In Proceedings of the Sixth International Conference on Cognitive Modeling, 52-57.
Mahwah, NJ: Lawrence Earlbaum.

Routine Procedural Isomorphs and Cognitive Control Structures

Michael D. Byrne, David Maurier, Chris S. Fick, Philip H. Chung
{byrne, dmaurier, cfick, pchung}@rice.edu
Department of Psychology
Rice University, MS-25
Houston, TX 77005

Abstract

A major domain of inquiry in human-computer interaction is the execution of routine procedures. We have collected extensive data on human execution of two procedures which are structurally isomorphic, but not visually isomorphic. Extant control approaches (e.g., GOMS) predicts they should have the same execution time and error rate profiles, which they do not. We present a series of ACT-R models which demonstrate that control of visual search is likely a key component in modeling similar domains.

Introduction

Every day, people execute countless procedures which are more or less routine. Many of these are uninteresting, but many of these occur in contexts such as emergency rooms and command-and-control centers where failures of speed or correctness can have serious consequences. Thus, understanding how humans execute routine procedures is critical in at least some domains. Card, Moran, and Newell (1983) and John and Kieras (1996) define a routine cognitive skill as one where the person executing the skill has the correct knowledge of how to perform the task and simply needs to execute that knowledge. Roughly speaking, that can be thought of as the point where people are no longer problem solving, but rather applying proceduralized knowledge to a relatively familiar task.

This level of skill has been the focus of attention for an entire family (the GOMS family; John & Kieras, 1996) of techniques for analysis and execution time prediction. This is largely due to the fact that such a wide array of situations fall under this classification, from occasional but not infrequent programming of VCRs to situations involving highly-motivated people in safety-critical situations, such as commercial pilots and medical professionals. As noted, GOMS, which stands for goals, operators, methods, and selection rules, is one of the primary techniques for predicting human performance under these conditions, and the empirical success of GOMS is well-documented (again, see John & Kieras, 1996). A typical GOMS analysis is based on a hierarchical goal decomposition and then a listing of the primitive operators needed to carry out the lowest-level goals. Thus, GOMS analyses are highly sensitive to the goal-based task structure and the number of primitive operations required.

What such an analysis predicts is that two tasks with the same goal/method/operator structure should produce identical performance. While this may be true in a great many cases, it is not universally true. We will present data from two tasks which would yield equivalent GOMS models (which we term "GOMS-isomorphic") but produce significantly different profiles in terms of the time of execution for each step, as well as the error rates at each step. This is not intended as a criticism of the GOMS modeling approach, but rather as the identification of an opportunity for improvement.

This presentation will focus on performance in a series of laboratory experiments in which participants were trained on a number of relatively simple computer-based tasks and then in a subsequent session, returned to perform those tasks along with a concurrent memory-loading task. This paradigm is essentially the same as that used in Byrne and Boviar (1997), which focused on a particular type of procedural error, the postcompletion error. This line of research is primarily concerned with errors made in the task, but to fully understand the errors made, we felt it would first be necessary to understand the cognitive control structures which would produce execution times similar to those we found in the lab. In order to understand these experiments, a relatively thorough understanding of the tasks is required.

The Tasks

Common Procedures

The two tasks under examination were both set in a fictional Star Trek setting to encourage engagement of the undergraduate participants. Participants came in for two sessions spaced roughly one week apart. The first session was training, in which participants were given a description for each task and a manual, walked through the task once with the manual in hand, and then had to repeat each task until they performed it without error three times. In the second session, participants performed the tasks on which they were trained in the first session, along with a concurrent memory-loading task. In this task, they had to monitor a stream of spoken letters which was occasionally interrupted with a beep, after which they responded with the last three letters heard. Participants earned points for correctly executed steps, lost points for errors, received bonus points for rapid performance, and lost points for incorrect answers to the memory probes. High scorers received additional compensation.

While participants were trained on several tasks, not all of which were the same from experiment to experiment, the current research is focused on two tasks, called the Phaser and the Transporter. These two tasks are isomorphic in that they have the same number of steps which were grouped in

the training manuals in the same subgoals. The names of those goals, and the names of the buttons and some of the displays and actual controls, however, were different between the two tasks.

