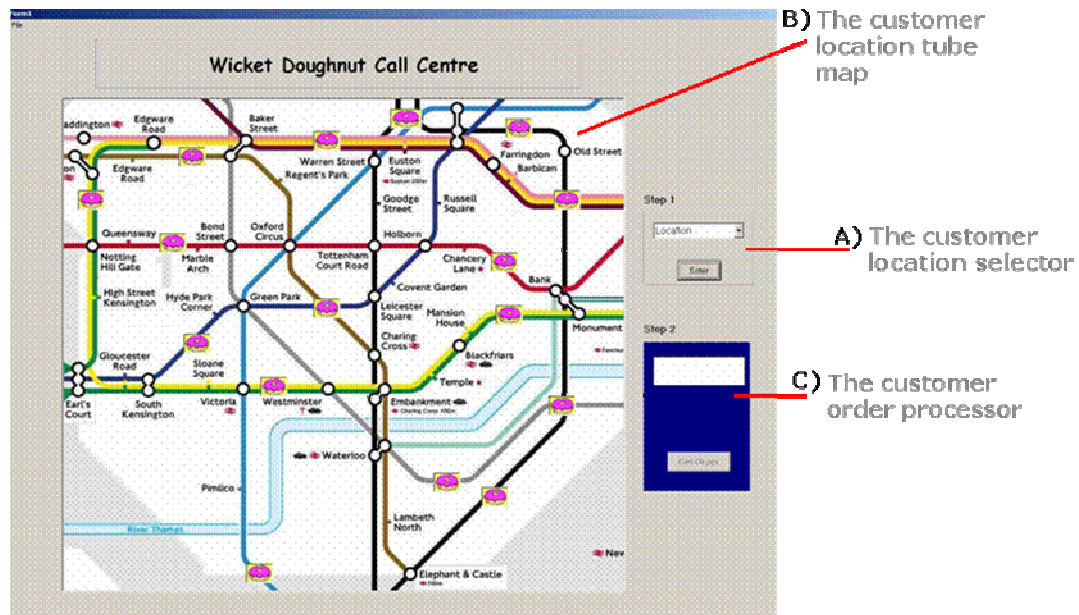


Figure 1. The Wicket Doughnut Call Centre interface used by Li (2006) to simulate a call centre for a doughnut-making operation. Participants used the interface to retrieve incoming orders from different customer locations.

The Wicket Doughnut Making Machine, shown in Figure 2, was used to simulate production of the doughnuts ordered through the call center. The Order Sheet (item A in Figure 2) displayed the order details after the “Next Order” button had been pressed. Participants were required to enter data from the Order Sheet into the five machine components around the outside of the interface, to produce doughnuts that matched the order. The components had to be operated in the following sequence: Dough Port, Puncher, Froster, Sprinkler, Fryer. Prior to entering data in any component, the corresponding *selector* button (see item B in Figure 2) had to be pressed in order to “activate” that component. For example, before entering the desired quantity of dough in the Dough Port component, the Dough Port selector button (circled in Figure 2) had to be pressed. After entering data into all five components, participants had to press the “Process / Clean” button (item C in Figure 2) once to process the order, and a second time to clean the machine.

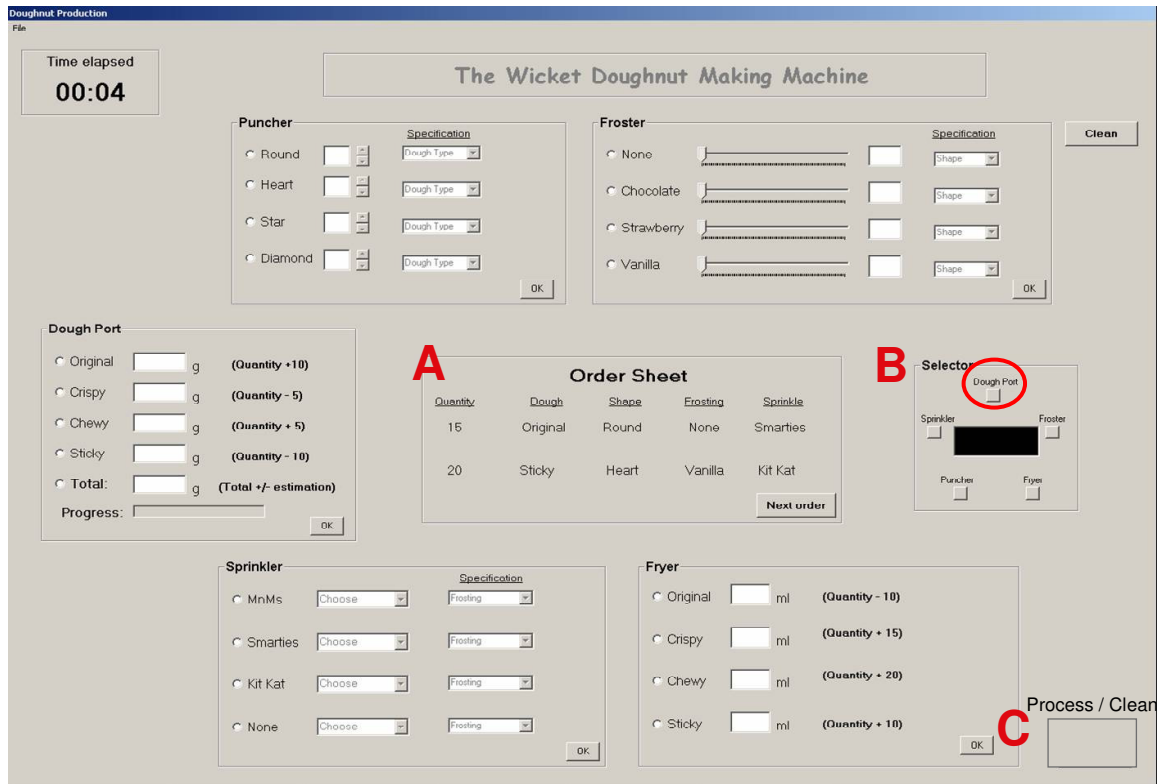


Figure 2. The Wicket Doughnut Making Machine used by Li (2006) to simulate the production of doughnuts (the red letters have been added by this author for the purpose of identifying specific parts of the machine). The device initialization step is to select the “Dough Port” button (circled) prior to entering any data in the Dough Port machine component (far left).

An additional aspect of the task was the need to slightly transform the data presented in the Order Sheet before entering it into the corresponding components. For example, the Order Sheet displayed the number of doughnuts required, but the quantity entered into the Dough Port component was for the amount of dough (in grams). As such, participants would apply simple mathematical rules specified in the Dough Port, such as to add 5 to the number of doughnuts required in order to get the amount of dough needed.

2.2 The device initialization error: omission of the Dough Port selector step

Li (2006) investigated factors that influence omissions of the last step in the task sequence (the second press of the “Process / Clean” button, item C in Figure 2). Omissions of this step fall into a category of errors called postcompletion errors (Byrne & Bovair, 1997), and will be discussed in more detail in subsequent sections. However, Li also reported a large number of omissions on *selector* steps (i.e., steps that involved clicking one

of the selector buttons in item B in Figure 2), which he termed skip-selector errors. Li described skip-selector errors as the “correct task sequence execution but incorrect usage of the device interface.” They occurred when participants attempted to enter data into a component without first pressing the corresponding selector button to activate it. Approximately 67% of all skip-selector errors in Li’s experiment occurred for the Dough Port, the first component operated in the task sequence; pressing the Dough Port’s selector button was the first step in the main doughnut-making task, and was also the most commonly omitted step in the procedure. Omission of this step (circled in Figure 2) is the error referred to as the *device initialization error* in this thesis.

2.3 Understanding the device initialization error in terms of the device’s and user’s task structures

When attempting to identify the source of an omission error, Byrne and Bovair (1997) noted the importance of identifying which aspects of the task structure are determined by the device and which are determined by a user’s goals. Figure 3 depicts a possible representation of the task structure for Li’s doughnut-making task as determined by the device design. The task structure for operating the Dough Port component is highlighted in blue, and the action for the device initialization subgoal is drawn with a dashed border. The procedural path required by the device involves an *Operate Dough Port* subgoal, which in turn has two further subgoals: *Activate Dough Port* (the device initialization subgoal), and *Enter Dough Port data*. The high rate of omission of the *Click Dough Port selector* action, which is the action for the device initialization subgoal, suggests that the path followed by users in order to fulfill their main task-goal often diverged from the path required by the device; this divergence is represented by the red dotted line in Figure 3. To satisfy the task-goal of making doughnuts, participants attempted to omit the *Activate Dough Port* subgoal and directly fulfill the *Enter Dough Port data* subgoal, indicating that the path imposed by the device did not match the natural execution path for the user.

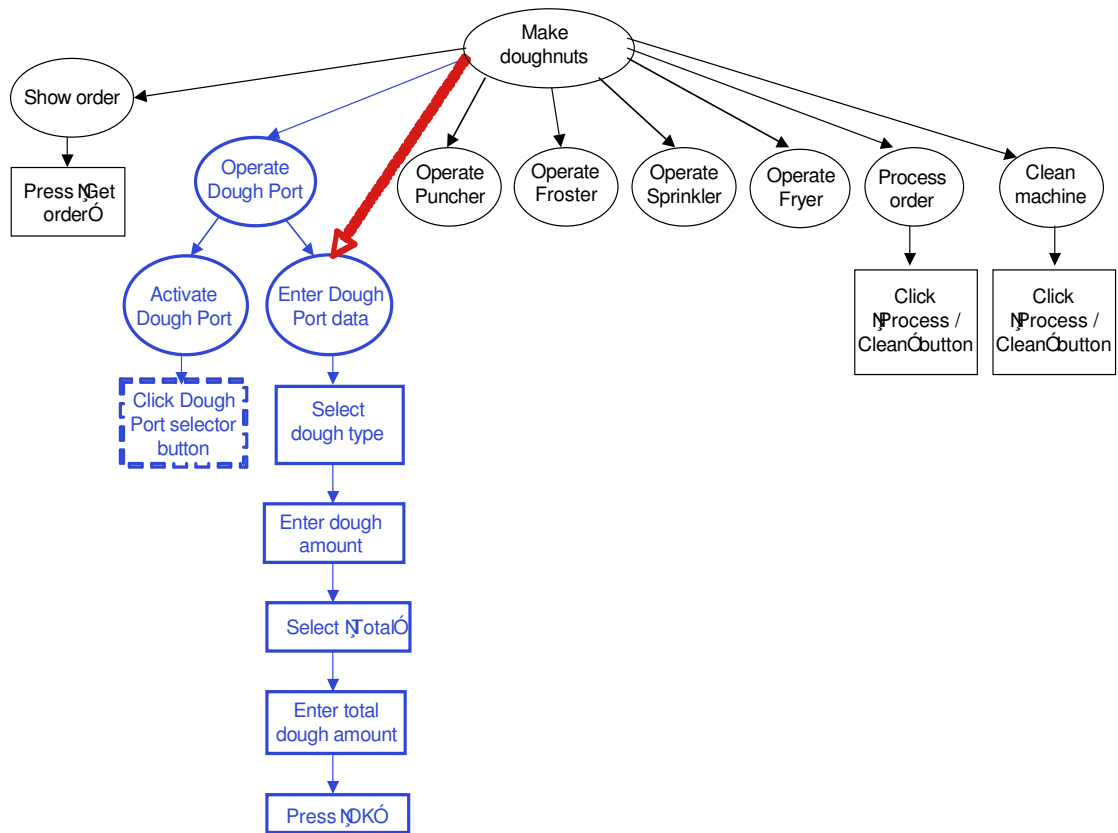


Figure 3. Task structure for the doughnut-making task as determined by the device design. As in Byrne and Bovair (1997), ovals represent goals and subgoals, rectangles represent actions required to fulfill subgoals, and subgoals are executed from left-to-right. A dashed border identifies the action for the device initialization subgoal. The red dotted line indicates a divergence between the path required by the device, and the execution path that users naturally attempted to follow.

The work reported in this thesis provides a starting point for understanding the underlying cause for this divergence, and strategies that can be employed to encourage a user's execution path to match the path imposed by the device.

3 Motivation for the study of errors

Each error has associated costs that can be used to evaluate its severity, and the degree to which we pay attention to our errors seems to depend on how significant the cost is: the greater the perceived cost, the more effort we put into avoiding similar errors in the future. Of course, an error's severity will be judged differently by people in different roles. For example, the student described in the introduction section who calculated an incorrect sum on her math test may not be concerned about losing a few points, but her teacher might

interpret the error as an indication that his teaching was ineffective; the busy mom who locked herself out of the house may miss her child's rugby game, while the locksmith gladly collects his fee for opening the door; and the pilot who entered flight data incorrectly might lose her life as well as those of her passengers, while the airline is held accountable by the public. These examples illustrate that in some cases errors are an important part of human life that allow us to adapt our behaviour, while in others they have tragic consequences that we want to avoid at all costs. As Reason (2002) suggests, errors are not intrinsically bad, nor can they be completely eliminated, but through careful study they can be better understood.

Several accidents with high costs to the public have been attributed, at least in part, to human error (such as Three Mile Island nuclear accident in 1979, and the Challenger space shuttle accident in 1986) and have resulted in a greater drive to understand errors and their causes so they can be prevented in future situations (Reason, 1990). Reason suggested that through a better understanding of the human mental processes responsible for both error-free and error-ful behaviour, more effective methods for predicting and reducing dangerous errors should emerge.

In our personal lives we employ a number of different strategies to assist us in reducing the amount of errors we make, such as using checklists, developing routines for "double-checking", and creating visual reminders (Reason, 2002). However, these strategies don't help us to understand the *cause* of our errors, nor do they help us to predict when an error is likely to occur. Studies that contribute to our understanding of distinct factors that contribute to error commission, and in what combinations, should provide opportunities to improve the accuracy of our error predictions (Reason, 1990, p.4), and therefore should also improve our ability to manage errors in safety-critical situations.

Studies of error may also support the design of work and task flows, as well as training programs. The same mental processes that govern error-free behaviour are also at play when errors are made, and careful study of the errors people make in procedural tasks can help us understand more generally how people learn and execute procedures correctly (Byrne & Bovair, 1997, p.58). As such, task flows and training material can be developed to capitalize on this type of knowledge, and the design of devices themselves can be optimized according to people's natural learning and behavioural patterns.

4 Common approaches to the study of errors

4.1 *Naturalistic studies*

One approach to studying errors and their causes is to conduct naturalistic observations in the context of situations where errors have occurred or might occur in the future. For example, in a retrospective analysis of the Three Mile Island nuclear accident, Le Bot (2004) discusses the complexity of human behaviour and the rich interactions between human operators and their surrounding technical environments, arguing for a holistic study of system operation and associated errors. Le Bot describes how Distributed Cognition, an ethnographic approach developed by Hutchins (1995) for studying socially distributed cognitive activities, can be used as a framework for understanding and explaining accidents in context rather than as an instance of individual operator failure. While such approaches have strong explanatory power, they have limited ability to contribute to predictions of future errors.

