Decomposing Interactive Behavior

Michael J. Schoelles (mschoell@gmu.edu)
&
Wayne D. Gray (gray@gmu.edu)
George Mason University
Fairfax, VA 22030 USA

Abstract

Interactive behavior emerges from the interaction of embodied cognition with task and the artifacts designed to accomplish the task. The current study focuses on how subtle changes in interface design lead to changes in the cognition, perception, and action operations that compose interactive behavior. The Argus Prime task is explained and the nature of the modeling effort is discussed. Insights obtained by exploring differences between model and human performance in one aspect of the Argus Prime task are presented.

Introduction

The Argus Prime simulated task environment (Gray, in press) places subjects in the role of radar operators whose job it is to assess the threat value of targets on a radar display. Our goal is to determine the strategies that people use in performing the task and to study how these strategies change as a function of subtle changes in interface design. Cognitive models are built that implement these strategies at the embodiment level (Ballard, Hayhoe, Pook, & Rao, 1997). Changes in strategy that accompany changes in the interface are interpreted as due to least-effort trade offs among the cognitive, perceptual, and action elements of embodied cognition. This work has implications for interface designers of dynamic systems characterized by rapid shifts of attention and time-pressured decision making such as in air traffic controllers, emergency medical systems, and nuclear power plant systems.

Our models are written using ACT-R/PM (Byrne & Anderson, 1998) — an architecture of cognition that enables us to capture the parallelism between cognition, perception, and action. By getting the interactions right at the embodiment level (approximately one-third of a sec), we hope to reproduce process and outcomes all the way up to the scenario level (each scenario requires 12-15 min to complete).

In comparing our models to human performance, we have been alternatively pleased and disappointed. It is not uncommon for our models to match the overall performance of our human subjects (at the 12-15 min level) only to mismatch greatly at a finer level of analysis.

When part of the model misfits its part of the data, we attempt to base changes of the model on a combination of two classic approaches. First, we observe subjects and analyze action protocols of their behavior. The action protocols include response times, eye movements, and mouse movements. Second, we introduce a small change to one part of the interface. We then run the model on the two versions of Argus and compare it s predictions with empirical data collected from human subjects.

Subtle changes in interface design may result in large changes in the strategies used to perform the task. For example, in Argus Prime it is important to maximize time on unclassified targets by, in part, minimizing time spent on targets that have already been classified. Hence, a change in interface design that varies the display-based indication of a target s classification status (classified or not classified) may have a profound effect on the number and combination of cognition, perception, and action operations used to perform Argus Prime.

In this paper, we marshal both human and model data to interpret the effect of interface changes on cognitive as well as on perceptual-motor performance. We use a broad brush to describe our task, current study, and model. After discussing how well the model s performance matched overall human performance, we limit the rest of the paper to two subparts of the Argus Prime task; namely, target selection and target check. These subparts provide an example of how subtle changes in interface design can produce unexpected interactions at the embodiment level.

Argus Prime: Simulated Task Environment

Argus Prime is a complex but tractable simulated task environment. In Argus Prime the subject s task is to assess the threat value of each target in each sector of a radar screen depicted in Figure 1. The screen represents an airborne radar console with ownship at the bottom. Arcs divide the screen into four sectors; each sector is fifty miles wide. The task is dynamic since the targets have a speed and course. A session is scenario driven; that is, the initial time of appearance, range, bearing, course, speed, and altitude of each target are read from an experimenter-generated file. The

scenario can contain events that change a target s speed, course, or altitude. New targets can appear at any time during the scenario.