The displays for the two tasks appear in Figures 1 and 2 and the list of subgoals and steps appears in Table 1. The main goal in the Phaser task is to destroy the hostile Romulan vessel; the main goal in the Transporter task is to energize it to return some crewmembers to safety. One of the immediately obvious visible differences between the two layouts is that the controls for the Transporter are visually grouped according to subgoal while in the Phaser they are not.

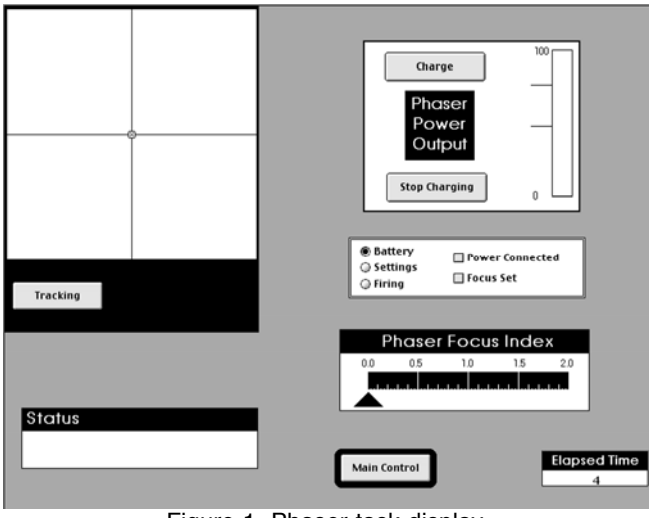

Figure 1. Phaser task display

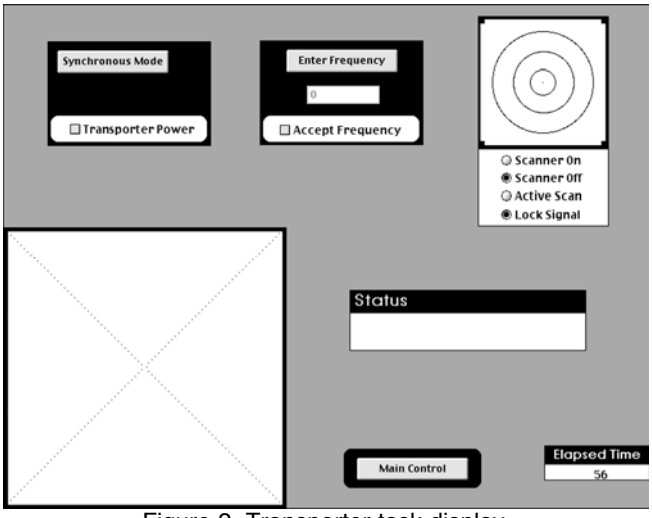

Figure 2. Transporter task display

There are some other important features to note as well. After Step 3 in both tasks, the participants had to wait until the display reached an acceptable state before clicking the next button. Step 6 in both tasks involved, or could involve, multiple actions: multiple drag adjustments to the slider in the case of the Phaser and multiple keystrokes in the case of the Transporter. Step 10 in both procedures involved a somewhat extended tracking task, done with the arrow keys for the Phaser and with the mouse for the Transporter. All of the other steps required the simple clicking of a button.

The exact responses of the display and some of the task structure did differ between the two tasks for Steps 11 and 12 as part of manipulations concerned with postcompletion errors, so those steps were excluded from all present analyses.

The major dependent variables of interest here were step completion time and error frequency Step completion time is measured as the time between clicks. That is, the time for Step 2 in the Phaser is the time elapsed between the click on “Power Connected” and the click on “Charge.” For the first step, the start time was the start of the trial. Steps on which errors were made were excluded from the time analysis. Times for steps that include other actions (waiting, tracking) were be excluded from the analysis because this other time is difficult to factor out.

Error frequency was also measured. This hinges on the definition of what counts as an error. Each step can be considered a sequential choice (Ohlsson, 1996), so the definition was based on the step, not the action. If any incorrect action was taken at a step, that step counted as an error, regardless of the number of incorrect actions taken. For example, if a participant is at Step 4 in the Phaser task, and they click on the “Settings” button and then the “Firing” button, only one error was recorded because an error was made at that step. Frequency was calculated as the number of error-containing steps divided by the total number of steps executed.