Another approach to the study of errors is to collect, analyze, and categorize errors that occur naturally. This is one of the oldest methods employed, and has been useful for revealing the variety of different error types that occur, and for identifying common patterns of errors (Reason, 1990, p. 13-14). Collections of common errors have also supported the development of error classifications or taxonomies; although it sometimes feels like errors abound, correct actions are much more common than errors, and the errors that do occur take on a limited number of forms (Reason, 1990). Two common classification schemes come from Norman (1981, 1988) and Reason (1990), as described below.

4.1.1 Norman's classification scheme

Norman (1981, 1983a, 1988) categorizes errors in terms of *mistakes* (correctly executing the wrong action sequence) and *slips* (incorrectly executing the right action sequence). He suggests that mistakes occur at the *intention* level (e.g., the intention or goal that is formed is not appropriate for the situation), and therefore can be difficult for us to predict and detect. On the other hand, slips occur at the *action* level, during the execution of tasks that have become automatic and don't require a lot of attention. Slips tend to be easier to detect because they impede achievement of the primary goal.

Norman discusses the emergence of slips in terms of an *activation-trigger-schema* (ATS) system, a theory which proposes that action sequences are controlled by the activation, selection, and triggering of hierarchically organized memory units called schemas. Schemas can be activated by cues in the external world or internal to the individual, and activation of a schema represents the formation of an intention. Each schema has a set of triggering conditions, and will be selected based on a combination of its activation level and the goodness-of-match of its triggering conditions (Norman, 1983a). Slips occur due to errors in the formation of an intention, activation of a schema, or triggering of a schema.

According to Norman's system, the device initialization error identified by Li (2006) might be categorized as a slip due to insufficient activation of the schema associated with the initial step. For example, it is possible that either something inherent in the task or something internal to the user results in the initial step losing activation compared to the schema for the subsequent step. Alternatively, the error might be categorized as a slip due to triggering; the goodness-of-match for the subsequent step's triggering conditions may be higher, resulting in it being selected over the initial step.

Norman (1983a) argues that certain classes of errors and their effects can be minimized through good system design, but also that designing to minimize one class of errors can actually increase the likelihood of other classes of errors. As such, it is very important to be aware of the tradeoffs between different design strategies.

4.1.2 Reason's classification scheme

Reason (1990) argues that slips and mistakes, as proposed by Norman, don't provide a complete account of all different error types because some errors appear to possess characteristics of both, making it difficult to classify them as one or the other. He proposes an alternative classification based on the three levels of cognitive control described by Rasmussen (1982): *skill-based* performance, in which stored patterns of instructions govern the execution of actions; *rule-based* performance, in which stored if-then rules govern the solutions applied to familiar problems; and *knowledge-based* performance, in which conscious processing and stored knowledge are used to plan action in response to novel situations. Reason's corresponding error types are as follows:

1. **Skill-based slips and lapses** occur during routine activities that are executed automatically and without conscious control, due to failures in the execution or storage of an action sequence.
2. **Rule-based mistakes** occur during problem solving activities, when an existing plan or rule is applied inappropriately (e.g., a rule that worked in a previous, similar situation is applied, but doesn't fit the current situation).
3. **Knowledge-based mistakes** result from lack of expertise or incomplete knowledge related to a particular problem.

The device initialization error is most likely a skill-based slip, because the error occurs during a routine and nonproblematic activity which does not involve rule or knowledge-based performance; participants in Li's (2006) study were well practiced at the task, and knowledge of the omitted step appeared to be intact. The error also preceded detection of a problem, which is another defining characteristic of errors at the skill-based level (Reason, 1990, p. 56).

Distraction or preoccupation that results in attentional 'capture' is also a necessary condition for an action slip. Further analysis of the task provided by Li reveals that participants may have been triggered to process information required for the second step in the doughnut-making procedure at the precise moment they were expected to execute the device initialization step, therefore potentially providing the essential distraction or preoccupation. As described in section 2 of this thesis, the first step in the doughnut-making task used by Li (after the call-centre task was completed) was to select the Dough Port selector button (this was the device initialization step), and the second step was to enter data from the Order Sheet into the Dough Port component. Interestingly, the data that was to be entered into the various components was revealed in the Order Sheet immediately before participants were to execute the device initialization step. Because the data had to be transformed before being entered, requiring participants to engage in conscious and deliberate problem-solving, it is possible that its display triggered an immediate transition from skill-based to rule-based performance, thereby capturing their attention and exposing them to the risk of an action slip on the first selection step.

4.1.3 Summary of naturalistic studies of error

Naturalistic studies have enabled the identification and description of naturally occurring errors, and have provided the basis for error classification schemes. Retrospective investigations of error incidents have provided valuable insights into the complex environments that errors thrive in, and have contributed to our understanding of how different environmental factors might influence error occurrence. However, these studies have limited predictive power and make only small contributions to our understanding of the internal, cognitive causes of error, because they don't provide visibility to the underlying mechanisms involved (Gray, 2004).

Error classification schemes have been useful for providing a common language with which to discuss errors and for enabling the post-hoc analysis of errors. However, errors that are categorized together because they exhibit similar characteristics may in fact stem from completely different causal mechanisms (Reason, 1990), which greatly limits the utility of such categorizations. In addition, they have also been heavily criticized for their lack of specificity and explanatory power (e.g., Byrne & Bovair, 1997; Chung & Byrne, 2004; Gray, 2004).

4.2 Empirical studies of error

Fifteen years after Reason (1990) emphasized the need to develop a thorough understanding of the human mental processes responsible for both error-free and errorful behaviour, Blandford, Back, Curzon, Li, and Rukeenas (2006) argued that while the surface manifestations of many error types have been well described in the literature, their underlying cognitive causes are still poorly understood. Gray (2004) argues that years of naturalistic studies of error have not been fruitful, and that rigorous study of the nature, detection, and correction of errors should be pursued in laboratory settings. Laboratory studies enable the systematic manipulation of factors that are hypothesized to mitigate or provoke errors, allowing researchers to explore specific causal explanations in a controlled setting (Reason, 1990, p.14).

There are two primary approaches to the study of errors in lab settings. The first, exemplified by Gray (2000), involves the collection of large amounts of both errorful and error free data for the same task, which is then subject to fine-grained analyses. The goal of this approach is to develop an understanding of correct, error-free performance in addition to an understanding of the nature of the errors that occur.

The second approach, exemplified by Byrne & Bovair (1997), is to develop an experimental task paradigm that induces an error rate high enough to be studied and analyzed in detail. Different factors that might contribute to or mitigate the error rate can then be explored in the controlled environment of the lab.

Both of these approaches have to overcome the general problem that eliciting systematic procedural errors in a lab environment is very difficult to do (Byrne & Bovair, 1997, p. 41-42). In addition, as with any laboratory study, the generalizability of results is somewhat limited due to the unnatural and artificial tasks used, and the unnatural behaviour of participants in lab settings (Byrne & Bovair, p.42).

Despite these challenges, the approach taken in this thesis is a lab-based study using an artificial task paradigm similar to that used by Li (2006), as described in section 2. Li's task does elicit the error of interest at a sufficient level for detailed study, and at this early stage in the investigation of the device initialization error at the beginning of tasks, it is felt that more information can be gained about its nature from a controlled study than through naturalistic observations or reports.

5 Existing studies of omission errors

Leaving out necessary steps in a task sequence is the most common type of human error (Reason, 2002), yet despite extensive study on certain types of systematic omission errors they are still poorly understood (Blandford et al, 2006). This section presents previous research that has been conducted on omission errors, and discusses how that work might relate to the device initialization error.

5.1 Postcompletion errors

The most commonly studied omission error was initially described by Byrne and Bovair (1997), who observed that people are more likely to omit steps in a task that occur after the main task-based goal has been accomplished; they termed such omissions "postcompletion" errors (PCEs). A classic example used to illustrate PCEs is forgetting to retrieve the last page of the original document from a photocopier after making copies; the task-based goal is to get copies, and the procedure is structured such that a device-specific clean-up step remains after that goal has been accomplished: to retrieve the original. It is this final step that is most often omitted, even though knowledge of the step is intact (i.e.,

most people don't make the error every time, or even most of the time, they make photocopies).

5.1.1 The role of task structure in postcompletion errors

Reason (2002) suggested that task structure plays an important role in the omission of certain steps, and identified four “omission affording characteristics” that are common to steps prone to postcompletion errors (such as removing the last page of the original from the photocopier):

1. The task-based goal of the activity is achieved before the entire procedure finishes execution, which introduces a “false completion signal” (some sort of feedback that falsely indicates the procedure has been completed).
2. The postcompletion step is positioned at the end of the procedure, subjecting it to interference from preoccupation with the subsequent task.
3. The postcompletion step is functionally isolated, because there are no cues to prime it.
4. There is no visible reminder of the need to perform the step.

Li (2006) also emphasized the relevance of certain task characteristics that contribute to the likelihood of a PCE occurring. In particular, he suggested that presence of a false completion signal competes with the PC step, cueing people to move on to their next task and omit the PC step (which suggests that having a follow-on task to move on to is also an important factor).

A preliminary analysis of Li's task according to the characteristics identified by Reason (2002) reveals some important similarities between the postcompletion step and the device initialization step:

1. In both cases, the omitted step lays outside the boundaries of the main task-based goal. The task-based goal is achieved **before** the postcompletion step is to be executed, while progress towards the task-based goal does not begin until **after** the device initialization step is to be executed.
2. In both cases, the position of the omitted step within the task procedure may result in interference from an upcoming task. The postcompletion step is positioned at the end of the task sequence, so attention may be consumed by preoccupation with the

subsequent task. The device initialization step is positioned at the beginning of the task sequence, so attention may be consumed by preoccupation with the current task.

3. In both cases the step is functionally isolated, with no cues to prime it.
4. There is no visual reminder for either step.

5.1.2 Studies of postcompletion errors

Studies of PCEs are relevant to the study of the device initialization error because they appear to share some common traits, as noted previously. These studies might provide useful insight into the factors that contribute to occurrence of the device initialization error.

Byrne and Bovair (1997) conducted some of the earliest empirical studies of PCEs, and were the first to reliably replicate them in a laboratory setting. They proposed a theory that PCEs result from goal forgetting in working memory; since knowledge of the correct task sequence is known to be available in long-term memory (LTM), their theory assumes that goal forgetting occurs in working memory. They explain this behaviour in terms of goal activation. Parent goals (i.e., task-based goals) in working memory have associative links to the subgoals and actions that are necessary in order to achieve them, and activation is provided to subgoals through these associative links. When a parent goal is satisfied, it is eliminated from working memory and therefore no longer provides activation to any of its remaining subgoals. Due to the task structure, the corresponding subgoal for a PC step is not satisfied by the time its parent goal is satisfied and eliminated from working memory, and therefore it is no longer supplied activation from its parent. Therefore, in cases where working memory load is high due to other active goals in memory, Byrne & Bovair predicted that the PC subgoal would be more likely to go unsatisfied and result in a PC error, because high working memory load is associated with faster decay of information from working memory. Byrne & Bovair instantiated this theory as a cognitive model, which was evaluated alongside the empirical tests described below. The model predicted that high working memory load would be associated with a higher rate of PCEs.

To evaluate the effect of working memory load on PCEs, Byrne and Bovair created an artificial task environment based on Star Trek, designed with a final step that either occurred before the task-based goal of the procedure had been accomplished (in the control condition) or after the task-based goal had been accomplished (in the PC condition). They

also introduced a working memory load condition, in which participants had to concurrently recall auditorily presented stimuli at random intervals.

The results from this study confirmed their model's prediction, showing that when working memory load was low, participants rarely made PCEs, but when working memory load was increased the frequency of PCEs also increased. The fact that other procedural errors were found not to be affected by the change in working memory load strongly suggests that PCEs are unique from other types of procedural errors, and might therefore result from different underlying mechanisms; this possibility has since led to a number of further investigations into the role of goal activation in PCEs, as well as other factors that might provoke or mitigate the error.

5.1.3 A memory theory approach to goal activation: The AGM model

Altmann and Trafton, (2002) proposed an influential memory theory approach to cognitive goal representation and management in an effort to address what they felt were faulty assumptions about the way goal memory operates. Traditional accounts assumed that goals were stored in a special memory that functions like a first-in-last-out stack structure. However, Altmann and Trafton provided a model that is based exclusively on general memory constructs, arguing that a special goal memory is not necessary based on our current understanding of cognitive constructs.

Altmann and Trafton's model is similar to that proposed by Byrne and Bovair, and they also use it to provide an account of PCEs (which are assumed to be mediated by goal structures). The basic assumption of their model, called the activation-based goal memory model (AGM), is that in order for a subgoal to direct behaviour it must be the most active goal in memory. The model identifies three constraints on goal-directed behaviour:

1. **The interference level.** This represents the collective effect of distractor goals. A target goal must be above the interference level in order to be retrieved over any distractors. However, even if the target goal is above the interference threshold, there is no guarantee that it will be sampled over distractors that are also above the threshold.
2. **The strengthening constraint.** Activation of a goal must be increased through strengthening in order to overcome interference from other goals. However, the

more active a goal becomes the more interference it will cause for subsequent goals, so the system must strike a balance.

3. **The priming constraint.** If a goal is suspended, it can only be resumed after it has been primed from an associated cue. This cue can come from the outside world, or it can be part of the individual's internal mental context (e.g., it might come from their long-term knowledge about the task).

This model makes the important assumption that a goal's activation decays gradually but continuously. Therefore, unlike the model proposed by Byrne and Bovair, goal forgetting occurs in two ways: through interference from other elements in memory, and through the decay process. This highlights the importance of priming cues to increase activation of a decaying target goal at the appropriate time.

In terms of this model, we can begin to make some observations about the device initialization error. First, the goal associated with the device initialization step is often *not* the most active goal in memory at the appropriate time, otherwise it would successfully direct behaviour and lead to the corresponding step being correctly executed. This suggests that the device initialization goal is either consistently below the interference threshold for some reason, or that the activation level for the subsequent step is consistently higher for some reason. In either case, the salience or importance assigned to that initial goal appears to be much lower than that of its neighbour, resulting in lower activation. An important avenue of investigation, therefore, is whether or not the salience of that goal can be increased, in order to reduce the rate of omission.

5.1.3.1 Using visual cues to increase goal salience

Related studies have been conducted with regard to the PCE, in which visual cues were introduced into the task environment to determine their impact on the salience of PC goals in procedural tasks. Chung and Byrne (2004) examined the effect of two different types of visual "interventions": the first intervention was a visual cue in the form of blinking red and yellow arrows, which was presented "just-in-time" for the PC step (i.e., it appeared immediately before the PC step should be executed); the second intervention was a visual mode indicator that displayed a change in the system state using highlighting and contextual information, which was presented prior to the PC step.

Using the same task environment as Byrne and Bovair (1997), Chung & Byrne found that just-in-time display of the visual cue resulted in error free performance on the PC step across all trials, whereas presentation of the mode indicator did not have a significant effect on the PC error rate. They concluded that the visual cue acted as a primer to the PC step, contributing to its level of activation and allowing its goal to be satisfied, while the mode indicator did not sufficiently prime the PC step.

