The Influence of Source and Cost of Information Access on Correct and Errorful Interactive Behavior

Wayne D. Gray & Wai-Tat Fu
Human Factors & Applied Cognition
George Mason University
Fairfax, VA 22030 USA
+1 703 993 1357
gray@gmu.edu

ABSTRACT

Routine interactive behavior reveals patterns of interaction among the cognitive, perceptual, and motor elements of embodied cognition and the task and artifact used to perform the task. Such interactions are difficult to study, in part, because they require collecting a large quantity of mostly correct behavior. The current study varied conditions that were designed to affect the ease and likelihood that information would be stored in-the-world versus in-the-head. The data are examined to determine how subtle differences in the source and cost of information access may lead to different patterns of correct and errorful behavior.

INTRODUCTION

Interactive behavior emerges out of the constraints and opportunities provided by the interaction of embodied cognition (Kieras & Meyer, 1997) with task goals and the artifact used to perform the task (the ETA, η, triad). The interactions among the components of the ETA triad that determine interactive behavior may be extremely subtle with small changes in costs leading to large shifts in performance. For example, changing information gathering from an eye movement to a mouse movement influenced the decision-making strategies adopted in a classic decision-making paradigm (Lohse & Johnson, 1996). When the cost of making a move in solving simple puzzles increased from one keystroke to several (O'Hara & Payne, 1998; O'Hara & Payne, 1999; Svendsen, 1991) the strategy used to solve the puzzles shifted from one in which search was "reactive and display-based" to one in which search was more plan-based. The subtlety of change in response to minor variations in interface design should not be underestimated. For example, by increasing the cost of information acquisition from a simple saccade to a head movement, Ballard (Ballard, Hayhoe, & Pelz, 1995) induced a shift from a memoryless strategy to one that required holding information in working memory.

In the work reported here, we were interested in how the requirement to access information *in-the-world* versus *in-the-head* would influence routine interactive behavior. Almost by definition, most routine interactive behaviors are successfully executed. Hence, our focus is not on outcome measures of success, but on process measures of performance. Two important sources of clues regarding process are patterns of information access and errors that are made, detected, and corrected during performance.

Unfortunately, errors in routine interactive behavior are relatively rare and collecting enough such errors to discover underlying patterns requires collecting a large quantity of correct interactive behavior. For example, Gray (in press) found only 96 keypress errors in a data set of 1,946 keypresses collected from 9 people as they programmed 56 shows on a simulated VCR.[1] For this reason, we collected massive amounts of data under a variety of conditions that were designed to vary the ease and likelihood that show information would be stored in-the-world versus in-the-head. The raw data were analyzed to yield three categories of information; patterns of information access during performance, types of erroneous goals attempted (*push errors*), and correct goals that were abandoned prematurely (*premature pops*). These categories were then interrogated to determine how subtle differences in information access may lead to different patterns of correct and errorful behavior.

The next section introduces the model and the approach on which the determination and classification of errors was based. We then present the methods and procedures used in the current study. The empirical results are discussed in three sections. The first provides an overview of performance, the second discusses the fit of the data to model, while the third presents error data. We conclude with a summary and discussion of how varying the cost of information access during routine performance influences correct as well as errorful behavior.

[1] Participants used a mouse to interact with the simulation. The actual VCR was operated by pressing and sliding various physical buttons. Hence, neither the simulated nor the actual VCR required key presses. Few task analysis methods analyze behavior down to the level of physical actions (see, e.g., the survey of task analysis methods reported by Kirwan & Ainsworth, 1992). Throughout this paper, our use of the terms "keypress" reflects the fact that by including mouse clicks (or button presses) in the analysis, the task analysis is at the "keystroke level." This usage of the term "keystroke level" follows the distinction made by Card, Moran, and Newell (1983).

CONSTRAINED INTERACTIVE BEHAVIOR IN UNDERCONSTRAINED INTERFACES

Task goals for programming a VCR include setting a program's day-of-week, start time, channel, and end time (see Figure 1). Unfortunately, programming an actual VCR entails mapping these simple task goals into a variety of device specific goals. The result is a task-to-device rule hierarchy such as is shown in Figure 2.