Step #	Phaser	Transporter
First subgoal		
1	Power Connected	Scanner On
2	Charge	Active Scan
3	Stop Charging	Lock Signal
4	Power Connected	Scanner Off
Second subgoal		
5	Settings	Enter Frequency
6	<slider>	<type>
7	Focus Set	Accept Frequency
Third subgoal		
8	Firing	Transporter Power
9	Tracking	Synchronous Mode
10	<tracking task>	<tracking task>
Fourth subgoal		
11	Tracking	Synchronous Mode
12	Main Control	Main Control

Table 1. Steps in the two task isomorphs

Because these tasks are essentially isomorphic, there is no *a priori* reason to necessarily expect different performance on the two tasks (through Step 10), except perhaps slightly longer step completion times for those steps where the mouse has further to go. Nor was assessing such differences the original purpose of the three experiments we will report; those experiments were primarily focused on postcompletion errors in the Phaser task.

Results

While three separate experiments were run, these experiments differed from each other in detail only. Experiment 1 actually included subsequent sessions with a variety of between-subjects manipulations; Experiment 2 added visual cueing at the postcompletion step of the Phaser; Experiment 3 used a cue and a mode indicator to attempt to mitigate postcompletion effects in the Phaser at step 11; the exact point system used in the three experiments differed slightly; etc. However, none of these surface dissimilarities made much difference; the results are nearly identical for all three experiments (inclusion of "experiment" as a between-subjects variable reveals no main effects or interactions involving that variable). Across the three experiments, data from a total of 164 participants were used. Figure 3 shows the results for step completion time for the Phaser and the Transporter, while Figure 4 presents the error frequency. Steps 3, 6, and 10 are excluded from Figure 3 because those steps involve other processes (e.g., tracking) or possibly multiple actions, as described above. Note that both graphs also include the 95% confidence intervals (non-pooled error).

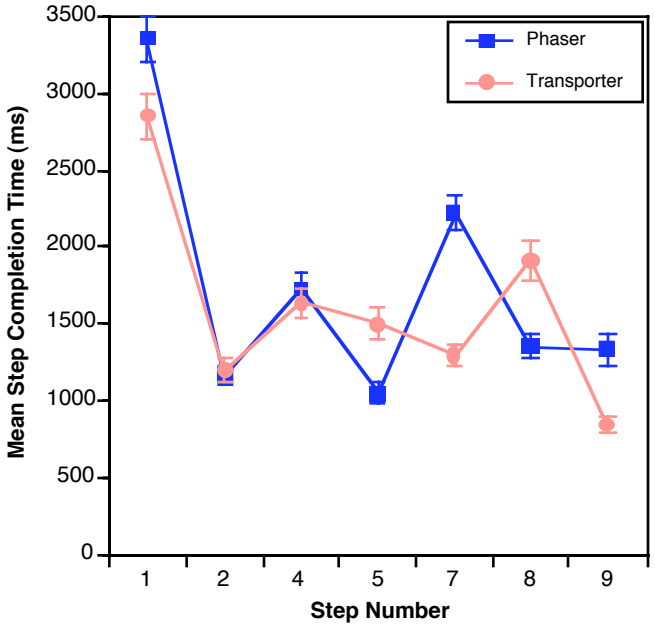

Figure 3. Mean step completion times by task and step

So, while the tasks are isomorphic in terms of subgoals and steps, they produce clearly different step completion time profiles. This runs clearly counter to any account which relies entirely on the GOMS-level structures.

Similarly, if one assumes that the same failure mechanisms are in operation in each task, the two tasks should produce identical error rate profiles as well. This is also obviously not the case. While both tasks share a spike in error rate at step 4 in the procedure (this is, in fact, a postcompletion error), the Phaser shows other spikes at steps 1 and 6 while the Transporter only shows another spike at step 8. Note that these spikes in error rate are not particularly linked to exceptionally large or small step times, either; for example, step 7 in the Phaser is particularly slow, but is not especially error-prone. Step 1 is slow in both tasks, but only markedly error-prone in the Phaser.