This work demonstrated that it is possible to increase the saliency of a frequently omitted PC step using visual cues, and also suggested that specific properties of the cue contribute to it being more or less effective. The fact that the visual mode indicator did not have a significant effect on the PC error rate extended previous work by Chung (as cited in Chung & Bovair, 2004), in which the onset of a red dot next to the PC step also did not result in a reduction in the PC error rate. As such, Chung and Byrne suggested that simply adding visual cues is not enough for consistent error reduction. They noted that appearing “just-in-time”, containing movement, and having a meaningful shape appear to be important characteristics for a visual cue to reduce PCEs, but also that properties of the task and the interface itself can impact the effectiveness of a visual cue.

Li, Blandford, Cairns, and Young (2005) also observed a positive effect from visual cues in a study of PCEs in non-procedural tasks. They used slightly adapted versions of the Missionaries and Cannibals logic problem (see Ernst & Newell, 1969), in which participants had to move various items back and forth across a river to accomplish a specific task-based goal. The PC step was to move the transportation vessel back across the river at the end of the task. Participants were asked to solve the problems using one of two interfaces: Text or Pop-up. The Text interface required that participants type into a text box the name of each item they wanted to move across the river, and did not contain any visual information that might cue the need to execute the final PC step. The Pop-up interface allowed participants to choose which item they wanted to move by selecting it from a pop-up menu. In this case the menu contained an entry for the transportation vessel, which may have served as a subtle visual cue to the PC step.

Li et al found that participants committed significantly fewer PCEs in the Pop-up condition than in the Text condition, suggesting that presence of the PC item in the menu served as a reminder or cue to the execution of the PC step. Interestingly, the difference in error rate was significant in tasks where there were only three items in the pop-up menu, but not in tasks where there were five items in the menu. Li et al suggested that the five-

item list provided more distractors from the PC item, making it less prominent as a visual reminder. This finding lends further support to Chung and Byrne's (2004) suggestion that the effectiveness of a visual cue is very sensitive to specific properties of the cue as well as the task and interface environment. In Li et al.'s case, the cue did not appear just-in-time, nor did it have movement or meaningful shape, and the interface provided additional distractors that may have captured participant's visual attention.

5.1.3.2 Studies on procedural cueing

The influence of a second type of cueing, cueing from previous steps in a procedural task, has also been examined with regard to PCEs. Altman and Trafton's (2002) AGM model makes the assumption that each step in a well-learned procedure can act as an associative cue to the subsequent step. This has been used to provide a possible explanation for why PCEs don't occur every time a task is executed, as the step that precedes a PC goal may become an internal associative cue that primes activation to the PC step (Altmann & Trafton, 2002). The notion of procedural cueing suggests that disruptions that occur immediately prior to a PC step should interfere with the priming process, resulting in a negative effect on the PCE rate. In addition, the AGM model's notion of gradual decay suggests that a longer disruption prior to a PC step might be more disruptive than a short one, as it would allow for greater decay of the PC goal in working memory before the task was resumed, and therefore the likelihood of a subsequent omission would be increased.

Based on these predictions, Li, Cox, Blandford, Cairns, Young, and Abeles (2006) investigated the effect of interruptions on PCEs, looking specifically at how the position and duration of an interruption impacts the number of PCEs committed; they used the same doughnut-making task reported by Li (2006) and described in section 2 of this thesis. In their investigation of interruption position, participants were presented with a mental arithmetic task that lasted for 75 seconds in one of three positions: just before the PC step, in some other position, or not at all (i.e., no interruption presented). As predicted by the AGM model, they found that participants made significantly more PCEs when the interrupting task occurred immediately before the PC step than in any other position. In their investigation of interruption duration, three different interruption lengths were used: 75 seconds, 45 seconds, and 15 seconds. Based on predictions from the AGM model, Li et al. (2006) hypothesized that the position effect observed in the previous study should

persist for the longer interruptions, but that the position effect should disappear for the shorter interruptions as there would not be enough time for substantial decay of the PC goal to occur. However, the results did not reveal a significant effect of interruption duration, demonstrating that even very short interruptions can have a negative effect on the PCE rate.

In terms of the AGM model, these results support the notion that associative links between procedural steps can act as internal cues to the next step, and that due to the decay process these internal cues are highly sensitive to disruptions. Further, it suggests that the reason the PCE only occurs occasionally might be because the preceding step acts as an internal cue, reminding participants of the need to execute the final step. This introduces the possibility that the device initialization error might occur so frequently because as the first step in the procedure there are no preceding steps to act as internal cues and contribute to the priming process.

5.1.4 Observed differences between postcompletion and device initialization errors

Based on some of the similarities observed between the device initialization error and the PC error, it may be tempting to classify them as different manifestations of the same underlying cause. However, Li (2006) also reported some fundamental differences between the two error types. Although he was specifically investigating the effect of interruption position and duration on PCEs, the device initialization error was also observed within the same task. Li suggested that because PCEs were more likely to occur immediately following a disruption, knowledge of PC steps might be dynamic in nature. In contrast, given the device initialization step's position at the beginning of the main task sequence, interruptions were not presented before its execution and therefore errors on the device initialization step occurred independently of interruptions. In addition, errors on similar device-specific steps in the task (i.e., the selector steps for the other 4 components, as described in section 2) did not appear to be affected by interruptions. Li therefore suggested that while knowledge of PC steps appears to be dynamic, knowledge of how to operate the interface correctly is stable during task execution. While more investigation is certainly necessary, the results from Li's preliminary investigations suggest that different cognitive mechanisms might be at play in the two errors, despite the earlier noted similarities.

6 This experiment

Altmann and Trafton's (2002) AGM model predicts that presence of a priming cue which is associatively linked to a target goal will result in increased activation of that goal, and therefore should increase its likelihood of being correctly sampled and directing behaviour. As discussed previously, various researchers have investigated the effectiveness of presenting visual priming cues as a means to mitigate the occurrence of the postcompletion error, which exhibits some similar characteristics to the device initialization error. However, as Blandford et al (2006) note, visually salient procedural cues do not necessarily equal cognitively salient cues. In addition, there may be real-world instances where modifying the task environment to include visual cues, in an attempt to reduce systematic omission errors, is not an immediately feasible or desirable solution.

Altmann and Trafton (2002) do state that cues for priming a goal need not be external, but rather can also come from an individual's internal mental context; a suggestion that is central to this thesis. The work reported by Li et al (2006) supported the notion that associative links from previous steps in a procedural task can act as internal cues, and the experiment reported here investigates two different approaches to strengthening internal cues for the systematically omitted device initialization step. Both approaches use specially designed training material in an attempt to enhance participants' internal mental context while they learn how to execute the procedural task: the first approach uses training material designed to emphasize a different task-based goal that is more closely linked to the device initialization step, and the second approach uses training material designed to enable participants to develop a more accurate mental model of the device and the role of the device initialization step in its operation.

6.1 Approach 1: Improving cognitive salience by modifying the task-based goal

The first hypothesis explored is that the device initialization step has low cognitive salience because of its limited relevance to the task-based goal. This hypothesis is based on the notion that the task-based goal provides activation to all of its subgoals through associative links, as suggested by Byrne and Bovair (1997). Because the device initialization step does not appear directly related to accomplishing the task-based goal, it is attributed less importance and receives less activation, making it more prone to omissions.

As such, by modifying the task-based goal such that its accomplishment is directly linked to the correct execution of the device initialization step, and training participants on this new goal, it is expected that the cognitive salience of the step will be increased and result in fewer device initialization errors during execution of the procedure.

6.1.1 Related studies on goals and performance

6.1.1.1 Motivation

Back, Cheng, Dann, Curzon, and Blandford (2006) investigated a similar notion with regard to PCEs, by looking at whether motivating participants to correctly execute a PC step might influence the systematicity of errors. By creating a task environment in which correct performance of the PC step was important to fulfillment of the task requirements, Back et al hypothesized that participants would be motivated to ensure the step was correctly executed, thereby reducing the error rate.

The task environment used by Back et al was a space invader video game, in which participants' task was to shoot alien ships and rescue astronauts as they fell from the sky, while avoiding alien fire. A PCE occurred if participants forgot to change their own ship from rescue mode back to shooting mode after rescuing an astronaut, and every time a PCE was made their score was reset to zero. The objective of the game was to achieve one of the highest scores, so motivation to avoid PCEs should have been high. However, Back et al found that the PCE remained systematic despite participants being motivated to avoid it. They also suggested that the complexity of achieving the 'task critical' step (the step that achieves the task, and immediately precedes the PC step - in their case, making an astronaut rescue) might play a role in the occurrence of PCEs; they found that PCEs were more likely to occur if making an astronaut rescue was more difficult due to game circumstances such as increased alien fire or distance to the base.

The study reported by Back et al indicates that motivating participants by introducing a penalty (e.g., loss of points in a game environment) does not result in an improvement in error performance. However, it is possible that participants did not see achieving a high score as their main objective, making it a less-than optimal motivator; the higher error rate observed when game play became more difficult suggests that participants may have focused more on moving forward through the game than on actively tracking their points, and as such the level of motivation may not have been as high as anticipated.

The study reported in this thesis explores a related strategy that involves training participants to achieve a completely different task-based goal for which the device initialization step is highly important to its accomplishment.

6.1.1.2 Goal specificity

Burns and Vollmeyer (2002) have demonstrated that working towards different goals can affect participants' behaviour, focus, and performance in problem solving tasks. They examined the effect of goal specificity on participants' operation of a computer based *water-tank* system, a linear system with three numeric inputs, three numeric outputs, and weighted links between them. Changes to an input value resulted in different changes to the corresponding outputs depending on the weights for the links between them. Burns and Vollmeyer provided one group of participants with a nonspecific goal (NSG), which was to explore the system, and provided a second group with a specific goal (SG), which was to change the input values in the system so that a specific set of output values was reached. They found that participants in the NSG group engaged in more hypothesis-testing activity, whereas SG participants focused on actions directed towards achieving the desired output settings. This difference in focus resulted in the NSG participants acquiring a better understanding of the system's structure and how to control the outputs, and better performance on subsequent tasks using the system.

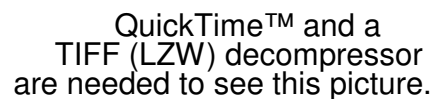
While the study reported in this thesis looks at structured, procedural interactions, the work by Burns and Vollmeyer does support the possibility that focusing on different goals can alter internal processes and influence the types of interactions users deem important.

6.2 The task-based goals used in this experiment

The task environment used in this experiment was similar to Li's (2006), and two related goal descriptions were used as the basis for training on the procedure: a low device relevance (LDR) goal description and a high device relevance (HDR) goal description. The LDR description (see Appendix 1) was intended to emphasize the outcome of the procedure (whether the doughnuts produced matched the customer's order specification) but not to emphasize particular interactions with the device. It positioned participants as bakers for the Wicket Doughnut Company, and instructed them that their task-based goal was to

produce batches of doughnuts that exactly matched orders from their customers (similar to the goal used by Li, 2006).

In contrast, the HDR description (see Appendix 2) was intended to emphasize attending to specific device interactions (including execution of the device initialization step) rather than the outcome of the procedure. Participants were positioned as Machine Testers for the company, and instructed that their task-based goal was to acknowledge and evaluate the machine's visual response each time a different selector button was pressed to activate a component in the doughnut-making machine. As shown in Figure 4, the *Selector* box in the doughnut-making machine provided textual feedback each time a new component was activated, and the role of Machine Testers was to report whether the device displayed the correct information.



QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Figure 4. A sample of the visual feedback displayed each time a new component was activated by pressing its selector button.

While all participants executed the same procedural steps in the same order, it was expected that participants who received the HDR goal description during training would make fewer device initialization errors during the experimental trials, because execution of the device initialization step (i.e., pressing the corresponding selector button and evaluating the machine's feedback) was necessary in order to correctly report on the accuracy of the device's overall feedback.

6.3 Approach 2: Improving cognitive salience through mental models

It has been suggested that people develop mental models of the systems and devices they interact with, and use those models to guide their interactions and predict how a system will behave and respond (e.g., Norman, 1983a; Norman, 1983b). Norman (1983b) described mental models in the context of three related concepts:

1. The *target system*: the system or device a person is using or learning to use.
2. The *conceptual model*: a model used to understand or teach the target system, and should ideally be the basis for the system design (it is an invention of designers, engineers, or teachers). It should provide a complete and accurate representation of the target system.
3. The *mental model*: The model that actually exists in a user's head. It is a functional representation of the target system, which includes knowledge about the internal structure and operation of the target system (which might not be accurate). It is used to understand and predict the target system's behaviour.
4. The *scientist's conceptualization*: a model of another person's mental model.

Norman (1983b, 1988) emphasized the need for a correspondence between a user's mental model and the conceptual model that a system is designed around; greater correspondence facilitates better learning of the system and leads to improved performance and problem solving (Norman, 1988). The notion that human performance can be negatively affected if a user's mental model does not correspond with the conceptual model that a system was designed from appears to also be significant to the study of human error; although performance is often measured in terms of speed and efficiency, surely errors are also an important factor to consider.

This leads to the second hypothesis examined in this thesis. Performance on tasks using Li's (2006) device could have been influenced by a mismatch between the device designer's conceptual model and the mental model formed by participants. In the conceptual model that the system design was based on, the 5 selector steps played a central role in the device's operation: prior to operating any part of the machine, the corresponding selector button had to be used to activate that machine part. In addition, each of the five selector steps played an equally significant role: the selector for the Dough Port machine component was of equal importance to the selector for each of the other machine components. However, the high error rate observed during operation of the device on the

selector steps in general, and the Dough Port selector step (i.e., the device initialization step) in particular, might suggest that the mental models developed by participants and used to guide their interactions did not correspond sufficiently to this conceptual model. The pattern of omissions reported by Li suggests that the selector steps did not play a comparatively important role in participant's mental models, especially the Dough Port selector step.

As such, it is hypothesized that by deriving a conceptual model from the system design that explains the internal structure and operation of the system relevant to the selector steps, and using this to train participants on how to operate the device, should support (though not guarantee) development of a mental model that incorporates more of the information important for correctly using the device initialization step during system operation. As such, performance (as measured by the device initialization error rate) should improve.

6.3.1 Related studies on mental models and performance

Kieras and Bovair (1984) conducted an empirical study that was designed to examine what role, if any, mental models play while learning to interact with simple devices. Their investigation was motivated by previously reported results that were ambiguous about whether having detailed knowledge of a system's internal mechanisms is important for successful operation of the system. For example, Halasz and Moran (1983) examined the effect of having a mental model on participants' ability to use a stack calculator. They found that having a mental model did not improve performance on some tasks, such as those involving routine problem solving, but did improve performance when solving novel problems.

Kieras and Bovair conducted related studies in which participants learned different procedures for operating a simple device. Some participants were given a *device model* that described how the device worked before they learned the operating procedures, and others simply learned the procedures "by rote". Those given the device model learned the procedures faster, retained the procedures better, and applied more efficient procedures more often during the testing phases of the experiment. They were also able to infer new procedures for operating the device using fewer actions. Based on these results, Kieras and Bovair suggested that having a device model improves the learning and retention of

operating procedures for a device, by supporting users in inferring what the correct operating procedures must be.

In a final study, Kieras and Bovair attempted to identify more precisely what information in the device model was most important for enabling participants to correctly infer the operating procedures, and therefore achieve better performance. The results suggested that for their device the critical information was the system topology combined with information about how power flows through the system. Kieras and Bovair then suggested that the ambiguity of previous studies might stem from their use of non-critical information in the models provided to participants.