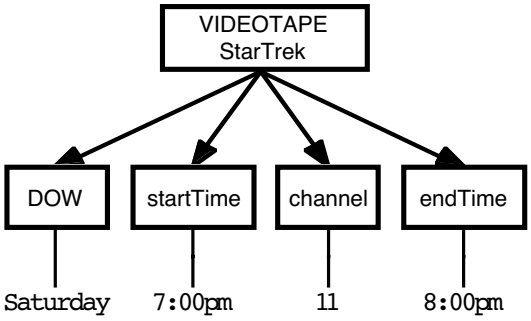

Figure 1: Task goals for programming a VCR.

The controversial part of what is being asserted is not that there is such a mapping, but that, in many cases, there is one least-effort mapping that, if left to themselves, most users will discover and use. If this least-effort mapping is taught, most users will conform to it despite a plethora of alternatives. The task-to-device rule hierarchy is encouraged, not mandated, by soft constraints derived from principles of cognitive least-effort (described in Gray, in press).

For many interactive devices, the sequence and methods of operation are highly constrained by design. For example, if your task goal is to take $100 out of your checking account using an ATM, you must find an ATM; insert your card; key in your pin number; press fast cash; take the money; and then take the card. For any one ATM, there is not much variability in the set of methods or their sequence.

In contrast, if you are programming the VCR simulated in our study, the device does not prevent you from clicking on the start mode button, setting the start hour, clicking on the end mode button, setting the end hour, clicking on start mode button (again), setting the channel to 10, setting the day of week to Saturday, going back and setting the channel to 11, clicking on the clock set mode button, clicking on PROG REC, clicking on end mode (again), setting the 10min, setting the min, clicking on start mode (yet again), setting the 10min, setting the min, and finally, clicking on the clock set mode button (again).

Although somebody could program the VCR in this way, in fact, nobody does. In the study reported by Gray (in press), out of 9 participants who were not taught how to program the VCR, but discovered the methods by themselves, seven adopted the task-to-device rule hierarchy of Figure 2 and two adopted minor variants. In the studies reported below, of the 72 participants shown Figure 2 as the experimenter programmed the first show, all but two used the task-to-device rule hierarchy to program the next four shows. Although extreme variation was possible, little variation was found.

The task-to-device rule hierarchy shown in Figure 2 was derived (Gray, in press) from three sources. The first was a simple task analysis of the methods available for programming shows on the simulated VCR. The second was an analysis of participant behavior during the instructionless learning phase of the study. The third was the analyses of the unsuccessful trials – those that ended without the VCR being successfully programmed. The resultant task-to-device rule hierarchy was used to analyze the 56 trials which were successfully programmed. By definition, any errors made on these *okay* trials were detected and corrected by the participants before telling the experimenter that they were done programming the VCR.

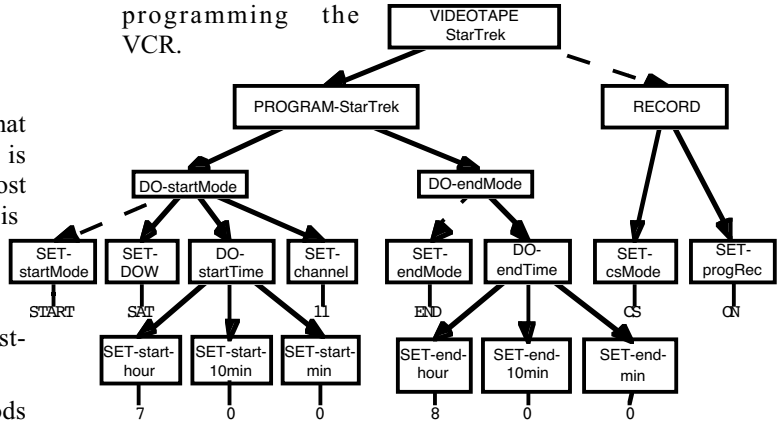

Figure 2: A mapping of the task goals from Figure 1 onto the device. This task-to-device rule hierarchy is largely determined by soft constraints. (Subgoals are represented by boxed nodes. Leaf nodes are unboxed and may represent multiple keystrokes. The dashed line leading from DO-startMode and DO-endMode indicate that subgoals SET-startMode and SET-endMode must be performed before the others. Contrariwise, the dashed line from VIDEOTAPE to RECORD indicates that RECORD must be performed last. With those three exceptions, the subgoals of a goal may be performed in any order.)