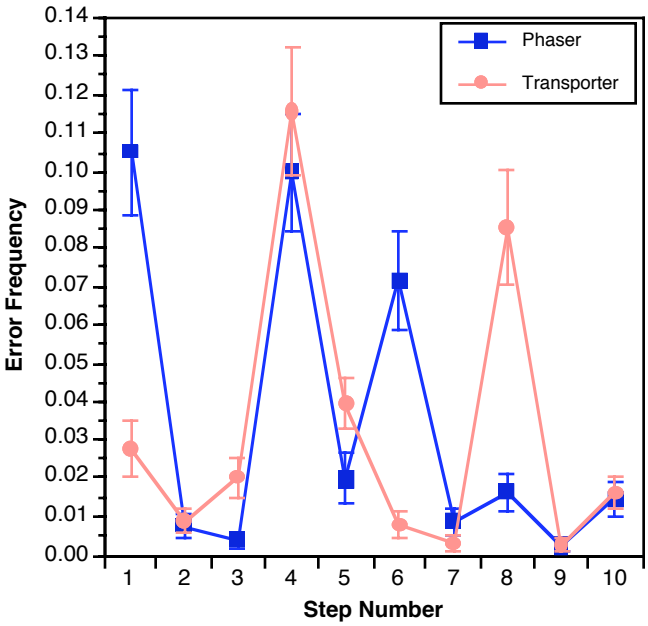

Figure 4. Mean error frequency by task and step

Discussion

These data are obviously problematic for any account which relies solely on the goal-subgoal-method structure for predicting execution time. It is hard to know how a GOMS-style account might accommodate these data. Subjects were probably not at the level of skill where extreme interleaving of cognitive, perceptual, and motor operations is required to model their peformance, thus it is not clear that the CPM variant of GOMS (e.g., Gray, John, & Atwood, 1993) would be appropriate. This is not to say that motor operators are unimportant; there are some differences in going from button to button in terms of pointing time as predicted by Fitts's law, but these differences are relatively small (as will be shown later).

One possibility that seems straightforward is that each of these buttons has to be visually located in order for the mouse to be moved to the button and a click registered. However, there is no single "visual search" operator in GOMS (or ACT-R or Soar for that matter) which would obviously capture the differences here. Each button on the display is at least approximately equal in terms of visual salience; while one might argue that the larger gray pushbuttons are more salient and should thus be found

faster, there is little difference between steps 8 and 9 of the Transporter, one of which is a large gray button and the other is simply a labeled checkbox. Furthermore, consider step 7 in the Phaser is markedly slower than step 7 in the Transporter and yet both are simple check boxes with two-word labels. So, if the difference is simply in a "visual search" operator, this operator must itself be driven by something substantially more sophisticated than what is present in a typical GOMS analysis. Furthermore, if the only difference between the two tasks is in their visual search latencies, the source of the differential error spikes remains a mystery.

This obviously raises the question of what kind of control structure could account for the differences between these two tasks? Accounting for the error profiles seems extremely difficult with any model at this point; generative theories of error are in their infancy at best (though that is ultimately our goal, see also Byrne, 2003). Thus, we entered into a modeling exploration with the modest goal of trying to understand what drove the step completion times.

Modeling

We constructed a number of models of this task using ACT-R 5.0 (Anderson, et al., in press). This was done not so much because of a strong commitment to any particular mechanism in ACT-R, but rather because ACT-R contains the full suite of perceptual, motor, and cognitive functionality required for these tasks. It is likely that some version of Soar or EPIC would have served equally well for present purposes but we are much more familiar with ACT-R (and further suspect we will need the subsymbolic mechanisms for future error modeling).

We constructed four models of each task. It was our hope that this way we might "bracket" performance (Kieras & Meyer, 2000; Gray & Boehm-Davis, 2000) and see if the models could provides reasonable predictive bounds. The four models represented a crossing of two dichotomies:

Goal organization. The first dichotomy was whether the model used a hierarchical representation of the goal structure, with intermediate subgoals (e.g., "charge the phaser") or a "flat" goal structure where 12 low-level goals were simply executed in sequence. There is reason to believe that even well-practiced experts do not entirely flatten their goal hierarchies (Kieras, Wood, & Meyer, 1997) and that, in fact, often times fairly slow retrieval-based strategies are appropriate (Altmann & Trafton, 2002). The hierarchical goal strategy is noted with "Hier" in the model label and the flat with "Flat."