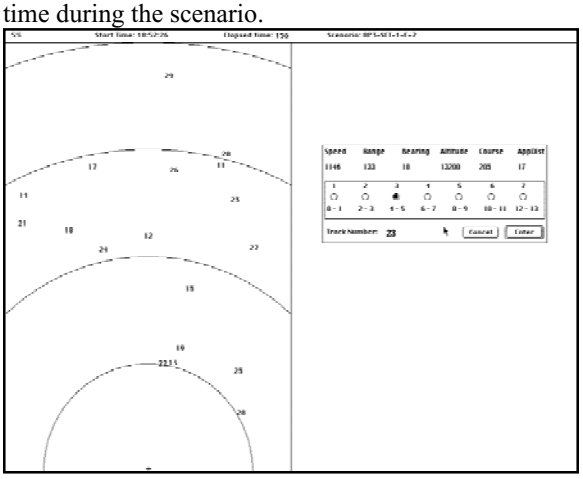

Figure 1: Argus Prime Radar Screen (left) and Information/Decision Window (right)

The subject selects (i.e., hooks) a target by moving the cursor to its icon (i.e. track number) and clicking. When a target has been hooked, an information window appears that contains the track number of the target hooked and the current value of target attributes such as speed, bearing, altitude, and course. The subject s task is to combine these values, using an algorithm that we have taught them, and to map the result onto a 7-point threat value scale (shown on the right side in Figure 1.)

Targets must be classified once for each sector that they enter. If a target leaves a sector before the subject can classify it, it is considered incorrectly classified and a score of zero is assigned.

For the versions of Argus Prime discussed here, immediate feedback was provided for each classification decision. In addition, summative feedback was provided on the percentage of targets correctly classified. (See Schoelles & Gray, in press, for more details.)

Empirical Study

This paper provides a partial report on the third study that we conducted. The results of prior studies indicated that a ubiquitous feature of the task was keeping track of which targets had been classified. In those studies there was nothing on the radar screen to indicate whether a target had been classified; that is, when a classification was made, its on-screen icon did not change (noChange). However, in both studies, if an already classified target was reselected, the target s current classification (CC) was shown in the information window (i.e., its radio button, see Figure 1, remained highlighted). We call this combination of no change to the target s on-screen icon and persistence of the classification in the information window the noChange-CC interface.

The current study manipulates the ease of retrieving status information (i.e., Is this target classified?) from the display. In addition to noChange-CC, two new interfaces are used. The noChange-noCC interface is similar to the noChange-CC in that the target s on-screen icon does not change when a classification is made. It differs from noChange-CC in that the information window contains no record as to whether a target is currently classified (i.e., once the ENTER key is pressed, the radio button is unhighlighted, see Figure 1). In contrast, for the Change interface the on-screen icon for targets that have been classified changes color. When a target is no longer classified (i.e., when it crosses a sector boundary) the icon reverts to the unclassified color.

In the first two studies subjects frequently reselected already classified targets. Their pattern of behavior suggested that for the noChange-CC interface subjects did not try to remember whether a target had already been classified. Rather, the pattern suggested that subjects simply clicked on targets until they found one that was not classified.

It was unclear in the previous studies whether this memory-less strategy (Ballard, Hayhoe, & Pelz, 1995) is adopted by choice or whether, under the conditions of the study, human cognition is incapable of retrieving target status information. This issue is tested empirically and analytically by the data and models built for the current study.

Performance on the noChange-CC interface is used as a baseline with which to compare the other conditions. We expect the noChange-noCC interface to force the memory versus memory-less issue. If subjects have no memory for having classified a target, they will be required to waste time re-computing the algorithm to reclassify already classified targets. In contrast, the Change condition provides a memory-less way to avoid classified targets and to focus on unclassified ones. Hence, subtle changes in the interface will enable different sets of strategies between the three conditions. These different strategies are expected to be differentially successful and to result in stable differences in performance.

The experiment was conducted over two sessions. In the first 2-hr session the subjects were instructed on how to do the task, did a sample scenario with the experimenter, and then did five 12-min scenarios in the noChange-CC condition. In the second 2-hr session the subjects did a 12-min practice scenario in the noChange-CC condition and then did two scenarios in each of the three conditions (noChange-CC, Change, noChange-noCC).

The Model

Our model runs under ACT-R/PM with the Eye Movements and Movements of Attention Extension (EMMA) (Salvucci, 2000). The ACT-R/PM architecture combines ACT-R s theory of cognition (Anderson & Lebiere, 1998) with modal theories of visual attention and motor movement (Kieras & Meyer, 1997). ACT-R/PM explicitly specifies timing information for all three processes as well as parallelism between them. The software architecture facilitates extensions beyond the modal theory of visual attention and motor movements.