EXPERIMENT

The current study used a new simulation of the VCR task adopted by Gray (in press). One of our goals for the current study was to determine whether new groups of participants in slightly different task conditions would conform to the task-to-device rule hierarchy shown in Figure 2. Another goal was to verify and extend the error taxonomy.

Although these goals are important, they are not the main goals of this paper. Rather, our main goal is to explore how correct as well as errorful interactive behavior is

affected by changing the cost of information access. For the *control* group, the show's start time, end time, day-of-week, and channel were clearly visible to participants.

The *gray-box* condition was designed to increase the effort required to obtain show information. For the control condition, information access required an eye movement to the show information window. In contrast, for the gray-box condition, the labels in the show information window were visible but the fields were covered by gray boxes. For example, to see the channel field, the participant had to move the cursor to and click on the gray box covering that field. The value stayed visible as long as the cursor remained in the field.

The *memory-test* condition encouraged the storage of show information in long-term memory. For each trial, clicking on the START button removed the show information window and opened a memory test window. The memory test required the participant to select the show's start and end hour, 10min, min, as well as day-of-week and channel from a series of pop-up menus. Prior to programming the show, the participant iterated between the show information window and the memory test until the test was passed.

When the VCR was being programmed, we encouraged the memory-test condition to retrieve show information from memory by discouraging the use of the show information window. As per the gray-box condition, the fields of the information window were covered by gray boxes. In addition, moving the cursor out of the VCR window caused the VCR to be covered by a black box. The black box stayed until the participant moved the cursor back to and clicked on the VCR window. Hence, for the memory-test condition, when a participant moved to and clicked on a gray box, the corresponding setting of the VCR (indeed, all settings of the VCR) was covered by the black box.

Method

The experiment used VCR 2.0, a simulation of a commercial VCR built in Macintosh Common Lisp. All keypresses on any button object in VCR 2.0 were time stamped to the nearest tick (16.667 msec) and saved to a log file along with a complete record of the information in the VCR's displays (e.g., mode, time, day-of-week, channel, and so on).

Participants

Sixty-four George Mason University undergraduates participated in the experiment for course credit. Participants were randomly assigned to conditions and were run individually. Each session took approximately 30 min.

Procedure

The study began with the task-to-device rule hierarchy (Figure 2) in front of the participant. The experimenter programmed the first trial of show-0. As the show was programmed, the experimenter pointed to the figure, relating each step of programming to a node in the figure. After the first trial, the experimenter watched as the participant programmed show-0 to criterion. At that point, the experimenter left the room while the participant programmed shows 1 through 4 to the criterion of two successive correct trials. (As show-0 was an instruction and practice show, it is excluded from the analyses reported below.)

For all conditions each trial began with the VCR covered by a black box and a clearly visible information window that contained the current show's name, start time, end time, day of week, and channel. This information could be freely studied before the trial began. The information window also contained the START button. Clicking on the START button began the trial, changed the label from START to STOP, and either removed the black box that had covered the VCR (for control and gray-box) or opened the memory test window (for the memory-test condition).

At the end of each trial, the participant was given feedback as to how long the trial took and as to whether the show had been programmed correctly. If the show was not programmed correctly, the participant was provided feedback on the first error that the software found. The order in which errors were checked was: clock time, start time, end time, day of week, channel, and program record.

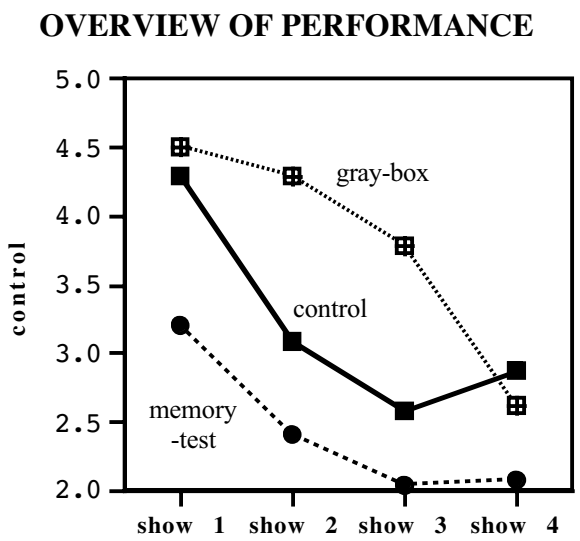

Figure 3: Mean trials to criterion.