Visual search. We implemented two very simplistic visual search strategies here: one in which the location of each button had to be determined through serial visual examination with a tendency to search near the current focus of visual attention (Fleetwood & Byrne, 2003) and one in which the model is assumed to have declarative knowledge of the locations of the buttons which must be retrieved for each button. Various ACT-R models (Ehret, 2002; Anderson, et al. in press) have shown that this kind of learning is a key component of skill development in similar interfaces. The unguided serial search strategy is noted with "DS" (for dumb search) and the alternate with "RL" for "retrieve location."

It should be noted that in ACT-R, these dichotomies may interact. ACT-R's visual system has a memory for which locations (though not explicitly which objects) have been viewed recently, but this memory decays over time (we used 1.5 seconds for this decay time; the models are indeed sensitive to this parameter but in unusual ways which are beyond the scope of this presentation). Thus, additional time spent in traversing the goal hierarchy can result in the loss of this information, which may affect the time course of the serial visual search.

ACT-R also embeds Fitts's law for prediction of mouse movement times. We used ACT-R to calculate the expected movement time between the various buttons to make clear the movement time contribution to the results. We did not compute it for step 1 because the initial location of the cursor was not recorded; informal observation of the participants indicated that many of them moved the mouse around before clicking anyway.

Finally, these models are all stochastic. Time for memory retrievals and perceptual-motor operations in ACT-R can be made noisy and ACT-R chooses randomly between options in various subsystems in cases of ties, so each run of the model is not identical to the last. We present the mean model-generated times for 100 runs of each model.

Model Results

Figure 5 presents the data and the model predictions, as well as the Fitts's Law time, for the Phaser task. Figure 6 presents the same for the Transporter.

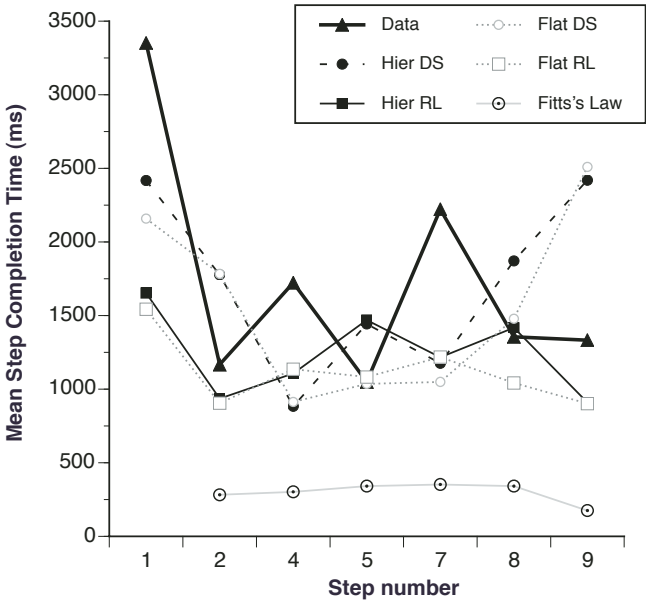

Figure 5. Model and data for the Phaser task

None of the four models provides a particularly good fit; which model is the "best" model by fit metric depends on which metric of fit one uses: by r-squared, the best model is Flat RL at 0.73; by RMSD, the best model is the Hier DS at 640 ms; by mean absolute deviation (MAD) the best model is the Hier RL model at 26%. These are fairly

fine distinctions since RMSD ranged from 640–739 ms and MAD ranged from 26–33%. Note that the model variant here which is most similar to a GOMS-style model is the Hier RL model. This model uses hierarchical goal decomposition as per GOMS, and essentially has a fixed time "find-on-screen" operator (the retrieval of the location). This model is generally good, if a bit too fast, for the Transporter, but is a poor model for the Phaser.