The ACT-R/PM code executing the model runs as a separate process from Argus Prime. This process starts when the scenario starts. All communication between the model and Argus Prime is through the motor and vision module commands of ACT-R/PM.

Model Description

The recurrent task of hooking a target can be analyzed into a series of unit tasks (Card, Moran, & Newell, 1983): target selection; target check; target classification; and feedback. Each unit task has memory retrieval, visual attention, and mouse movement requirements. In ACT-R retrieval latency is a function of the activation of the memory element being retrieved. In addition, if the activation of a memory element is not above threshold, the retrieval will fail.

Movement of attention is a combination of two ACT-R/PM commands. The *Find Location* command is a pre-attentive search for a feature that returns a location to use as a parameter in the *Move Attention* command. The Move Attention command encodes a declarative memory element representing the visual object at the specified location. With the EMMA extension, a series of eye movements follows the initiation of the move attention command. The time to encode the visual object is a function of the eye movements.

Mouse movements are executed via ACT-R/PM s *Move Mouse* command. The input to this command is an object representation. The time to complete the movement is a function of Fitts Law. Mouse clicks are executed with the *Click Mouse* command. The overall operation of the model is an interleaving of productions that perform the cognitive operations of memory retrieval and goal modifications with the perceptual-motor operations of pre-attentive search, movement of attention, eye, and mouse movement.

In this paper, we focus on the target selection and target check unit tasks. In all three conditions (noChange-CC, noChange-noCC, and Change) the model begins target selection by retrieving a memory trace of the area in which it is currently searching; for example, the lower right-hand portion of the radar screen. It then pre-attentively searches for targets within this area. If a target is found, attention is moved to the feature to encode the target. (The track number is part of the encoded representation.) At that point the Move Mouse command fires and the cursor moves to and clicks on the target.

The above procedure varies slightly as a function of interface condition. In the noChange-CC and the noChange-noCC conditions, after a target is found and encoded, but before the Move Mouse command is executed, the model attempts to perform a target check by retrieving an episodic trace of a previous classification of the track number. If it retrieves this trace then it knows that the target is already classified; hence, the model will search for another target. If it cannot retrieve the trace, then the actions of moving the cursor to the target and clicking on it are performed.

In the noChange-CC condition, after clicking on a target the model will do a second target check by conducting a feature search in the information window to detect the highlighted radio button. If one is found the search for a new target will begin. Otherwise the Target Classification unit task begins. The noChange-noCC condition does not have this double-check. If it cannot retrieve a memory that the target is already classified, it will reclassify the target.

In the Change condition, targets change color after they are classified. As a consequence, the distinction between the target selection and target check unit tasks disappears. Hence, after a search area is retrieved, the pre-attentive search looks for the color feature that separates the unclassified targets (yellow) from the classified ones (blue). This strategy is purely memory-less in that no use is made of the episodic information regarding a target s prior classification. (After all targets in the retrieved area have changed color to blue, the model will do a feature search over the entire screen for yellow targets.)

Model and Subject Data Comparison

There are three limits to the model and analysis. The first is that the model was not fit to individuals. The same configuration and architectural parameters were used for all runs of the model.

Second, within an interface condition, the model uses the same strategies throughout the scenario. For example, the model uses the same target selection and target check strategies for the initial phase of the scenario when no targets had been classified as for the later stages when most targets had been classified.

Third, the base model was developed on the noChange-CC condition. In general, the way in which the model performed each unit task (i.e., target selection, target check, target classification, and feedback) was based on strategies that we observed our subjects using in the first two studies. As the unit task strategies required from 3 to 30 sec to execute, our

caveat is that in ACT-R/PM these strategies were implemented at the embodiment level using productions that required 50-100 msec to execute. Hence, the implementation of the various unit task strategies required us to make assumptions regarding memory retrieval, attention shifting, and motor movements for which we did not yet have empirical support.