Trials-to-criterion

A two-way analysis of variance (ANOVA) was conducted on the number of trials to reach the criterion of two successive correct shows. Condition (control, gray-box,

memory-test) was a between-subjects factor and show (1-4) was within-subjects. The main effect of condition was significant, F (2, 69) = 4.478, p = .015 (MSE = 10.035), as was the main effect of show, F (3, 207) = 5.896, p = .0007 (MSE = 5.053). The interaction of condition by show was not significant (F < 1) (see Figure 3).

Planned comparisons by condition yielded a significant difference between gray-box and memory-test (p = .0002) as well as between control and memory-test (p = .0370). The difference between the control and gray-box condition was not significant.

Checking the Show Information Window

In all conditions, participants were free to study the show information before each trial began. During the trial we expected the greatest reliance on the show information window for the control condition, less reliance for the gray-box condition, and the least reliance for the memory-test condition. Unfortunately, as we did not collect eye movement data, any discussion of what the control group did is speculation. However, we do have data that supports our interpretation of the tradeoff between information in-the-head versus in-the-world for the other two conditions.

For trials that were successfully programmed (for which all errors were detected and corrected before the participant clicked on the STOP button), the gray-box condition checked information 293 times (a mean of 1.31 checks per show). In contrast, participants in the memory-test condition checked an information field 10 times (0.05 checks per participant per show). This contrast suggests that the memory-test group almost exclusively relied on memory as their source of show information.

For the gray-box condition, 149 of the information checks were made immediately prior to the use of the information (e.g., checking the day of week field and then setting day of week). In contrast, only 33 checks were made on an information field immediately after the corresponding VCR display was set.

These patterns of checking suggest that the gray-box participants did not memorize show information to the degree forced on the memory-test condition. However, the low number of information checks per show (a mean of 1.31 fields checked per show) suggests that the perceptual-motor strategy was the backup strategy, not the primary strategy for this group. Furthermore, the 149:33 (or 4.5 to 1) disparity between information acquisition checks versus information verification checks suggests a trust in working memory that the trials-to-criterion data indicates was not justified.

These data are consistent with the notion that the cognitive system minimizes local effort, not necessarily total effort (see also Gray, in press). For the gray-box condition, the failure to verify saved several seconds worth of effort during a good trial, but may have resulted in more trials ending in error and, when compared to the memory-test condition, more trials needed to reach criterion. A similar conclusion is suggested by some of the error data that we review below.

FIT OF DATA TO MODEL

A goal and subgoal analysis was conducted on trials that ended successfully. This restriction meant that any errors made during the trial had to be detected and corrected before the participant pressed the STOP button.

For these analyses, ACT-PRO (Fu & Gray, 1999) was used to parse the log file into goals, subgoals, and operators. Each deviation from the task-to-device rule hierarchy shown in Figure 2 was noted and classified by ACT-PRO. (The classification categories used here are an expansion of those reported by Gray, in press).

Over the course of the study 36,877 keypresses were collected. ACT-PRO parsed these into 12,704 goals and subgoals. Of this number, 98.4% (12,560) are goals that are captured by the task-to-device rule hierarchy.

Of the uncaptured goals and subgoals, 56 can be readily interpreted as the participant returning to a mode to double-check a setting. These additions increase the percentage of goals and subgoals accounted for to 98.8%.

The remaining 148 goals can be examined to determine if they represent errors or are simply alternatives to the task-to-device rule hierarchy used by the model. Of these potential errors, 16 represented alternative ways of correctly programming the VCR. These alternatives were manifested by five participants. Only two of these five participants used the alternative on a majority of trials. Hence, although there may be hundreds of ways of segmenting and sequencing the task of programming this VCR, the model shown in Figure 2 accounts for the vast majority of correct behavior shown by the overwhelming majority of participants.

ACCOUNTING FOR ERROR

The taxonomy developed by Gray (in press) relied on model-tracing (Anderson, 1993) to identify deviations from the task-to-device rule hierarchy as *push errors* or *pop errors*. Any key that is pressed at a time or place where the model would not press it is a push error. Any goal or subgoal that is abandoned, or popped, before the model would end it is a pop error.