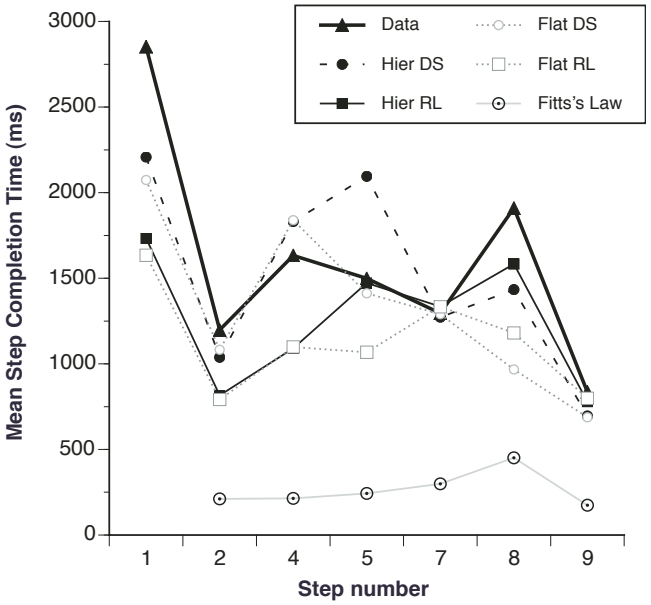

Figure 6. Model and data for the Transporter task

While the models do not provide outstanding levels of fit, they do provide some important insights. First, Fitts's Law alone provides an r-squared of 0.28 for those steps where it is applicable. Obviously time is grossly under-predicted by Fitts's Law—it is hardly surprising that more is going on here than simple motor movement, though it is clearly a contributor.

In general, the ordinal effects one would expect from the basic construction of the models held: the Flat models were faster than the Hier models and the DS models were generally faster than the RL models. Note that in general, the RL models' performance on the two isomorphs was quite similar. This is a reflection of their isomorphic task structure. The DS models, on the other hand, reflect differences between the tasks. This is consistent with the notion that it is the visual aspects of the display—the DS models are sensitive to button location with the RL models are only in the Fitts's law sense—that drives the differences between the two tasks.

However, there were a few cases where the DS and RL models were rougly equivalent, and even one case where the DS models were faster (Phaser step 4, Power Connected). There were some degenerately bad performances by the DS model, notably Phaser step 9, Tracking, and the Hier DS on Transporter step 5, Enter Frequency. Both of these cases involve a visual shift to a location which has a lot of competition from items much closer to the starting attentional focus.

On the other hand, the slower DS resulted in better fit in some instances, namely step 1 for both tasks, and steps 2 and 3 for the Transporter. Step 1 is particularly problematic for all four models; this is a very slow step for both the models and the participants, but more so for the participants. We suspect this is due to some kind of initial orienting or goal construction on the part of the participants which was not well-represented in the model, but may be partially represented by the DS behavior of taking an initial visual survey of the display. This plays into the next insight we gained from these models.

In general, the RL models were slightly better than the DS models. What this suggest to us is that participants in this case are at an intermediate point in their learning of the locations of the objects on the interface. Our next model will likely not start with the locations explicitly encoded in declarative memory but will instead use the strategy of attempting to retrieve them from memory, but this time from the memories created as a by-product of visual searches conducted along the way.

Comparisons between the Flat and Hier models are also revealing. These models differed primarily at steps where either they interacted with the visual search process (Transporter step 5, Enter Frequency is a good example of this) or there was a delay for additional goal traversal (steps 1, 5, and 8. This additional time appears correct for both tasks for steps 1 and 8, but step 5 indicates something else going on. Both Hier models are too slow for step 5 in the Phaser, but the Hier RL model is right on target for step 5 for the Transporter.

Finally, some points were fairly strategy-insensitive. Transporter step 7 (Accept Frequency) was fit equally well by all four models. This is an interesting case for two reasons: [1] the DS visual search strategy will almost always search the correct location first here because of visual proximity to where the model is looking prior to this step, [2] it is the last subgoal within the second goal, and thus not differentially affected by the goal organization, and [3] the completion time for the similar Phaser step is radically different. None of the models captured this deviant time in the Phaser at all.