6.4 The device model used in this experiment

While the overall role of mental models in performance remains unclear, they do appear to have an established effect on learning to operate simple devices as long the critical information is provided (Kieras and Bovair, 1984). As such, the device model used to train participants in this study, and shown in figure 5, is similar to that provided by Kieras and Bovair.

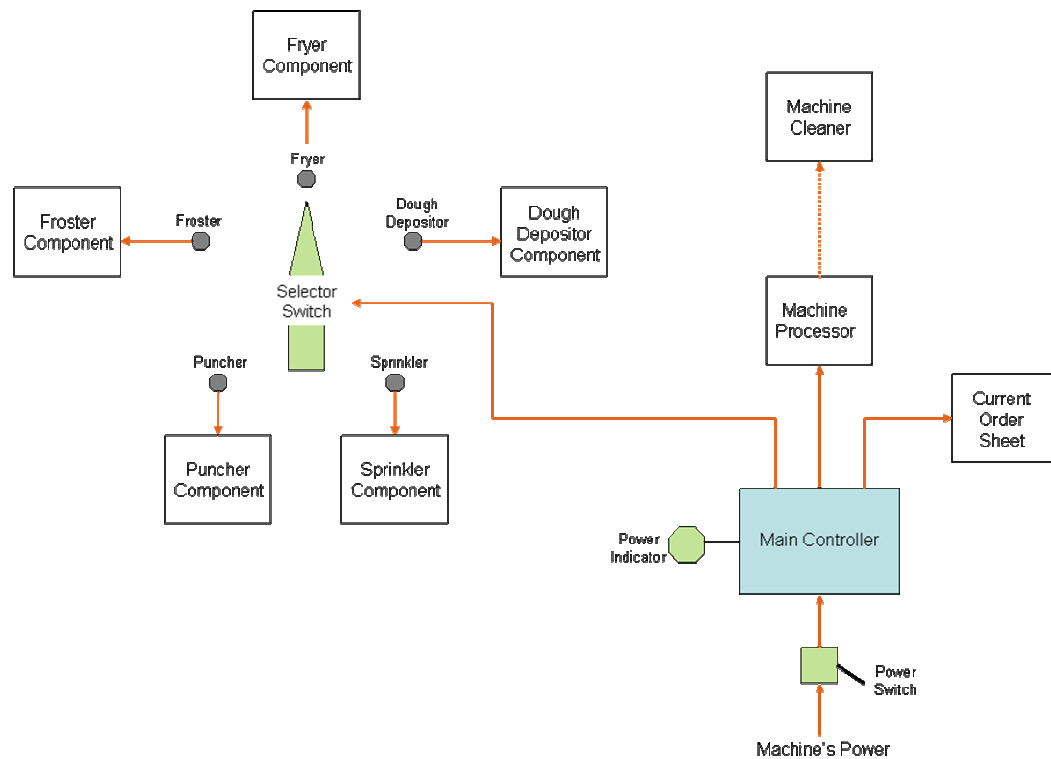


Figure 5. System topology representation of the device model for the doughnut-making machine

The full device model description is presented in Appendix 3; in summary, participants learned that the *Power Switch* is the on/off controller for the main power source, and when turned on causes the *Main Controller* to warm up and the *Power Indicator* light to turn green. Power always flows from the main controller to the *Current Order Sheet* display and to the *Machine Processor*, and the remaining power coming into the main controller is allocated to the different machine components (dough depositor, puncher, froster, sprinkler, and fryer) via the *Selector Switch*. The *Machine Processor* automatically controls power flow to the *Machine Cleaner*. The device model description did not contain any information about the doughnut-making procedure that participants were subsequently trained in.

Learning the device model was expected to support the development of a corresponding mental model that encapsulates the integral role of the device initialization step in the correct operation of the device, thereby acting as an internal cue to facilitate improved performance and result in fewer device initialization errors.

7 Method

7.1 The procedural task environment

A variation of the task-environment introduced by Li (2006) and described in section 2 was used. The *Wicket Doughnut Call Centre* interface and related task were left the same, while some slight modifications were made to the interface for the doughnut-making machine (the modified interface is depicted in Figure 6):

1. A power button was added in the top right-hand corner to support the device model that was created for the doughnut-making machine. The device model explained operation of the machine in terms of how power flowed through the system, so providing participants with a way to turn the power on and off was important to reinforce this notion.
2. The name of the first machine component was changed from “Dough Port” to “Dough Depositor” to ensure that each component name enabled visualization of the operation it represented. The name “Dough Port” was difficult to associate with a corresponding action, while the remaining four component names were easily

associated with actions that could be visualized (e.g., as the “Puncher” component might invoke images of the dough being punched into different shapes, the “Dough Depositor” might invoke images of dough being deposited onto a surface).

3. A progress bar that was present in the Dough Depositor component was removed to make this component consistent with the other four components, none of which contained a progress bar.
4. In Li’s (2006) experiment, status text was presented briefly after each selector step was executed indicating that the component had been activated. Similar status text, saying “machine cleaned”, was added after the “clean” step (i.e., the PC step) had been executed. This was to make the outcome of the device-specific PC step consistent with the five selector steps.

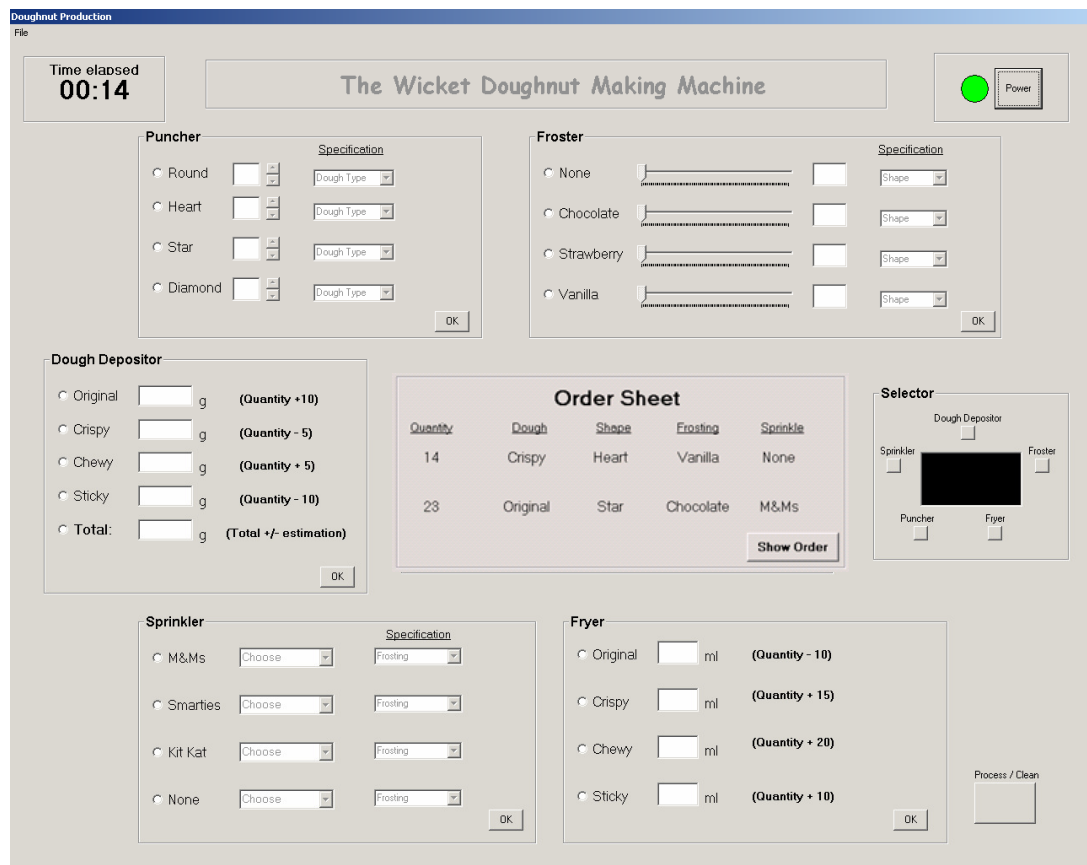


Figure 6. The modified Wicket Doughnut Making Machine interface.

The basic procedure for executing the preliminary call-centre task procedure and the main doughnut-making task procedure remained the same as in Li (2006), as detailed in Table 1 and Table 2.

Table 1. List of steps for completing the preliminary task procedure

	Step	Interface Action(s)	Machine
1.	Turn the power on	<ul style="list-style-type: none"> • Press the “Power” button 	Doughnut-making
2.	Enter the call location	<ul style="list-style-type: none"> • Select the location in the <i>customer location selector</i> • Press “Enter” 	Call-centre
3.	Register and submit the incoming order	<ul style="list-style-type: none"> • Find the location on the <i>customer location tube map</i> • Click and drag the location into the <i>customer order processor</i> • Press “Get Order” 	Call-centre
4.	Show the order details	<ul style="list-style-type: none"> • Press “Show Order” 	Doughnut-making

Table 2. List of steps for completing the main task procedure. The device initialization step is identified with bold text, and the postcompletion step with italic text.

	Step	Interface Action(s)	Machine
1.	Activate the dough depositor machine component	<ul style="list-style-type: none"> • Press “Dough Depositor” component selector button 	Doughnut-making
2.	Enter dough depositor data for first doughnut type	<ul style="list-style-type: none"> • Click dough type • Enter dough quantity 	Doughnut-making
3.	Enter dough depositor data for second doughnut type	<ul style="list-style-type: none"> • Click dough type • Enter dough quantity 	Doughnut-making
4.	Enter total dough depositor data	<ul style="list-style-type: none"> • Click total • Enter total quantity 	Doughnut-making
5.	Finish with dough depositor machine component	<ul style="list-style-type: none"> • Press “OK” 	Doughnut-making
6.	Activate the Puncher machine component	<ul style="list-style-type: none"> • Press “Puncher” component selector button 	Doughnut-making
7.	Enter Puncher data for first doughnut type	<ul style="list-style-type: none"> • Click shape • Enter quantity • Enter dough type 	Doughnut-making
8.	Enter Puncher data for second doughnut type	<ul style="list-style-type: none"> • Click shape • Enter quantity • Enter dough type 	Doughnut-making
9.	Finish with Puncher machine component	<ul style="list-style-type: none"> • Press “OK” 	Doughnut-making
10.	Activate the Froster machine component	<ul style="list-style-type: none"> • Press “Froster” component selector button 	Doughnut-making
11.	Enter Froster data for first doughnut type	<ul style="list-style-type: none"> • Click frosting type • Enter quantity • Enter dough shape 	Doughnut-making
12.	Enter Froster data for second doughnut type	<ul style="list-style-type: none"> • Click frosting type • Enter quantity • Enter dough shape 	Doughnut-making
13.	Finish with Froster machine component	<ul style="list-style-type: none"> • Press “OK” 	Doughnut-making
14.	Activate the Sprinkler machine component	<ul style="list-style-type: none"> • Press “Sprinkler” component selector button 	Doughnut-making
15.	Enter Sprinkler data for first doughnut type	<ul style="list-style-type: none"> • Click sprinkle type • Enter quantity • Enter frosting type 	Doughnut-making
16.	Enter Sprinkler data for second doughnut type	<ul style="list-style-type: none"> • Click sprinkle type • Enter quantity • Enter frosting type 	Doughnut-making
17.	Finish with Sprinkler machine component	<ul style="list-style-type: none"> • Press “OK” 	Doughnut-making
18.	Activate the Fryer machine component	<ul style="list-style-type: none"> • Press “Fryer” component selector button 	Doughnut-making
19.	Enter Fryer data for first doughnut type	<ul style="list-style-type: none"> • Click dough type • Enter quantity 	Doughnut-making
20.	Enter Fryer data for second doughnut type	<ul style="list-style-type: none"> • Click dough type • Enter quantity 	Doughnut-making
21.	Finish with Fryer machine component	<ul style="list-style-type: none"> • Press “OK” 	Doughnut-making
22.	Process the order	<ul style="list-style-type: none"> • Press “Process / Clean” 	Doughnut-making
23.	<i>Clean the machine</i>	<ul style="list-style-type: none"> • <i>Press “Process / Clean”</i> 	<i>Doughnut-making</i>
24.	Turn the power off	<ul style="list-style-type: none"> • Press the “Power” button 	Doughnut-making

7.2 Participants

Forty-eight individuals (24 men and 24 women) volunteered to participate, ranging in age from 20 to 67 and with a mean age of 30. All participants had either previously completed an undergraduate university degree or were currently enrolled in one at the time of participation, and all were fluent readers and speakers of the English language. Volunteers were either personally known to the experimenter (and received no compensation for their participation) or were recruited via the University College London Psychology subject pool (and received £4.00 for their participation).

7.3 Materials

The materials used in this experiment included paper-based descriptions of the low device relevance task-based goal (see Appendix 1) and high device relevance task-based goal (see Appendix 2), a paper-based diagram and description of the device model (see Appendix 3), three short paper-based quizzes used to evaluate comprehension of the paper materials (see Appendix 4 (i)), training material developed in Microsoft PowerPoint used to describe the task procedure in detail, two Microsoft Windows PCs with one running the call-centre code and the other running the doughnut-machine code (both written in Microsoft Visual Basic), and a post-study questionnaire printed on paper but delivered verbally by the experimenter (see Appendix 5).

7.4 Design

The experiment used a single-factor between participants design. The independent variable, type of training, initially had three levels:

1. **Control.** The control group was given the low device relevance goal (which was to focus on producing batches of doughnuts that matched customer orders), and received basic rote training that walked through the call centre and doughnut-making tasks step by step (the training was presented via PowerPoint).

2. **Machine tester.** The machine tester group was given the high device relevance goal, which was to evaluate the machine's visual feedback after each selector button was used to activate a component. They were also instructed that while executing the doughnut-making task they still had to enter data in each machine component because the machine would not allow them to proceed without it, but that it was *not* necessary to enter data that matched the doughnut order (as their goal was to test the machine's feedback and not to produce a specific number of doughnuts). They were also given a description of two specific problems with the machine's visual feedback that they should look out for: that the machine sometimes provides *no* visual feedback after pressing a selector button, and that it sometimes provides the *wrong* visual feedback. Although neither problem actually occurred during the experimental trials, it was thought that providing specific scenarios to test for might help emphasize participants' role as machine testers. Their training on the procedure was otherwise the same as the training for the control participants, except the PowerPoint presentation included visual examples of the two problems they had been advised to look out for.

3. **Device model.** Like the control group, the device model group was given the low device relevance goal, but also studied the device model materials (provided in Appendix 3) prior to being trained on the procedure. Their training was the same as for the control participants, but also visualized the current state of power flow before and after each device interaction, in a manner that was consistent with the device model they had studied.

Roughly half way through running the experimental trials it became clear that the training devised for the machine tester group was not having the desired effect of focusing participants on a different goal; despite having only been exposed to the HDR goal, they were focusing much more strongly on the LDR goal of baking doughnuts to match customer orders (likely because the interface design for the doughnut-making machine so naturally emphasized this goal). As such, a fourth level of training was also introduced: **machine tester-enhanced**. These participants also received the HDR goal and the same training as the machine tester participants, with one small difference. In addition to being informed that it was not necessary to enter data that matched the doughnut orders, they

were also instructed to enter the value of “1” (or some other arbitrary number) for the quantities in the data entry steps of the procedure (i.e., steps 2-4, 7-8, 11-12, 15-16, and 19-20 in Table 2). This was expected to reinforce the notion that matching the doughnut orders was not related to successful completion of their testing goal.