In summary, at the unit task level of analysis the models implemented strategies that were cognitively plausible. For example, for the Change interface condition, pilot subjects told us that they performed target selection by looking for yellow targets. The implementation of such strategies at the embodiment level was based on our knowledge of the pre-attentive search literature, the ACT-R/PM cognitive architecture, and inspired guesses. The testing of those inspired guesses is what the current effort is all about.

Statistics Used

To compare human and model data, we report ANOVA and planned comparisons. A measure of variability for our model subjects is derived as follows. For each of 12 human subjects, a model subject was created. This model subject received the same six scenarios as the corresponding human subjects received during the second session. That is, if subject 1 did scenario 1 and 2 in the noChange-CC condition, scenario 3 and 4 in the Change condition and scenario 5, and 6 in the noChange-noCC condition, then a model subject was run on the same set of scenarios and under the same interface conditions. Hence any variability between model subjects (within condition) can be attributed to (a) unintended differences in how the 12 scenarios were designed, and (b) the randomness built into the architecture. Unlike human subjects, all model subjects in the same condition always follow the same strategies.

We confess that our second set of statistics is a willful abuse of ANOVA. The practical outcome of this is to inflate our Type I error rate; that is, the reduced variability due to model subjects should lead us to identify more differences between model subjects and human subjects than actually exist. We accept this inflated Type I error rate. The cost to our research of an inflated Type I error rate will be to cause us to spend time and attention looking for differences in strategies at the embodiment level that may not exist.

For each figure, we present the 95% confidence interval for human subjects and model subjects. The root mean square deviation (RMSD) of each human subject from his or her scenario-matched model subject is also reported.

Total Task Performance Comparison

Interface condition has a significant effect on the performance of human subjects, $F(2, 22) = 31.71$, $p < .0001$, $MSE = 33.9$. As Figure 2 shows, the Change condition does best (85.5%), followed by noChange-CC (76.4%), with noChange-noCC (66.6%) the worst. Planned comparisons show that the difference between each pair of conditions is significant at the level of significance adopted for this report ($p < .05$).

Overall, our model subjects do about as well as our human subjects (see Figure 2). However, although the main effect of model versus human is not significant, $F(1, 22) = 1.64$, $p = .21$, the interaction by interface condition is [$F(2, 44) = 7.3$, $p < .002$]. The Figure suggests that humans do slightly better than the model for noChange-CC and Change conditions but about equal to the model for noChange-noCC. As Figure 2 shows, the model makes the explicit prediction that the two noChange groups will have equal performance (noChange-CC = 67.8%; noChange-noCC = 67.5%). The significant interaction suggests that our human subjects are reacting to the interface conditions in a way that the model does not.

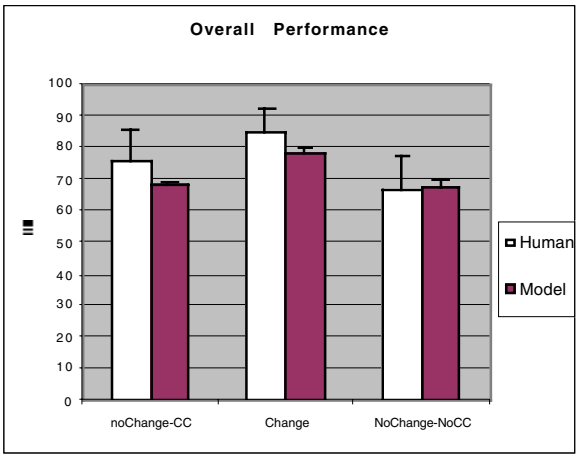

Figure 2: Overall Performance comparison of human and model. The RMSDs are 17 for noChange-CC, 13 for Change and 16 for noChange-noCC.

The variability shown by the model reflects differences between scenarios, not differences in strategies within conditions, and not differences inherent to individual subjects. Hence our model-driven approach provides us with an independent way of accessing the equivalence of the scenarios. As shown by the small confidence interval for the model subjects, our efforts to create equivalent scenarios was largely successful.