Discussion

While it may appear that our goal was to somehow falsify or criticize GOMS models, but that was not our intent. Instead, we wanted to explore where and why models based purely on the GOMS-style structure would misfit, not for the purposes of finding fault, but to find opportunities for improvement. One of the primary things GOMS-style models lack is a consideration of the visual task faced by interface users. This was certainly reasonable when most users faced command-line tasks which were indeed primarily cognitive, but the shift to increasingly visual interfaces has raised the importance of systematically addressing the problem of how the visual and cognitive systems are integrated. While this has certainly been a big topic for some cognitive scientists for many years (see Pylyshyn, 1999 and the associated commentary for an excellent discussion), it has not been a prominent theme in computational modeling of human-machine interfaces until

fairly recently and in cases where the task is clearly defined as primarily a visual search task (e.g., Fleetwood & Byrne, 2003; Everett & Byrne, 2004; Hornof & Kieras, 1997, 1999) . Our research suggests this may be an important part of routine procedure execution even when visual search may *not* appear to be a dominant factor. Furthermore, it appears that it is neither the case that the most optimistic assumption (users memorize the location of all controls) or the most pessimistic assumption (users search randomly every time) is an appropriate representation of user behavior, at least at this level of skill. This suggests that more research is needed on the integration of cognitive mechanisms such as representing and traversing goal structures with visual-cognitive mechanisms such as search strategies. While we doubt anyone would have denied that this is an important domain in a general sense, we suspect that most researchers in this area would underestimate the impact such considerations might have on execution of routine procedures.

To end on a speculative note, consider the hint provided by Phaser step 5 (settings). In that case, the Hier models are too slow and the Flat models are spot-on, suggesting that the goal traversal performed by the model is not being done by the participants. This might be an indicator that the participants have re-configured their internal representation of the task structure to match the visual structure of the interface! This suggests a possibly important role for the match between the task structure and the visual layout of an interface, something clearly not predicted by extant GOMS-class models.

Acknowledgements

We would like to acknowledge the support of the Office of Naval Research under grant number N00014-03-1-0094 to the first author. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ONR, the U.S. Government, or any other organization.

References

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

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Quin, Y. (in press). An integrated theory of the mind. To appear in *Psychological Review*.

Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies, 55(1)*, 41–84.

Byrne, M. D. (2003). A mechanism-based framework for predicting routine procedural errors. In R. Alterman & D. Kirsh (Eds.) *Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

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

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

Ehret, B. D. (2002). Learning where to look: location learning in graphical user interfaces. In *Human Factors in Computing Systems: Proceedings of CHI 2002* (pp. 211-218). New York: ACM.

Everett, S. P., & Byrne, M. D. (2004). Unintended effects: Varying icon spacing changes users' visual search strategy. *Human Factors in Computing Systems: Proceedings of CHI 2004* (pp. 695–702). New York: ACM.

Fleetwood, M. D. & Byrne, M. D. (2003). Modeling the visual search of displays: A revised ACT-R/PM model of icon search based on eye-tracking and experimental data. In F. Detje, D. Dörner, & H. Schaub (Eds.) *Proceedings of the Fifth International Conference on Cognitive Modeling* (pp. 87–92). Bamberg, Germany: Universitas-Verlag Bamberg.

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied, 6*, 322-335.

Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: A validation of GOMS for prediction and explanation of real-world task performance. *Human-Computer Interaction, 8*, 237–309.

Hornof, A. J., & Kieras, D. E. (1997). Cognitive modeling reveals menu search is both random and systematic. In *Human Factors in Computing Systems: Proceedings of CHI 97* (pp. 107–114). New York: ACM Press.

Hornof, A. J., & Kieras, D. E. (1999). Cognitive modeling demonstrates how people use anticipated location knowledge of menu items. In *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems* (pp. 410-417). New York: ACM.

John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction, 3*, 320-351.

Kieras, D. E., & Meyer, D. E. (2000). The role of cognitive task analysis in the application of predictive models of human performance. In J. M. Schraagen & S. F. Chipman (Eds.), *Cognitive task analysis* (pp. 237-260). Mahwah, NJ: Erlbaum.

Kieras, D. E., Wood, S. D., & Meyer, D. E. (1997). Predictive engineering models based on the EPIC architecture for multimodal high-performance human-computer interaction task. *Transactions on Computer-Human Interaction, 4*(3), 230-275.

Ohlsson, S. (1996). Learning from performance errors. *Psychological Review, 103*, 241–262.

Pylyshyn, Z. (1999). Is vision continuous with cognition? The case of impenetrability of visual perception. *Behavioral & Brain Sciences, 22*(3), 341-423.