Table 3. Summary of the differences in the training material presented to the four groups.

Condition	Task-based goal	Additional instructions or information	Enhancements to the PowerPoint material
Control	Low device relevance	<ul style="list-style-type: none"> • None 	<ul style="list-style-type: none"> • None
Tester	High device relevance	<ul style="list-style-type: none"> • Do not need to enter accurate data • Instructed to look out for two specific feedback problems: <i>no</i> feedback, and the <i>wrong</i> feedback 	<ul style="list-style-type: none"> • Enhanced with visual examples of the <i>no</i> feedback, and the <i>wrong</i> feedback problems
Tester-enhanced	High device relevance	<ul style="list-style-type: none"> • Do not need to enter accurate data • Instructed to enter arbitrary quantities in the data-entry steps • Instructed to look out for two specific feedback problems: <i>no</i> feedback, and the <i>wrong</i> feedback 	<ul style="list-style-type: none"> • Enhanced with visual examples of the <i>no</i> feedback, and the <i>wrong</i> feedback problems
Device model	Low device relevance	<ul style="list-style-type: none"> • Studied the device model topology diagram, which described how power flowed through the system 	<ul style="list-style-type: none"> • Enhanced to visually show the flow of power through the system as steps were executed

The resulting experimental design therefore was a single-factor between participants design with four levels: (1) control, (2) tester, (3) tester-enhanced, and (4) device model; the differences in training between each level are summarized in Table 3. Participants were randomly assigned to each group, and the main dependent measure was the number of omission errors made on the device initialization step during execution of the doughnut-making procedure. Data about omissions on all other steps was also recorded, as well as the timestamp for each device interaction and the overall trial-completion time. Additional dependent measures that were of interest were the number of correctly specified doughnut orders, and responses to the post-study questionnaire.

7.5 Procedure

Participants in all groups read the introductory, paper-based material that described their role (baker or machine tester) and corresponding task-based goal during the

experiment (LDR or HDR). The *device model present* group also studied the device model materials. None of the groups were shown the machine interfaces during this phase.

After reading through the introductory material, a brief pre-training quiz was administered to participants in all groups to ensure that the information presented had been adequately understood and internalized (see Appendix 4 (i)). If a participant failed to answer a question correctly, they were asked to reexamine the introductory material and the quiz was administered again.

Upon successfully completing the quiz participants were provided training on the procedural task using a PowerPoint presentation. The interfaces for the Wicket Doughnut Call Centre and the Wicket Doughnut Making Machine were revealed for the first time, and the purpose of both machines was explained along with the role of all device controls. The step-by-step procedure that had to be followed in order to use the call-centre and doughnut-making machines together was then described at a high level. Subsequently, a more detailed description was presented using animated sequences to demonstrate the precise mouse movements and device interactions required at each phase of the procedure; the corresponding enhancements for each condition, as outlined in Table 3, were also presented. Finally, the participant's task-based goal was reiterated one last time.

After training was complete participants moved on to the practice phase. Each participant executed the call-centre and doughnut-making tasks described in tables Table 1 and Table 2 using the machine interfaces, until they had completed two trials in a row without difficulty (this took an average of only 2-3 trials). When errors were made, the computer issued a simple dialog with the message "An error has been made. Please correct it and carry on." Participants had to press the OK button to dismiss the dialog, then detect and correct the error before proceeding (it is worth noting that very few participants made any errors at all during this practice phase).

Participants then completed ten experimental trials without the experimenter present, and were asked if they would like to take a break after the fifth trial. At the end of each trial, after the step for processing the order (step 22), a report was presented indicating either that the correct number of doughnuts had been made, or that the order was off by a given number of doughnuts. This report was not based on data actually entered by participants; a positive report was always shown on trials 1, 4, 6, 7, and 8, and a negative report was always shown on trials 2, 3, 5, 9, and 10. During the experimental trials a warning message was not presented when a procedural error was made (e.g., omitting a

step), but the interface was designed such that participants were unable to proceed to the next step until they detected occurrence of the error and corrected it.

After all ten trials were complete, participants participated in a short post-study interview in which they were asked to answer four questions (presented in Appendix). The first question was designed to provide insight into whether the relevance of the device initialization step was raised in any of the experimental conditions; the second question was a simple check to verify that no aspects of the procedure had been overemphasized during training in a way that would have unexpectedly influenced behaviour; the third question was included to identify what participants personally emphasized during the experimental trials; and the last question solicited general feedback about the experiment. The experimenter recorded participants' responses on paper. The whole procedure lasted approximately one hour.

8 Results

All data was included in the analysis. The purpose of the experiment was to examine the effect of different internal knowledge acquired through different training material on the device initialization error rate, which occurs when the first step in a procedural task is omitted. Errors were counted for each of the ordered task steps in the main procedure (see Table 2 for the full set of task steps). As in Li's (2006) work, attempting to execute a step incorrectly or out of order was counted as an error for the associated step, but the number of actions taken before the error was corrected did not contribute to the error count. For example, if a participant omitted pressing the Sprinkler's *OK* button (step 17 in Table 2) by proceeding directly to activate the *Fryer* component (step 18 in Table 2), this would be counted as an error for the *Sprinkler OK* step. Even if the participant clicked on the Fryer selector button 4 times before realizing her mistake and returning to select the Sprinkler's *OK* button, this would only be counted as a single error.

8.1 Overall errors

The total number of procedural errors across all 48 participants was 305; 92 were committed by control participants, 81 by tester participants, 48 by tester-enhanced participants, and 84 by device model participants. Of the 48 participants, 31 made at least

one device initialization error, and the device initialization errors account for 39.33% of the total errors.

The systematicity of errors, or *error rate*, at each task step was assessed by examining the number of error occurrences at a given step in relation to the number of opportunities for that error overall (see Byrne & Bovair, 1997, and Li, 2006). Since each of the 48 participants completed 10 trials, the total number of opportunities for each error was 480. The error rate at each step is illustrated in Figure 7; error rates that are above the .05 (or 5%) level can be considered to occur systematically (as in Byrne & Bovair, 1997). Consistent with Li's (2006) experiments, the device initialization step, which is the first step in the doughnut-making task procedure, exhibited the highest error rate across all task steps (overall error rate = .246), followed by the PC step, which is the last step in the doughnut-making task procedure (overall error rate = .146), and both can be considered to have occurred systematically. The error rate for the *Fryer's* selector step was also found to be above the .05 level (overall error rate = .077).

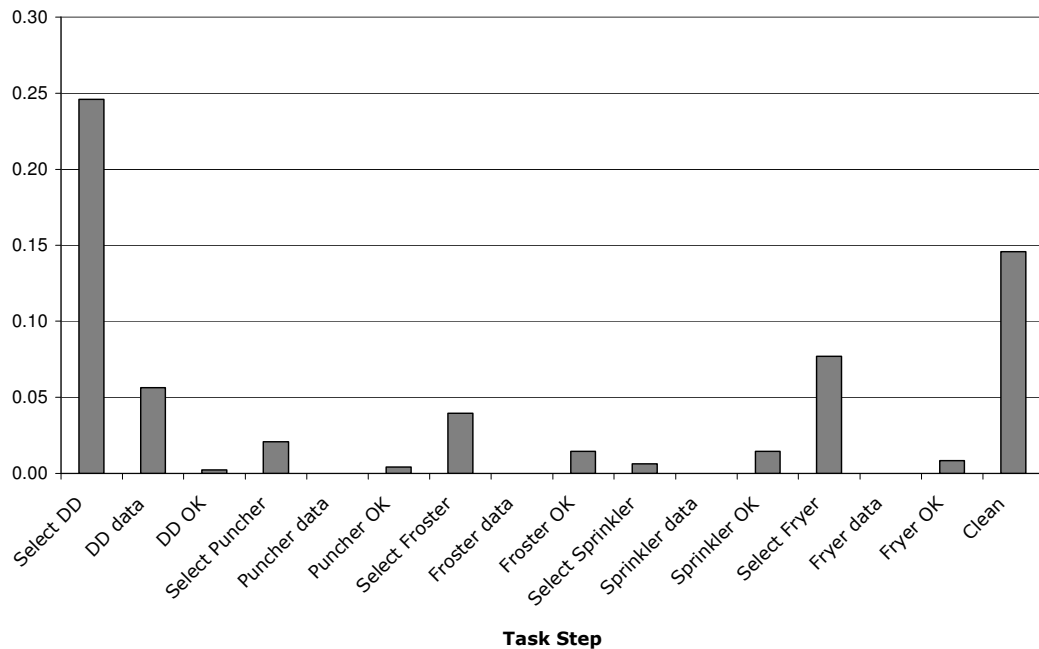


Figure 7. Error rates for task steps in the main procedure. Error rates above .05 suggest a systematic error.

8.2 Device initialization errors

Mean error rates for the device initialization step (i.e., the step for pressing the Dough Depositor selector button at the beginning of the doughnut-making task) for each of the 4 experimental conditions are displayed in Table 4, and also depicted in Figure 8. The error rates were computed by dividing the number of device initialization errors in each condition by the number of opportunities for that error.

Table 4. Mean error rates and standard deviations for the device initialization error, by condition.

	Mean Error Rate	Standard deviation
Control (<i>N</i> =12)	.308	.300
Tester (<i>N</i> =12)	.267	.328
Tester-enhanced (<i>N</i> =12)	.100	.237
Device model (<i>N</i> =12)	.308	.257

Planned contrasts were conducted using the Wilcoxon Mann-Whitney rank-sum test to assess whether there was a significant difference between the error rates in the control condition and each of the experimental conditions. A reliable difference was found between the control and tester-enhanced groups (*Mann-Whitney U* = 30.5, *Wilcoxon W* = 108.5, *Z* = -2.527, *p* = .011), indicating that participants in the tester-enhanced group were able to develop superior internal cues to prime the device initialization step at the appropriate time, while the control participants were not. This result supports the hypothesis that relevance of the initial step to the task-based goal plays an important role in the likelihood of its omission during execution of a procedure, as execution of the initial step was critical to successful completion of the tester-enhanced participants' HDR goal but not to the control participants' LDR goal. However, no reliable differences were found between the control and tester groups (*Mann-Whitney U* = 60.5, *Wilcoxon W* = 138.5, *Z* = -.674, *p* = .500), despite the tester participants being instructed to focus on the same task-based goal as the tester-enhanced participants. The absence of a similar reduction in error rate for the tester group suggests that increases in the cognitive salience of the device initialization step are extremely sensitive, as both groups were exposed to very similar training prior to the experimental trials; this issue will be explored further in the discussion section.

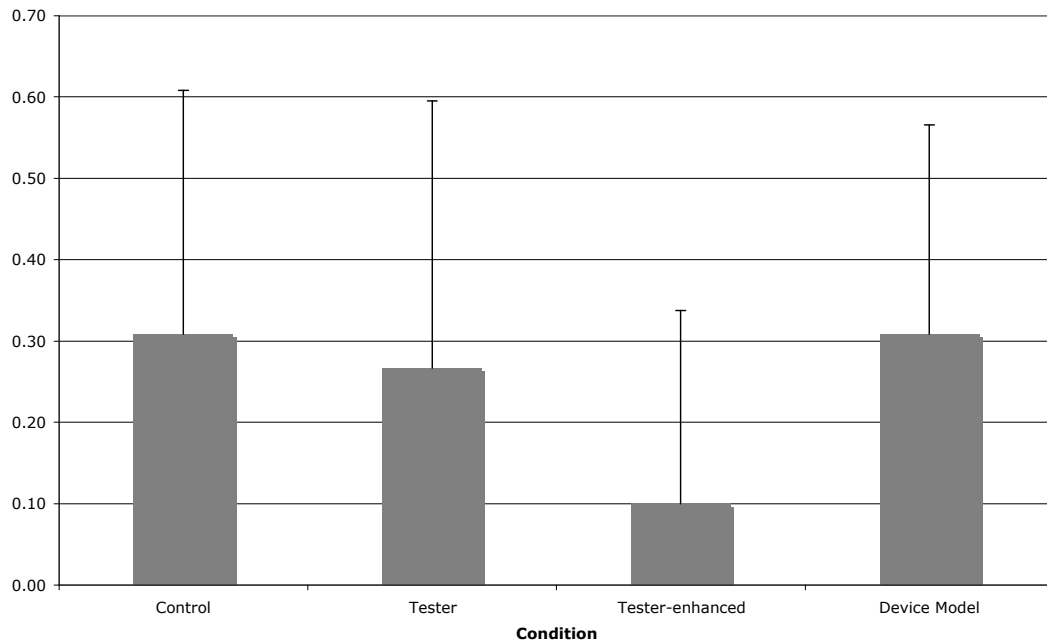


Figure 8. Mean error rates for the device initialization step in each condition, with error bars identifying the corresponding standard deviations.

A reliable difference was also not found between the control and device model groups (*Mann-Whitney* $U = 69.5$, *Wilcoxon* $W = 147.5$, $Z = -.146$, $p = .884$). This indicates that knowledge of the device's conceptual model and the role of the initial selection step in the overall device operation does not provide a sufficient internal cue to prime the device initialization step at the appropriate time. Implications of these results will also be discussed separately in the discussion section.

8.3 Categories of errors

Post-hoc analyses were conducted in order to further explore the impact of the training manipulations on the cognitive salience of different types of steps during execution of the main procedure. Errors made during the doughnut-making task were grouped into three categories: selector errors (errors made on any of the 5 selector steps; these are analogous to the *skip-selector* errors reported by Li, 2006), postcompletion errors (errors made on the *clean* step, the last step of the procedure), and data-step errors (errors made on the remaining procedural steps, which were related to entering data for the doughnut

orders). Table 5 shows the mean error rates across all experimental conditions (note that the data is not normally distributed).

Table 5. Mean error rates for selector errors, postcompletion errors, and data-step errors, by condition (data not normally distributed).

	Selector errors		Postcompletion errors		Data-step errors	
	Mean Error Rate	Standard deviation	Mean Error Rate	Standard deviation	Mean Error Rate	Standard deviation
Control (<i>N</i> =12)	.090	.077	.233	.357	.019	.018
Tester (<i>N</i> =12)	.097	.109	.100	.148	.022	.024
Tester-enhanced (<i>N</i> =12)	.027	.048	.058	.090	.043	.036
Device model (<i>N</i> =12)	.098	.046	.192	.274	.010	.010

Using a Wilcoxon Mann-Whitney rank-sum test, significant differences were found between the control and tester-enhanced groups: tester-enhanced participants exhibited a reliably lower error rate on selector steps overall (*Mann-Whitney* $U = 28.500$, *Wilcoxon* $W = 106.500$, $Z = -2.577$, $p = .010$), and exhibited a higher error rate on data steps (*Mann-Whitney* $U = 37.500$, *Wilcoxon* $W = 115.500$, $Z = -2.037$, $p = 0.042$). Despite showing an apparent trend towards a reduced error rate, differences on the postcompletion step were not significant (*Mann-Whitney* $U = 58.000$, *Wilcoxon* $W = 136.000$, $Z = -.889$, $p = 0.374$). These differences in categorical error rates indicate that the tester-enhanced manipulation improved internal cues for priming the selector steps while simultaneously weakening internal cues for the data steps in the procedure. This result supports the notion that it is possible to influence the cognitive salience or relevance of different steps in a procedure through exclusively internal manipulations, and further suggests that a step's relation to the task-based goal is an important factor in the likelihood of an error occurring: selector steps were more important to accomplishing the tester-enhanced participants' goal than the control participants' goal, and the corresponding error rate on these steps was lower for the tester-enhanced participants; in addition, the data steps were less important to accomplishing the tester-enhanced participants' goal than the control participants' goal, and the corresponding error rate on these steps was higher for the tester-enhanced participants.