Push Errors

As discussed above, ACT-PRO classified 148 goal pushes as violations of the model's task-to-device rule hierarchy. After we subtract those behaviors that can be interpreted as alternative rule-hierarchies we are left with a data set of 132 push errors. In this paper, space constraints force us to limit our discussion to the 31 erroneous attempts to

increment rather than decrement (or vice versa) the channel setting.

Except for channel, each of the other settings had only one button. For day-of-week, hour, 10min, or min this button would only increment, never decrement the setting. In contrast, channel had two buttons; one to increase the displayed setting and one to decrease it. Hence, whereas if an erroneous attempt to decrement the day-of-week, hour, 10min, or min, was detected and corrected by the participant, it would have gone unnoted by the experimenter. In contrast, any goal to decrement the displayed channel setting when it should have been incremented (or vice versa) would be obvious from the log file. (Note that the target channel setting was higher than the default setting for two shows and lower than the default for the other two shows.)

An ANOVA of errors by conditions for incrementing versus decrementing the channel revealed a marginally significant effect, F (2, 69) = 2.787, p = .069, MSE = .683. The mean per trial error rate was higher for memory-test (0.750) than for gray-box (0.333) and lowest for control (0.208). Planned comparisons showed that the difference between memory-test and control was significant (p = .027) while the difference between memory-test and gray-box was marginally significant (p = .087).

While programming, participants in the memory-test condition checked show information a total of 10 times. The reliance on information in-the-head versus in-the-world resulted in an increase in errors. However, the information was well-learned and participants soon retrieved the correct information and set the channel to the correct setting. The transient nature of this error suggests a momentary fluctuation in strength of the memory trace due to noise (Altmann & Gray, 1999; Anderson & Lebiére, 1998).

Pop Errors

By the analysis introduced by Gray (in press), not only can pushing a goal be an error, but popping can be errorful as well. Popping a goal before its target setting has been reached is a *premature pop*. The data set collected by Gray (in press) was too small to distinguish between various types of premature pops. However, the 182 premature pops collected in the current study is an order of magnitude larger than that previously obtained. This set permits us to distinguish between three types of premature pops.

Local premature pops (pp-local) entail beginning to program a VCR setting but stopping before the target setting is achieved. For example, if the target day-of-week is Saturday and the current day-of-week is Tuesday, pressing the DOW key twice and then going off and doing something else would be classified as a pp-local. Time premature pops (pp-time) entail completing one or two of the DO-startTime or DO-endTime subgoals (see Figure 2) but abandoning the goal before the remaining subgoals are completed. Similarly, mode premature pops (pp-mode) entail popping the DO-startMode or Do-endMode goal before all of their subgoals are completed.

Across the three types of premature pops a repeated measures ANOVA showed no main effect of condition (F < 1), a significant effect for type of premature pop [F (2, 138) = 12.868, p < .0001, MSE = .041] as well as a significant interaction of condition by type [F (4, 138) = 2.989, p = .021]. As Figure 4 shows, the gray-box condition made the most pp-local errors with the memory-test condition making the least. This pattern was reversed for pp-mode errors.

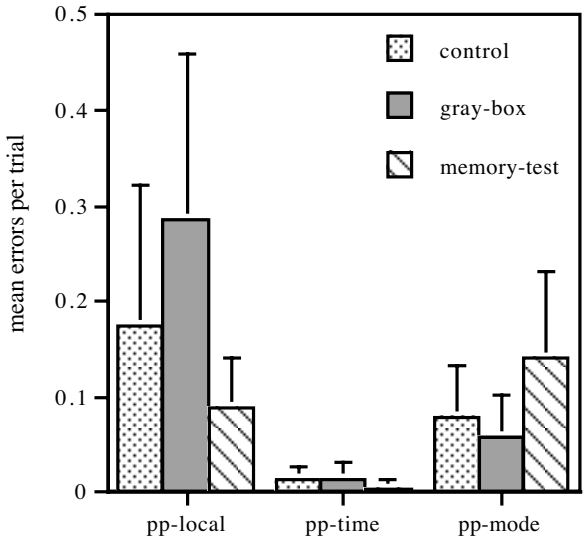

Figure 4: Premature pops by condition. (Error bars show the 95% confidence interval of the SEM.)

The higher pp-local error rate for gray-box is consistent with their pattern of fewer checks to verify show information. These errors – which were caught – as well as the errors that were not caught (i.e., those that led to greater trials-to-criterion for this group) may have resulted from the low rate of verification shown by this group.