Target Selection Comparison

The three interface conditions showed significant differences in the number of *unclassified* targets

selected. The Change condition selected the most unclassified targets (68.3), followed by noChange-CC (58.3), and then by noChange-noCC (50.0). The model differences mirrored the human differences.

The more interesting comparison examines the probability of reselecting (or rehooking) an already classified target. For humans, planned comparisons show that noChange-CC rehooks the most targets (79.1) with there being no statistical difference in the number of targets rehooked by the Change (3.7) and noChange-noCC (16.3) conditions. [The overall ANOVA yields a significant effect, $F(2, 22) = 44.3$, $p < .0001$, $MSE = 441.3$.]

Comparing model performance with human performance yields a number of small surprises. Although the overall human versus model comparison is not significant ($F < 1$), we find a significant interaction of model versus human by interface condition, $F(2, 44) = 5.3$, $p < .008$, $MSE = 228.8$). Compared to humans, the model rehooks fewer targets in the noChange-CC condition, the same number of targets in the Change condition, and more targets in the noChange-noCC condition.

In both noChange conditions, prior to selecting a target the model attempts to retrieve a memory of whether that target had been classified. Only if the retrieval fails will the model rehook an already classified target. The fact that humans rehook more targets than the model in the noChange-CC condition, implies that humans rely less on memory retrieval, in this condition, than does the model. In this condition, a memory check is, in some sense, unnecessary as clicking on the target will open the information window that will clearly show whether a radio button is highlighted or not.

It may be that the cost of a perceptual-motor check is so much less than the cost of encoding and retrieving a memory that the noChange-CC condition relies on a single activity strategy, rather than one that involves dual activities (i.e., memory and perceptual-motor).

To investigate this further, the model was modified to only perform a perceptual-motor check; that is, to exclude the attempted memory retrieval. As shown in Figure 3, the model without memory selects many more targets than do human subjects. The fact that the two models bracket human performance (see also Gray & Boehm-Davis, 2000) suggest that the memory-less strategy is the preferred but not the exclusive strategy. We are currently interrogating the human data for clues as to the circumstances under subjects in the noChange-CC condition will use a memory retrieval strategy.

In contrast to noChange-CC, the perceptual-motor strategy is not available to the noChange-noCC condition. In this condition, the cost of the failure of the memory retrieval strategy is high. Reclassifying an already classified target is effortful; consuming time that would be better spent classifying an unclassified target. Hence, this greater downstream cost may lead human subjects to encode a memory trace to a higher level of activation than the model. Alternatively, it may lead them to more attempted retrievals than the model or, perhaps, to adopt a lower retrieval threshold than the model.

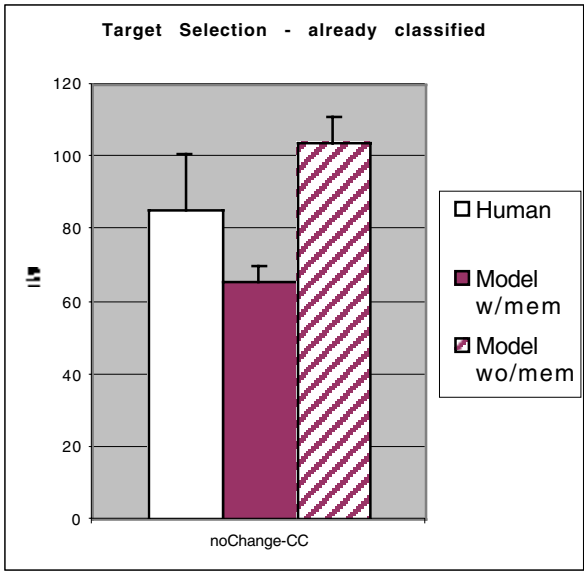

Figure 3: Target Selection for Human and Model with and without a memory retrieval. Model w/mem attempts to retrieve an episodic trace of the encoded target; only if the retrieval fails will a perceptual-motor check be performed. Model wo/mem will only perform the perceptual-motor check. The RMSDs are 42 for the human and model w/mem and 47 for the human and model wo/mem.