Again, no differences were found between the tester and control groups, implying that the tester manipulation did not have an overall impact on the importance of different

classes of steps. The device model group also did not show any significant differences from the control group, despite the device model group having access to more sophisticated information about how the device-specific steps related to the internal structure of the system. This suggests that during the execution of procedural tasks, different knowledge about the inner workings of a system will not necessarily result in differences in perceived importance of the procedural steps.

8.4 Qualitative results

Responses to the four post-study questions were reviewed for all participants (see Appendix 5 for the precise wording for each question). Question 1 was intended to help evaluate how relevant participants perceived the device initialization step to be, based on the task-based goal they were instructed to focus on during training. The question asked participants to describe how they fulfilled their task-based goal, and it was expected that participants who acted on the LDR goal (control and device model participants) would be less likely to mention the device initialization step in their verbal descriptions as it was less relevant to their goal, while participants who acted on the HDR goal (tester and tester-enhanced participants) would be more likely to mention it.

Table 6. Example segments of responses to question 1 that were included in the *explicitly mentioned* (EM) response category.

Control	Tester	Tester-enhanced	Device model
“click on the first one, dough something, to activate the thing”	“select the first thing”	“next start selecting functions, first the dough depositor”	“switch the power to the dough depositor”
“click the dough selector”	“click on the dough depositor button”	“start with the first of five, click it, and check the feedback”	“turn on the dough one”
			“then change power to the dough depositor”

Responses to this question were found to fall into three categories. The *explicitly mentioned* (EM) category included those responses in which participants directly mentioned executing the dough depositor selection step by name, or by its position in the sequence (see examples presented in Table 6). The *indirectly mentioned* (IM) category included those responses in which participants mentioned executing the selector steps as a

whole, but did not specifically mention (or exclude) the dough depositor selection step (see examples presented in Table 7). The *not mentioned* (NM) category included all other responses, in which the selector steps were entirely left out of the description, or other selector steps were mentioned explicitly but the dough depositor selector step was not.

Table 7. Example segments of responses to question 1 that were included in the *indirectly mentioned* (IM) response category.

Control	Tester	Tester-enhanced	Device model
“activate each of the 5 steps”	“select the indicator for each to see if its working”	“you need to select all the component things”	“I had to remember to activate all the individual machines’ power”
“you have to make sure to click each button”	“go through the 5 steps, select it. But sometimes I forgot.”	“the tests were to find out whether the machine gave you feedback after clicking”	“follow the procedure of switching between tasks in the right order, then fill in details”

Table 8 shows the distribution of responses between the three categories. There are no distinguishable patterns to the responses when examined by condition, and interestingly over 80% of the participants overall mentioned the execution of the device initialization step during their verbal descriptions of how they accomplished their goal: 50% explicitly mentioned it, and a further 31% referred to it implicitly. This confirms that any device initialization errors that were committed were likely not knowledge based, and in addition it suggests participants in all four conditions perceived the relevance of the device initialization step to be similar. That the corresponding omission rates for the step are contrastingly high in three of the four conditions (control, tester, and device model) introduces the possibility that something in the actual act of executing the procedure in these conditions interfered with the reliable execution of the step, or resulted in certain steps being primed over the device initialization step. However, it is difficult to make any concrete claims based on this informal measure of relevance; a more precise measure might indicate completely different relevance scores. This is an area that might benefit from further research.

Table 8. Number of responses to question 1 that fell into the *explicitly mentioned* (EM), *implicitly mentioned* (IM), and *not mentioned* (NM) categories, per condition.

Response	Control	Tester	Tester-	Device	Percentage of
----------	---------	--------	---------	--------	---------------

Category			enhanced	Model	total responses
EM	7	4	6	7	50%
IM	4	6	3	2	31%
NM	1	2	3	3	19%

The second question in the post-study interview was included as a sanity check to verify that the training process hadn't inadvertently emphasized parts of the procedure differently, in ways that might have unexpectedly influenced performance, across the conditions. Participants were asked if any of the instructions or training material stood out or regularly came to mind as they were executing the trials. There were no remarkable differences in responses between the conditions, and as such the response data for this question will not be reported in detail.

The third question was intended to evaluate what participants truly emphasized during the experimental trials, and if that correlated with what was emphasized in the training; the training material for each condition was designed to encourage participants to focus on different task-based goals, and it was important to understand how effective that training was in guiding their focus.

Table 9. The most frequent responses to question 3 in the post-study interview, which examined what participants emphasized during the experimental trials.

Rank	Control	Tester	Tester-enhanced	Device Model
1	Accuracy of data entry (7 responses)	Testing the machine's feedback (9 responses)	Testing the machine's feedback (7 responses)	Accuracy of data entry (8 responses)
2	Getting the math / quantities / calculations right (5 responses)	Accuracy of data entry (7 responses)	Speed (4 responses)	Speed (5 responses)
3	Speed (5 responses)	Speed (3 responses)	Following the procedure in order (3 responses)	Use of energy / power settings (4 responses)

Participants were asked what they had considered most important when they were executing the experimental trials, and the top three responses for each condition are provided in Table 9 (note that participants were not restricted to a single answer). The responses indicate that the provided training was effective in communicating the task-based goal, and that in practice participants correctly emphasized the corresponding goal: control and device model participants primarily emphasized the need to accurately enter the data

presented in the doughnut orders (the LDR goal), while tester and tester-enhanced participants primarily emphasized the need to test the relevant feedback (the HDR goal). That tester-enhanced participants did not report an emphasis on accuracy of data entry at all corresponds with the finding that error rates on the data entry steps were higher for this group than for the control group, further supporting the notion that salience of these steps was decreased.

The last part of the post-study interview was an extension to the previous question, to determine if there was anything particular to the interface or task design that caused significant frustration or annoyance and therefore may have drawn attention away from the primary task. The responses to this question varied widely, and most were related to general concerns about how data was handled (e.g., that there was no consistency between how quantities were entered, and that the labels in the order sheet didn't match the corresponding component labels where the data was entered). However, two interesting issues were raised.

First, of the 48 respondents, five in the tester condition and two in the control condition reported being specifically frustrated by having to mentally perform math calculations on some of the data entry steps. One of these participants elaborated on her response, reporting that she felt her "stress level" change depending on which component she was about to begin, because she "hates dealing with numbers." While this is only an informal account, Ashcroft (2002) suggests that such feelings are indicative of a specific form of anxiety known as math anxiety, which is characterized by fear, tension, or apprehension that can interfere with performance. Ashcroft and Kirk (2001), and Ashcroft (2002) have reported a link between high math anxiety and degraded performance on dual tasks when one task involves mental arithmetic, suggesting that the degradation stems from a temporary taxation of working memory resources due to the anxiety reaction rather than to inherent differences in working memory span. Given the reasonably high number of participants in the present study who volunteered information about math-specific frustrations, it seems worthwhile to consider the possibility that math anxiety may have played a role in participants' performance while executing the procedure, especially given that the only two out of five selector steps that exhibited error rates above the .05 level of systematicity were also the only two that immediately preceded data entry steps involving math calculations (i.e., the dough depositor and fryer steps, as depicted in Figure 7). Potential implications of this will be discussed further in the discussion section.

The second interesting result was that half of the tester-enhanced participants reported that they didn't experience any frustration during the trials, whereas all participants in the other conditions reported multiple points of frustration. This is likely related to the fact that the tester-enhanced participants were specifically instructed not to focus on entering the data accurately, which is what most of the concerns raised by the remaining participants were related to. This provides further evidence that tester-enhanced participants focused on their intended goal and were not sidetracked by the need to enter accurate data.

9 Discussion

The device initialization error, first reported by Li (2006), is a systematic procedural error that appears to have some characteristics in common with the commonly reported postcompletion error: they both occur on steps that appear to play a secondary role in accomplishment of the task-based goal, and both occur on steps that reside at the boundaries of goal activity (i.e., the device initialization step occurs just before the first step that represents progress towards the task-based goal, and the PC step occurs just after the task-based goal is achieved). Previous work on PCEs has sought to understand the role of the task structure in PCEs (Reason, 2002) and investigated external factors as mitigators to PCEs (e.g., Chung & Byrne, 2004; Li et al, 2005), but has rarely investigated the possibility of developing internal cues to raise the salience of PC steps in working memory in an attempt to reduce omissions without physically altering the device interface or task structure. One exception is the work reported by Back et al (2006) who investigated the role of motivation in influencing PCE occurrence, and found that introducing penalties did not have a reliable effect on error rates. This thesis explored two different approaches to increasing the cognitive salience of the error prone device initialization step in the same task used by Li (2006) to investigate PCEs, the results of which will be discussed in turn.

9.1 Approach 1: Improving cognitive salience by modifying the task-based goal

The first approach explored in this thesis was to raise cognitive salience of the device initialization step by changing participants' task-based goal, such that its achievement directly depended on execution of the step.

9.1.1 Interpretation of tester results

A non-significant difference was found between the device initialization error rates in the control and tester groups, suggesting that directly linking the device initialization step to accomplishment of the task-based goal is not an effective strategy for elevating the cognitive salience of that step. However, closer examination of the available data provides further insight into the null result.

First, it is important to look beyond the experimental manipulation of the task-based goal that participants in the tester group were presented with and trained on, to consider their actual behaviour. Participants in the tester group were instructed during training that entering data that precisely matched the doughnut orders was not necessary, as their primary goal was to test the selector feedback. However, an inspection of the data log files for the tester group reveals that all but one of the twelve participants entered data that closely matched the doughnut orders displayed on screen for each trial, despite being instructed that this was unnecessary. All twelve participants in the control condition also did so. Further, data from question 3 (the *emphasis* question) of the post-study interview reveals that the same number of participants in both the control and tester condition reported that they personally emphasized accuracy of data entry during the experimental trials, even though the tester participants were not required to do so. This suggests that while participants in the tester group did focus on the primary goal of testing the machine's feedback (as indicated by the fact that this was the most frequently reported answer to question 3), they also emphasized the same goal as those in the control condition (which was to produce the correct amount of doughnuts ordered). This implies that the tester participants actually acted on the same goal as the control participants, and simply *added* the additional goal of testing the selector feedback. Precisely why participants were so strongly drawn to the data entry tasks should be explored in future work, but is likely related to the fact that this is what the interface was primarily designed to support. The task report presented at the end of each trial, which regularly reported that the incorrect number of doughnuts had been made, might have further encouraged participants to focus on correctly entering data (as the negative report would have suggested they had done something wrong).

Taking this more detailed information into account allows for a more precise interpretation of the lack of difference between the control and tester error rates. To summarize:

- Control participants received training that strongly emphasized the importance of accurately entering the doughnut order data (a low device relevance goal), whereas tester participants received training that strongly emphasized the importance of testing the machine's selector feedback (a high device relevance goal).
- Control and tester participants both reported emphasizing the need to accurately enter the doughnut order data during the experimental trials, and in practice both expended the effort to enter accurate data.
- Only the tester participants reported also emphasizing the need to attend to the machine's selector feedback during the experimental trials.
- The error rate on the device initialization step was not significantly different between the two groups.

Taken together, this suggests that taking on the *additional* goal of focusing on the outcome of the device initialization step (in this case, the resulting visual feedback) does not on its own provide a sufficient internal cue to prime the step at the appropriate time during task execution. Further, the data demonstrates that device users might readily adopt additional goals that were not emphasized in their training on the device, especially if the device design itself naturally encourages it.

9.1.2 Interpretation of the tester-enhanced results

That a significant difference was found in the error rates between the control and tester-enhanced groups is very encouraging, as it indicates that it is possible to achieve a reduction in the device initialization error rate without physically changing the device interface or task structure, as has often been the recommended approach for reducing the risk of PCEs. Again, a more fine-grained analysis of the data allows a better understanding of these results and their implications.

As with the tester group, participants in the tester-enhanced group were instructed that it was not necessary to enter data that matched the doughnut orders on each trial. In addition, tester-enhanced participants were instructed to enter the value of "1" (or some other arbitrary number) for the quantities in the data entry steps in order to reinforce the notion that matching the doughnut orders was not related to successful completion of their

testing task. Inspection of the data log files indicates that during the experimental trials participants in this condition appear to have essentially ignored the order details, only entering data that resembled the presented doughnut order on 9 trials out of 120 (on the remaining trials it appears they entered data arbitrarily, although detailed analysis was not conducted to identify any existing patterns). In contrast, the control participants entered matching (or closely matching) data on all trials. This confirms that the tester-enhanced participants did not focus on the task-based goal of accurately making doughnuts. Further, responses to question 3 (the *emphasis* question) indicate that they correctly focused on the intended task-based goal of testing the selector feedback, and again confirms that accuracy of data entry was not emphasized.

That this difference in behaviour from the tester participants is also correlated with a significant decrease in the device initialization error rate compared to control participants highlights a direct relationship between the two events; focusing exclusively on the task-based goal of testing the selector feedback, rather than focusing on this goal in addition to the goal of accurately filling doughnut orders, appears to have led to an increase in the cognitive salience of different procedural subgoals, most importantly the goal associated with the device initialization step. The observed differences in error rates for the broader categories of selector and data-entry errors between control and tester-enhanced participants also provides more general evidence for the changes in cognitive salience of different subgoals in the procedure, suggesting that not only can a goal's salience be increased by making it more relevant to the task-based goal, but also that its salience can be decreased. The fact that a similar effect was not observed for the tester participants highlights once again that a critical difference appears to have been that tester-enhanced participants focused on a single task-based goal.

As such, while the results from this study suggest that the goal a participant focuses on can influence their performance, they also demonstrate that when multiple task-based goals related to a given procedural task are present it is difficult to predict which one(s) participants will choose to focus on, and it is also difficult to encourage them to focus on one over another. This is consistent with the results reported by Back et al (2006) and described in section 6. In their study of the effect of motivation on PCEs, knowledge of the association between executing the PC step and achieving a high score did not result in a significant decrease in the PC error rate. However, the task used in their experiment also involved two task-based goals: the first goal being to progress forward in the game, which

involved avoiding alien fire, shooting alien ships, capturing aliens, and rescuing astronauts in order to progress to subsequent levels; the second goal, designed to motivate correct execution of the error prone PC step, was to achieve a high score. Their results indicate that making a PCE was more likely when the first goal, making progress in the game, was at risk; this suggests that despite being encouraged to focus on achieving a high score, participants' behaviour was more strongly driven by the goal of moving forward in the game.