Likewise, the higher rate of mode errors for memory-test may be the result of their reliance on memory. Although the gray boxes covered up the values of the information fields, they did not cover the labels for those fields. Hence, the labels may have served as a type of goal posting. The control and gray-box conditions would have been reminded of the goals for the current mode every time they glanced at the show information window.

SUMMARY AND CONCLUSIONS

The most striking aspect of the between group differences in errors and performance is that all were avoidable. All performance differences can be traced to differences in willingness to either memorize or visually access show information. For each trial, the memory-test group had

quick and reliable access to show information in memory. The other groups made more undiscovered errors that resulted in more trials-to-criterion. Apparently verification is lower cost – and hence more likely – if based on knowledge in-the-head rather than accessing knowledge in-the-world.

On trials for which any error made was eventually detected and corrected, we found an interaction between group and type of premature pop. The gray-box condition was more likely to abandon the current key (pp-local) before completing a setting, whereas the memory-test condition was more likely to switch modes before all subgoals were completed (pp-mode). The pattern for pp-local errors is consistent with that for trials-to-criterion. In both cases, errors were made because the gray-box group was unwilling to invest in the time and effort needed to obtain reliable information.

Our interpretation of pp-mode errors suggested an advantage to relying on information in-the-world rather than in-the-head. Both the control and gray-box conditions accessed the show information throughout performance. In addition to obtaining the value of the information fields, accessing the show information window may have served as a type of goal posting to remind participants what settings they had programmed and what remained to be done. In contrast, the memory-test condition would have had to keep a corresponding checklist in-the-head. Unlike the show information that they memorized, the state of this mental checklist was dynamic and changed throughout task performance.

We interpreted the push error that we analyzed as evidence for fluctuations in the strength of items encoded in long-term memory. The fact that the misretrieved settings were detected and corrected without recourse to the show information window is consistent with the ACT-R assumption of transient fluctuations in strength (Altmann & Gray, 1999; Anderson & Lebiére, 1998).

The study of routine interactive behavior is not itself routine. To study how small changes in artifact design affect performance, massive amounts of correct behavior must be collected. The analysis of routine interactive behavior enhances our understanding of how the cognitive, perceptual, and motor elements of embodied cognition interact with task and artifact to affect correct and errorful performance. This report suggests that small changes in the cost of information access may result in differences in the trials needed to reach criterion and the patterns of errors made.

ACKNOWLEDGEMENTS

The work reported was supported by a grant from the National Science Foundation (IRI-9618833) as well as by the Air Force Office of Scientific Research AFOSR#F49620-97-1-0353.

REFERENCES

Altmann, E. M., & Gray, W. D. (1999). Preparing to forget: Memory and functional decay in serial attention. *Manuscript submitted for publication*.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Lebiére, C. (Eds.). (1998). *Atomic components of thought*. Hillsdale, NJ: Erlbaum.

Ballard, D. H., Hayhoe, M. M., & Pelz, J. B. (1995). Memory representations in natural tasks. *Journal of Cognitive Neuroscience, 7*(1), 66-80.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Fu, W.-T., & Gray, W. D. (1999). ACT-PRO: Action protocol tracer -- a tool for analyzing simple, rule-based tasks. Proceedings of the *Sixth ACT-R Workshop* (pp.). Fairfax, VA: ARCH Lab.

Gray, W. D. (in press). The nature and processing of errors in interactive behavior. *Cognitive Science*.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12*(4), 391-438.

Kirwan, B., & Ainsworth, L. K. (Eds.). (1992). *A guide to task analysis*. Washington, DC: Taylor & Francis.

Lohse, G. L., & Johnson, E. J. (1996). A comparison of two process tracing methods for choice tasks. *Organizational Behavior and Human Decision Processes, 68*(1), 28-43.

O'Hara, K. P., & Payne, S. J. (1998). The effects of operator implementation cost on planfulness of problem solving and learning. *Cognitive Psychology, 35*, 34-70.

O'Hara, K. P., & Payne, S. J. (1999). Planning and the user interface: The effects of lockout time and error recovery cost. *International Journal of Human-Computer Studies, 50*(1), 41-59.

Svendsen, G. B. (1991). The influence of interface style on problem solving. *International Journal of Man-Machine Studies, 35*(3), 379-397.