Our empirical data shows that subjects in the noChange-noCC condition reliably require 100 msec more than in the other conditions to classify a target. As it is not obvious why classification per se should take longer. However, the extra 100-msec is just enough time to sneak in one extra retrieval of the trace of the target just classified (Altmann & Gray, 1999), thereby increasing the success of the memory strategy for the Target Check unit task. Models incorporating this 100 msec of extra strengthening are being built and will be tested to determine if this strengthening suffices to produce the increment in performance shown by humans over the current model.

Discussion

Performance of the model subjects can be viewed as the embodiment of our theory of human performance. Comparisons that yield a significant main effect of model versus human signal places where our theory of human performance breaks down. Comparisons that yield a significant interaction of model versus human signal places in which our understanding of how

interface design influences interactive behavior are deficient. With this as our perspective, what does the performance of our model subjects tell us about our understanding of human interactive behavior?

A message that comes through loud and strong is that if our goal is to understand cognitive processes and not simply to predict performance outcomes, then obtaining good fits to an overall performance measure, such as Total Task Performance, can be misleading. The fit between model and human on overall performance can mask large and important differences in unit task performance.

On the first two unit tasks, Target Selection and Target Check, neither the main effect nor interaction of model versus human was significant for number of unclassified targets selected (first hook). This excellent fit of model to data broke down when we examined the number of times a target was rehooked. In this case the interaction indicated much more rehooking for noChange-CC than expected and less rehooking for noChange-noCC than expected. This interaction could be explained if the noChange-CC condition relied more on a perceptual-motor strategy and less on memory than did the model. Similarly, the noChange-noCC condition may be encoding the episodic trace of already hooked targets more highly than we had anticipated.

Conclusions

The goal of our research effort is to understand how subtle changes in interface design may lead to large changes in overall performance. As interactive behavior emerges from the interaction of embodied cognition with task and the artifacts designed to accomplish the task, an explanation of performance changes requires a consideration of the fine details of this interaction. In this article we have focused on one type of change and its effect on one part of task performance.

Although the fit of our model to overall performance was good, examining the fit of the model at the unit task level revealed important mismatches. For the Target Selection and Target Check unit tasks, the initial selection of unclassified targets was well fit by the model but the rehooks were not. Analyses of the model and the ways in which it matched and mismatched the data suggested three distinct target checking strategies that varied in their reliance on perceptual-motor versus memory operations.

Acknowledgments

This work was supported by Air Force Office of Scientific Research Grant # F49620-97-1-0353. We thank the many members of the Argus Group who have contributed to the Argus Prime studies: Erik M. Altmann, Deborah A. Boehm-Davis, Jeni Paluska, and Ryan Snead.

References

Altmann, E. M., & Gray, W. D. (1999). Serial attention as strategic memory. In M. Hahn, & S. C. Stoness (Eds.), *Proceedings of the Twenty-First Annual Conference of the Cognitive Science Society* (pp. 25-30). Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Lebiere, 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.

Ballard, D. H., Hayhoe, M. M., Pook, P. K., & Rao, R. P. N. (1997). Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences, 20*(4), 723-742.

Byrne, M. D., & Anderson, J. R. (1998). Perception and action. In J. R. Anderson, & C. Lebi re (Eds.), *The atomic components of thought* (pp. 167-200). Hillsdale, NJ: Erlbaum.

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

Gray, W. D. (in press). Simulated task environments: The role of high-fidelity simulations, scaled worlds, synthetic environments, and microworlds in basic and applied cognitive research. *Special Joint Issue of Cognitive Science Quarterly and Kognitionswissenschaft*.

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*(4), 322-335.

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.

Salvucci, D. D. (2000). A model of eye movements and visual attention. In N. Taatgen, & J. Aasman (Eds.), *Proceedings of the Third International Conference on Cognitive Modeling* (pp. 252-259). Veenendal, The Netherlands: Universal Press.

Schoelles, M. J., & Gray, W. D. (in press). Argus: A suite of tools for research in complex cognition. *Behavior Research Methods, Instruments, & Computers*.