The results of this study support the possibility that replacing a task-based goal in which the device initialization step plays a secondary role with one in which it plays a more central role can contribute to a significant reduction in the error rate on that step. However, an alternative explanation is that the change in focus fundamentally changed the task in a way that eliminated or reduced the influence of a different contributing factor. For example, by not focusing on the goal of accurately filling doughnut orders, the tester-enhanced participants also did not have to expend the effort to transform information in the doughnut-making machine's Order Sheet at each stage of the procedure to enter it into the corresponding data fields. While for the most part the data transformation was rather straightforward (as simple as entering data from the "shape" column into the "puncher" component), the task was repetitive and required participants to continuously refer back to the Order Sheet to retrieve different parts of the order. Although participants in the tester-enhanced condition still executed the identical procedure as control participants in terms of interacting with the device, the cognitive load imposed by the task would have been less as there was no need to engage working memory in order to find the relevant data in the Order Sheet and transfer it into the correct component fields. Working memory demands imposed by the task environment have been shown to have a significant impact in the occurrence of PCEs (Byrne & Bovair, 1997), which appear to share some common characteristics with the device-initialization error. As such, the possibility that a reduction in the working memory demands for participants in the tester-enhanced condition may have contributed to the corresponding reduction in the device-initialization error rate must also be considered. This issue should be pursued in more detail in subsequent research.

It is important to note that the tester-enhanced treatment didn't completely eliminate the device initialization error, as participants still produced an error rate of .10 (about 1/3 the size of the error rate for control participants). However, these results do provide a foundation for beginning to understand factors that might contribute to the error rate and

that are worthy of further investigation, such as the relevance of the step to the task-based goal, the number of task-based goals related to the task and their perceived importance, and task complexity.

9.2 Approach 2: Improving cognitive salience through mental models

The second approach explored in this thesis was to provide participants with a conceptual model of the device (called a *device model*, as in Kieras and Bovair, 1984) that emphasized the central role of the device initialization step in the device's operation as a mechanism to control power to the dough depositor component. Based on previous results indicating the positive influence of providing a device model on learning to interact with simple devices (e.g., Kieras & Bovair, 1984), it was expected that participants who learned this conceptual model prior to interacting with the device would develop a corresponding mental model that also integrated the step's central role, thereby increasing its cognitive salience.

No difference was found between the control and device model groups on the device initialization error rate, suggesting that this error likely does not stem from an inadequate mental model of the device, and also that learning a conceptual model that emphasizes the role of the device initialization step does not result in a sufficient internal cue to prime the step during execution of the procedure. Further, that no differences were found between the two groups on the broader error categories of selector, postcompletion, and data-step errors (as reported in Table 5), indicates that knowledge of the device model provided no performance advantage at all during execution of the procedure. This reveals the limited role of mental models in informing behaviour during procedural tasks.

It should be mentioned that participants in the device model condition exhibited clear signs that the information provided in the device model did affect the mental representations formed. In their responses to question 1 in the post-study interview (see examples in tables Table 6 and Table 7 in the results section), participants in the device model condition spoke primarily in terms of power flow when describing how they accomplished their goal (e.g., they spoke of “activating power to” the component, or “changing the power”). This is in contrast to participants in the other three conditions, who spoke primarily in terms of “clicking”, “pressing”, or “selecting” the selector button, which are more interface-level descriptions of the actions. Similarly, Kieras and Bovair (1984)

reported that participants in their model group “explained their actions and the device behaviour almost completely in terms of the model”. This suggests that the concept of power flow, and the specific role that the selectors played in controlling power flow, were successfully integrated into participants’ mental models. However, this did not result in an improvement in their performance.

This is consistent with work reported by Canas, Bajo, and Gonzalvo (1994), who evaluated the mental models formed by novice computer programmers and demonstrated that the availability of different functions in the programming tools used to teach programming concepts can affect the mental representations formed. For example, they found that exposure to the “trace” functionality while learning to program in *C* resulted in mental representations that emphasized the semantics of the language, whereas lack of exposure to this feature resulted in mental representations that emphasized syntactic elements (the trace function allows programmers to follow the flow and logic of code while debugging programs). Semantic organization is more in line with the mental representations exhibited by expert programmers (Canas et al, 1994), so was expected to be related to superior performance by the novices. However Canas et al reported that performance during programming and debugging tasks was not significantly different between the two groups despite the differences observed in participants’ mental representations.

The work reported in this thesis provides further evidence that differences in mental representations do not necessarily translate to differences in performance, and therefore contributes to the debate about the overall role of mental models in influencing performance. While having detailed knowledge of a system’s internal mechanisms has been shown to improve performance in novel problem solving tasks (Halasz & Moran, 1983), and in learning and retaining the operating procedures for a device (Kieras & Bovair, 1984), it does not appear to have an effect on performance in procedural tasks, especially on the likelihood of committing certain classes of omission errors.

9.3 The potential role of math anxiety

The previously noted reports of math-related frustrations by some participants in the current study hint at the possibility that math anxieties were present. Ashcroft (2002) reported that an individual’s self rating on a single question about their level of math anxiety can correlate strongly (anywhere from 0.49 to 0.85) with scores on a shortened version of the Mathematics Anxiety Rating Scale (MARS; Richardson & Suinn, 1972), and

as such there is a distinct possibility that at least those participants in the present study who volunteered this information without being specifically asked about it suffer from a form of math anxiety.

The effects of math anxiety that may be relevant to participants' performance in the present study are twofold. First, Ashcroft and Kirk (2001) reported a deterioration in performance by high math anxiety individuals on dual tasks when one of those tasks involved mental computation of math sums, due to increased demands on working memory. In the three conditions in this study in which participants performed mental arithmetic during the data entry steps (i.e., the control, tester, and device model conditions), systematic error rates were observed on the device initialization step (i.e., the dough depositor selector step) and the fryer selector step, both of which occur immediately prior to the math calculation steps.

Second, Ashcroft and Kirk (2001) reported that performance degradation was more severe when the mental arithmetic tasks involved multi-column addition with carrying. This is relevant because in the present study, the data entry task following the device initialization selector step involves computing three sums: to compute the amount of dough required for the first doughnut type; to compute the amount of dough required for the second doughnut type; and to compute the total amount of dough required by mentally adding the two previous values. The first two sums are quite simple and are comparable to those required in the fryer data entry task, involving addition or subtraction of simple values such as 5 or 10. However, the last step involves computation of a multi-column sum that often requires carrying. It is intriguing that the more complex data entry task is also associated with the highest error rate on the preceding selector step.

Based on Ashcroft (2002) and Ashcroft and Kirk (2001)'s work, the relationship between the presence of mental arithmetic steps and systematic selector errors in this procedural task might be explained by constraints on working memory imposed by math anxieties. It is possible that knowledge of the upcoming math steps invoked an anxiety reaction in certain individuals which demanded some of their available working memory resources, thereby interfering with their ability to correctly retrieve the appropriate selector step at the appropriate time. Considering this in terms of Altmann and Trafton's (2002) goal activation model, an anxiety reaction itself may serve as an internal cue to the data entry steps, strengthening their activation in working memory and increasing the overall

interference level, as well as increasing the likelihood that the first data entry step is sampled over the appropriate selector step.

Not only might this help explain the particularly high error rates on the device initialization and fryer selector steps over the other three, but it might also provide further insight into why the task-based goal manipulation appears to have had an effect on the device initialization error rate in the tester-enhanced condition but not in the tester condition. Five of the seven participants who reported frustration by the math steps were from the tester group, which (loosely) suggests this group may have included at least some math-anxious participants. It is possible then that anxiety reactions acted as internal cues that served to strengthen activation to the data entry steps for these participants, potentially competing with any strengthening of the device initialization step that was introduced by changing their task-based goal. In contrast, the tester-enhanced participants did not complete any math steps and therefore should not have experienced any additional strengthening of the data steps from anxiety reactions. This also raises the very interesting question of what the corresponding decline in error rate for tester-enhanced participants compared to control participants should be attributed to: increased cognitive salience of the selector steps, stemming from their more direct relationship to the accomplishment of the task-based goal; decreased cognitive salience of the data entry steps, stemming from the removal of potentially anxiety provoking mental arithmetic; or perhaps to some combination of the two. This appears to be a worthwhile question to pursue in future research.

9.4 Implications for training and device design

The results reported here have important implications for the role of training material in driving the way people interact with devices, by demonstrating its potentially small role in participants' actual use of a device. Specifically, the analysis of user behaviour in the tester condition shows that regardless of the task-based goal and procedural steps that are emphasized in training, individuals may inadvertently be drawn to adopt additional goals and emphasize different steps based on the device design itself. This demonstrates the incredible power that a device's design can have over defining user behaviour, and also that it may be very difficult to encourage end users to interact with a device as *intended* rather than as *designed*. If a system supports an end user's needs only marginally, or is designed to support multiple different tasks, this research suggests that training users to operate the

features specific to different tasks effectively will be very challenging, if not ineffective. This is an important consideration for systems in use in safety-critical situations.

The need for a device's design to match its intended uses is reminiscent of Norman's (1983b, 1988) argument for a correspondence between a system's conceptual model and the user's mental model. The results from this study emphasize that manipulating users' mental models through training to correspond with the conceptual model does not result in improved performance, emphasizing the need for the device design to be based on specific user needs and intentions from the very beginning.

The substantially different behaviour observed in the tester-enhanced condition versus the tester condition does show that a difference in focus (and corresponding behaviour) can be "unlocked" with only minor adjustments to the training; however, identifying the critical information to provide is the challenge.

10 Summary and conclusions

The work reported in this thesis contributes to our knowledge of the device initialization error by replicating the general results reported by Li (2006) and demonstrating the robustness of the error in his task environment. It also establishes the possibility that the error can be mediated without the need to make physical changes to the device interface, through the emphasis of certain types of information during training that contributes to an increase in the cognitive salience of the corresponding procedural step; modifying the task-based goal shows promise as an effective strategy for reducing the device initialization error rate, and more generally for changing the pattern of omission rates across different categories of steps. This result may be taken into consideration when similar systematic errors are identified elsewhere, as it indicates that there are (at least temporary) alternatives to immediately redesigning the device interface and task flow. In particular, carefully designed training programs may be of value.

This research also establishes that the device initialization error likely does not stem from a poor understanding of the role the corresponding procedural step plays in operation of the device. Providing detailed information about the system's internal mechanisms in the form of a device model does not appear to be an effective strategy for improving performance on the device initialization step, despite evidence of a change in participants' underlying mental representations. The general lack of influence of the device model on behaviour extends existing work on the role of mental models in performance,

indicating that they do not support users in accurately remembering the correct operating sequence for procedural tasks.

This work also provides several pointers to areas worthy of further investigation. Difficulties in encouraging participants to adopt a goal that is different from the one the device was initially designed to support indicates the need to identify the particular characteristics of the task environment that draw participants to one goal over another. In addition, the informal post-study interview conducted at the end of each experiment provided invaluable insights into additional factors that may have contributed to the reduction in the device initialization error rate observed for tester-enhanced participants, such as changes to the task difficulty and the removal of math-related steps. In order to fully understand the observed reduction in error rate, it will be important to explore these potential contributions further.

In summary, the results from this research extend our current understanding of omission errors during procedural tasks, particularly device initialization errors, as well as factors that can influence their occurrence. This research provides insight into practical ways of managing such errors, suggesting that training can have a positive effect on systematically omitted steps, and provides a starting point to guide further investigations.

References

- Altmann, E.M., & Trafton, J.G. (2002). Memory for goals: an activation-based model. *Cognitive Science*, 26, 39-83.
- Ashcroft, M.H. (2002). Math anxiety: Personal, educational, and cognitive consequences. *Current Directions in Psychological Science*, 11(5), 181-185.
- Ashcroft, M.H., and Kirk, E.P. (2001) The relationships among working memory, math anxiety, and performance. *Journal of Experimental Psychology*, 130(2), 224-237.
- Back, J. Cheng, W.L., Dann, R. Curzon, P. and Blandford, A. (2006). Does being motivated to avoid procedural errors influence their systematicity? *In Proc. HCI 2006*, 1.
- Blandford, A., Back, J., Curzon, P., Li, S.Y.W., & Rukeenas, R. (2006). Reasoning about human error by modeling cognition and interaction. *Proc. Resilience Engineering Symposium, Juan les Pins, France*.
- Burns, B.D. & Vollmeyer, R. (2002) Goal specificity effects on hypothesis testing in problem solving. *The Quarterly journal of Experimental Psychology*, 55A(1), 241-261.
- Byrne, M. D. & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science*, 21, 31-61
- Canas, J.J., Bajo, M.T., and Gonzalvo, P. (1994). Mental models and computer programming. *International Journal of Human-Computer Studies*, 40, 795-811.
- Chung, P. H. & Byrne, M. D. (2004). Visual cues to reduce errors in a routine procedural task. *In Proceedings of the 26th Annual Conference of the Cognitive Science Society*.
- 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 behavior. *Cognitive Science*, 24(2), 205-248
- Gray, W. D. (2004). Errors in interactive behavior. In W. S. Bainbridge (Ed.), *Encyclopedia of Human-Computer Interaction*, 230-235: Berkshire Publishing Group.
- Halasz, F.G., and Moran, T.P. (1983). Mental models and problem solving in using a calculator. In *Proceedings of CHI'83 Human Factors in Computing Systems*. New York: ACM.
- Hutchins, E. (1995). *Cognition in the wild*. Cambridge: MIT Press; 1995.
- Kieras, D.E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science*, 8, 255-273.

- Le Bot, P. (2004). Human reliability data, human error and accident models—illustration through the Three Mile Island accident analysis. *Reliability Engineering and System Safety*, 83, 153–167
- Li, S.Y.W. (2006). An Empirical Investigation of Post-Completion Error: A Cognitive Perspective. PhD Dissertation.
- Li, S.Y.W., Blandford, A., Cairns, P., Young, R.M. Post-completion errors in problem solving: A study. *In Proc. Cognitive Science Conference 2005*, 1278-1283.
- Li, S.Y.W., Cox, A.L., Blandford, A., Cairns, P., Young, R.M., Abeles, A. (2006). Further investigations into post-completion error: the effects of interruption position and duration. *Proc. Cognitive Science Conference 2006*.
- Logie, R.H., Gilhooly, K.J., Wynn, V. Counting on working memory in arithmetic problem solving. *Memory & Cognition*, 22(4) 395-410.
- Norman, D.A. (1981). Categorization of action slips. *Psychological Review*, 88(1), 1-15.
- Norman, D.A. (1983a). Design rules based on analyses of human error. *Communications of the ACM*, 26(4), 254-258.
- Norman, D.A. (1983b). Some observations on Mental Models. In A.L Stevens & D. Gentner (Eds.) *Mental Models*. New Jersey: Lawrence Erlbaum Associates, Inc.
- Norman, D.A. (1988). *The Design of Everyday Things*. New York: Basic Books.
- Rasmussen, J. (1980). What can be learned from human error reports? In K. Duncan, M. Gruneberg & D. Wallis (Eds.) *Changes in Working Life*. London: Wiley.
- Rasmussen, J. (1982). A taxonomy for describing human malfunction in industrial installations. *Journal of Occupational Accidents*, 4, 311-333.
- Reason, J. (1990). *Human Error*. Cambridge, UK: Cambridge University Press.
- Reason, J. (2002). Combating omission errors through task analysis and good reminders. *Qual Saf Health Care*, 11, 40-44.
- Richardson, F.C. & Suinn, R.M. (1972). The mathematics anxiety rating scale. *Journal of Counseling Psychology*, 19, 551-554.

Appendix 1

The low device relevance goal description presented to participants in the control and device model experimental conditions

In this session you will be acting as a Baker for the *Wicket Doughnut Company*, a large doughnut producer in the UK.

The company has just purchased the newest version of the *Wicket Doughnut Making Machine* for its bakeries, which is made up of five components used to bake batches of doughnuts: the Dough Depositor, Puncher, Froster, Sprinkler, and Fryer.

Older versions of the machine made it difficult for bakers to produce the correct quantity of doughnuts. Sometimes too many doughnuts would be made, leading to waste; other times too few doughnuts would be made, leading to unhappy customers. The new machine has been specifically designed to help bakers fill incoming orders more accurately.

As a Baker your main goal is to bake doughnuts exactly as they are ordered by your customers – by making the correct number of each type of doughnut ordered, you reduce waste and keep your customers happy. You will receive training on how to use the machine step-by-step, and then will use the machine on your own to fill orders for customers.

Appendix 2

The high device relevance goal description presented to participants in the tester experimental conditions

In this session you will act as a Test Engineer for the *Wicket Doughnut Company*, a large doughnut producer in the UK.

The company has just purchased the newest version of the *Wicket Doughnut Making Machine* for its bakeries, which contains five main components used to bake batches of doughnuts: the Dough Depositor, Puncher, Froster, Sprinkler, and Fryer.

The new machine has been designed to present a visual response each time a *Wicket* baker selects and activates a different component, so that she knows the new selection has been applied correctly. For example, when the baker is finished with the Dough Depositor component she will select the Puncher component, and the machine will present a visual response indicating that the Puncher has been activated.

However, since its installation the *Wicket* bakers have reported two types of problems with the new machine:

1. After making a selection, sometimes the machine fails to provide visual feedback about what has been selected
2. After making a selection, sometimes the machine provides feedback indicating that the **wrong** component has been activated

As a Test Engineer, you have been called in to test the machine and fill out a report on your findings for your boss. Since the problems only show up intermittently and are difficult to reproduce, you have been instructed to test the machine during normal operation (i.e., while using it to create doughnuts). You will receive training on how to test the machine step-by-step, and then will use the machine on your own to conduct a series of tests.

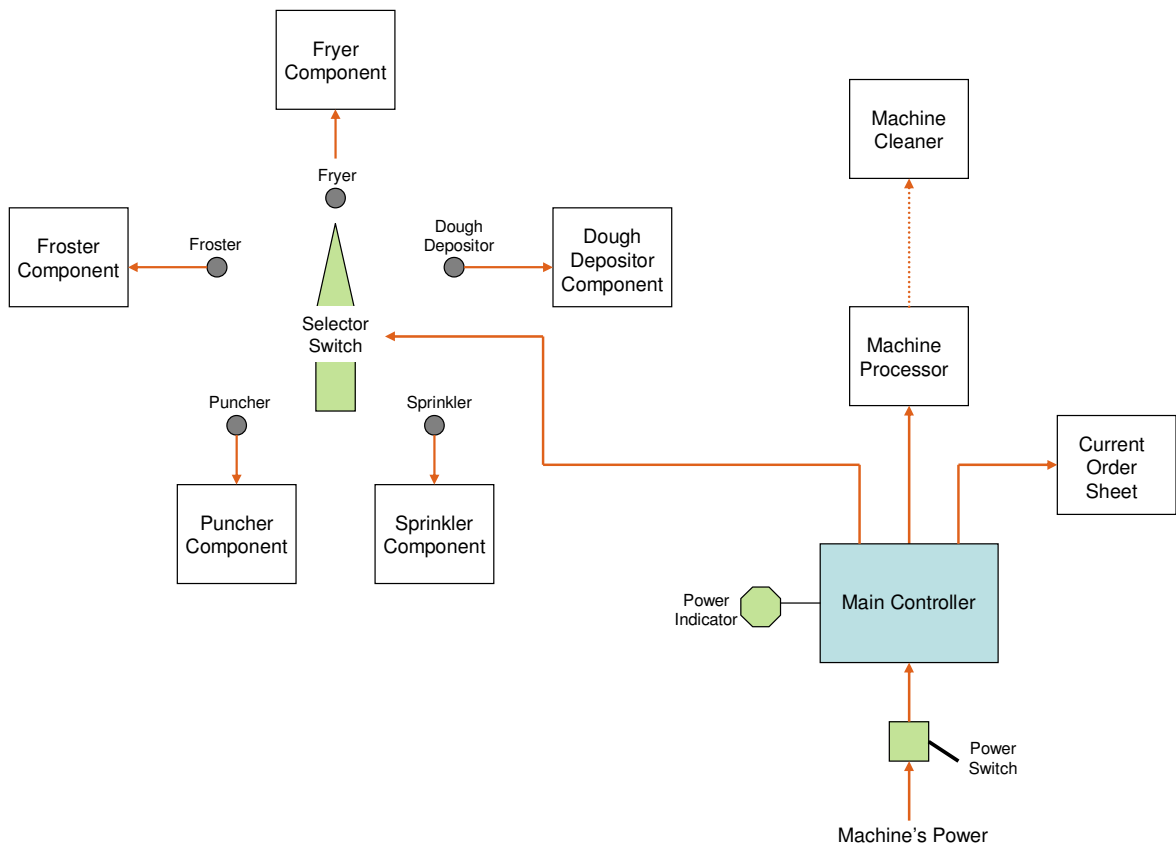
Your main goal is to **carefully observe the machine's visual responses (or lack thereof) to your input**, so you can fill out an **accurate report** for your boss about the machine's operation.

Appendix 3

The device model description presented to participants in the device model experimental condition

The back-end machines that are controlled by the new *Wicket* doughnut interface require a large amount of power in order to operate. In order to save money on operating costs and reduce their carbon footprint, *Wicket* has configured the system so that power can be distributed to the different machine components only when necessary. Due to recent budget restrictions, it is also important that you **operate the machine in an energy-efficient manner**.

The device diagram and an explanation of the power flow are provided below. Study these carefully for the next few minutes. We will give you a brief multiple-choice quiz before beginning the training to ensure you are familiar with these important concepts:



The machine's *Power Switch* activates the main power source. When the power is switched on and the *Main Controller* has warmed up, the *Power Indicator* light will turn green.

Power always flows to the *Current Order Sheet* display and to the *Machine Processor*, as these do not draw a lot of energy. The remaining power coming into the main controller is allocated via the *Selector Switch*; the machine component that has most recently been

selected will draw power from the main controller, and the remaining components will not receive any power (and therefore will be inoperable).

The *Machine Cleaner* is typically only run after a batch of doughnuts has been processed, so the *Machine Processor* has been designed to automatically control power flow to the cleaner – after the processor has been run, power will automatically begin flowing to the cleaner.

Appendix 4 (i)

The pre-training quiz provided to participants in the control experimental group

1. Which 5 of the following 8 baking components are included in the Wicket Doughnut Making Machine?

- | | |
|---|--|
| <input type="checkbox"/> Filling injector | <input type="checkbox"/> Dough Depositor |
| <input type="checkbox"/> Froster | <input type="checkbox"/> Former |
| <input type="checkbox"/> Puncher | <input type="checkbox"/> Fryer |
| <input type="checkbox"/> Kneader | <input type="checkbox"/> Sprinkler |

Appendix 4 (ii)

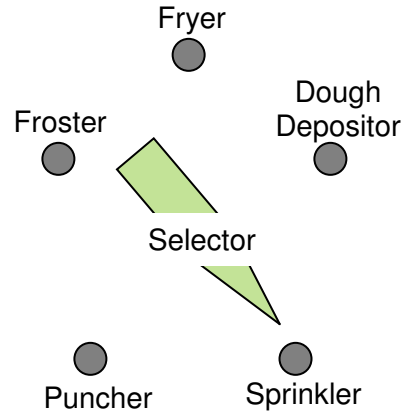
The pre-training quiz provided to participants in the tester and tester-enhanced experimental groups

Fill in the diagrams below to complete the example of each type of problem with the new doughnut machine, as reported by *Wicket* bakers.

Problem 1

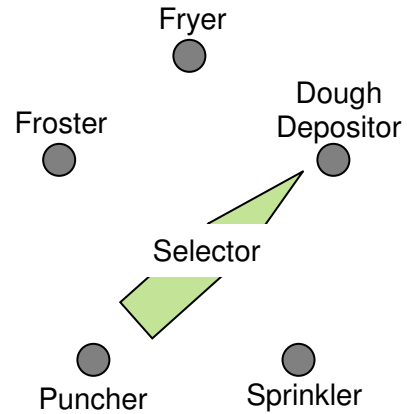
1. Machine's initial state:

The machine's selector is currently set to "Sprinkler"



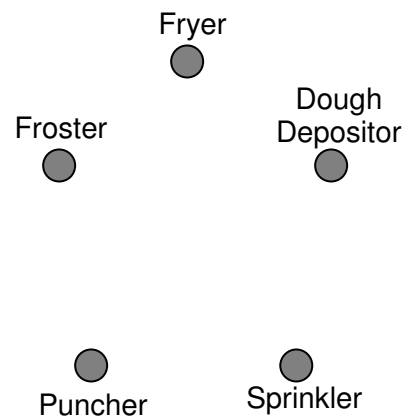
2. User action:

The baker selects "Dough Depositor"



3. Machine's response:

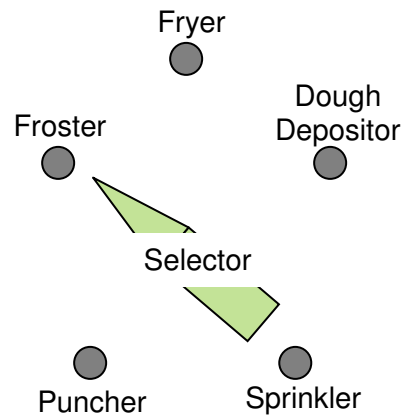
(draw the selector's position and any relevant feedback on the diagram to the right, and describe the machine's response in a few words below):



Problem 2

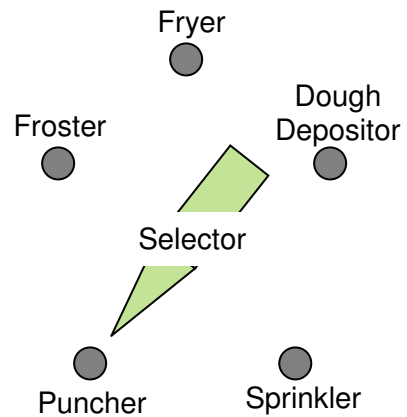
1. Machine's initial state:

The machine's selector is currently set to "Froster"



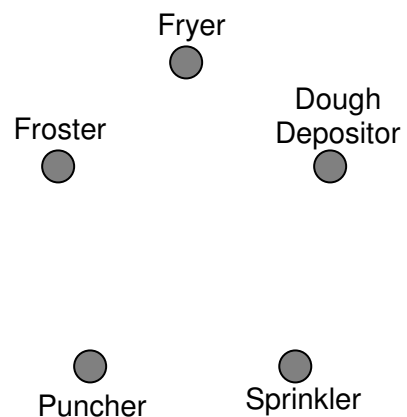
2. User action:

The baker selects "Puncher"



3. Machine's response:

(draw the selector's position and any relevant feedback on the diagram to the right, and describe the machine's response in a few words below):



Appendix 4 (iii)

The pre-training quiz provided to participants in the device model experimental group

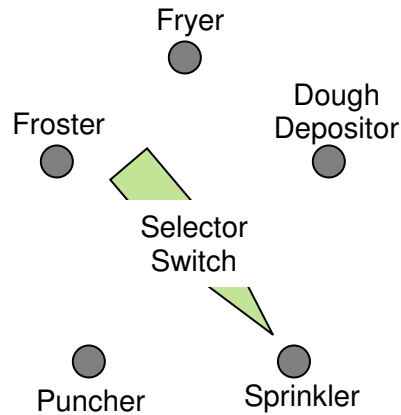
2. Which 5 of the following 8 baking components are included in the Wicket Doughnut Making Machine?
- | | |
|---|--|
| <input type="checkbox"/> Filling injector | <input type="checkbox"/> Dough Depositor |
| <input type="checkbox"/> Froster | <input type="checkbox"/> Former |
| <input type="checkbox"/> Puncher | <input type="checkbox"/> Fryer |
| <input type="checkbox"/> Kneader | <input type="checkbox"/> Sprinkler |

Please circle the most appropriate answer for each of the following questions (only circle one answer per question).

3. Which part of the machine activates the main power source?
- The *Power Switch*
 - The *Power Indicator*
 - The *Selector Switch*
 - The *Machine Processor*
4. Which parts of the machine always receive power flow, because they do not draw a lot of energy?
- The *Current Order Sheet* and the *Machine Processor*
 - The *Machine Cleaner* and the *Fryer* component
 - The *Dough Depositor* component and the *Fryer* component
5. What is the purpose of the *Selector Switch*?
- To indicate when the *Main Controller* has warmed up
 - To control power flow to the *Machine Cleaner*
 - To allocate power from the main controller to the *Fryer*, *Dough Depositor*, *Sprinkler*, *Puncher*, and *Froster* machine components
6. How can you activate power to the *Dough Depositor* component?
- The *Dough Depositor* component always receives power flow, it does not need to be activated
 - Set the *Selector Switch* to the *Fryer* component
 - Set the *Selector Switch* to the *Dough Depositor* component
 - Power to the *Dough Depositor* component is automatically activated after the *Machine Processor* has been run
7. How can you activate power to the *Machine Cleaner*?
- The *Machine Cleaner* always receives power flow, it does not need to be activated

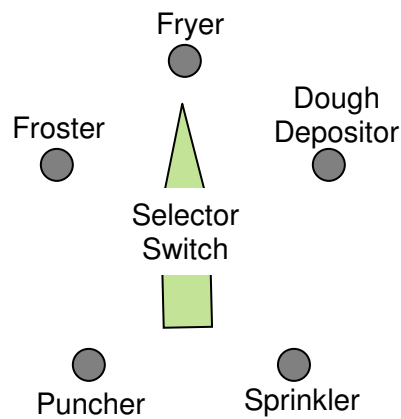
- B. Set the *Selector Switch* to the *Machine Cleaner* component
- C. Power to the *Machine Cleaner* is automatically activated after the *Machine Processor* has been run

8. Place a check-mark next to all machine components that will receive power if the *Selector Switch* is in the following position:



- Fryer component
- Dough depositor component
- Sprinkler component
- Puncher component
- Froster component

9. Place a check-mark next to all machine components that will receive power if the *Selector Switch* is in the following position:



- Fryer component
- Dough depositor component
- Sprinkler component
- Puncher component
- Froster component

Appendix 5

The four questions asked in the post-study interview

1. Can you describe the process you followed to accomplish your main goal?
2. Were there any instructions during the training that really stood out to you or stuck with you during the experiment?
3. When you were executing the procedure, what did you consider most important (i.e., what aspects did you emphasize)?
4. Was there anything that concerned or frustrated you during the trials?