Anticipatory Eye Movements in Interleaving Templates of Human Behavior

Michael Matessa (mmatessa@arc.nasa.gov)
NASA Ames Research Center, Mail Stop 262-4
Moffett Field, CA 94035 USA

Abstract

Performance modeling has been made easier by architectures which package psychological theory for reuse at useful levels of abstraction. CPM-GOMS uses templates of behavior to package at a task level (e.g., mouse move-click, typing) predictions of lower-level cognitive, perceptual, and motor resource use. CPM-GOMS also has a theory for interleaving resource use between templates. One example of interleaving is anticipatory eye movements. This paper describes the use of ACT-Stitch, a framework for translating CPM-GOMS templates and interleaving theory into ACT-R, to model anticipatory eye movements in skilled behavior. The anticipatory eye movements explain performance in a well-practiced perceptual/motor task, and the interleaving theory is supported with results from an eye-tracking experiment.

Introduction

Predicting skilled human performance by means of computer modeling is a valuable but difficult process. One easy way for modelers to describe performance would be a series of task-level (e.g., mouse move-click, typing) templates of behavior, laid end-to-end. But skilled performers do not complete all subcomponents of a task before going on to the next task. Instead, some subcomponents of the next task are interleaved into the earlier task. One example of this interleaving is anticipatory eye movements. It has been found that the eyes can move in anticipation of upcoming tasks in domains such as driving (Land & Lee, 1994), tea making (Land & Hayhoe, 2001), and hand-washing (Pelz & Canosa, 2001). In the hand-washing task example, while people perform the subtask of first getting their hands wet they interleave a look to the soap dispenser before performing the motor actions in the subtask of soaping their hands. So an easy-to-use but detailed modeling framework needs both task-level templates of human behavior and a theory of interleaving the lower-level perceptual, cognitive, and motor operators which make up the templates (Matessa et al., 2002). CPM-GOMS (John, 1988; 1990) is an example of such a framework, but it is only recently that the templates and interleaving theory of CPM-GOMS have been automated (John et al., 2002). Ongoing research is developing more templates and investigating the interleaving theory in computational systems. ACT-Stitch (Matessa, submitted) is a framework for automating the templates and interleaving theory of CPM-GOMS in the cognitive architecture ACT-R (Anderson & Lebiere, 1998). This paper will show how the interleaving theory of ACT-Stitch produces interleaved anticipatory eye movements which explain performance in a well-practiced perceptual/motor task. Then empirical support for the interleaving theory is given by results from an eye-tracking experiment.

CPM-GOMS

CPM-GOMS (John, 1988; 1990) uses templates of behavior to package at a task level (e.g., mouse move-click, typing) predictions of lower-level cognitive, perceptual, and motor resource use. Even behavior as simple as a mouse move and click requires coordination of the use of cognitive, perceptual, and motor resources. In order to carefully click on a target, it is necessary to find the target location, move the eyes to that location and perceive the target, verify the target location, move the cursor to the target location, and click the mouse button. CPM-GOMS templates are interleaved to reflect the ability of skilled people to perform parts of one task in parallel with another.

ACT-R

ACT-R (Anderson & Lebiere, 1998) is a computational theory of human cognition incorporating both declarative knowledge (e.g., addition facts) and procedural knowledge (e.g., the process of solving a multi-column addition problem) into a production system where procedural rules act on declarative chunks. Chunks are made up of slots containing information, and production rules which match the information in chunk slots are able to execute. The goal chunk represents the current intentions. Production rules have the ability to perceive objects and make motor movements through perceptual and motor buffers.

ACT-R does not have a built-in theory of multi-tasking which would interleave tasks, although some work has been done in modeling multi-tasking in the ACT-R architecture (Byrne & Anderson, 2001; Lee & Taatgen, 2002; Salvucci, 2002).

ACT-Stitch

ACT-Stitch (Matessa, submitted) uses a process of macro-compilation to translate CPM-GOMS templates of human behavior into ACT-R productions. More specifically, cognitive operators are translated into productions with ACT-R perceptual-motor commands that represent CPM-GOMS perceptual-motor operators. Productions also contain a control structure that allows ACT-R to implement CPM-GOMS interleaving and have productions from one template execute during the execution of productions from another template. This differs from the ACT-Simple system (Salvucci & Lee, 2003) that compiled a sequence of KLM-GOMS tasks into a series of productions which were controlled by an incrementing state counter.

Productions created from macro-compilation must ensure proper sequencing of motor actions, ensure the ability to allow the correct productions in future templates to interleave during the execution of productions in the current template, and ensure the ability to block the incorrect productions in future templates from interleaving with productions in the current template.

These three requirements are accomplished in productions by using information in the current goal as well as perceptual-motor buffers. Slots in the goal are created for the vision and hand resources for both the intended action and target making use of the resource. This makes four slots in the goal: vision action, vision target, hand action, and hand target. To ensure proper sequencing, the action slots in productions of the current template are filled with an intended action appended with the unique number of the current template. Also, the target slots are filled with an intended target. The intended action cannot be used alone since without the template number no sequence information would be stored. The template number cannot be used alone since there may be multiple actions in the same template using the same resource (e.g., mouse move and click). The intended target cannot be used alone since sequence information would be lost if a target appears twice in a sequence (e.g., clicking the same number twice). The intended target cannot be ignored since the same action could be used in a template for two targets (e.g., verify target and verify cursor).

To ensure the ability to interleave productions, separate action slots are used for each resource (vision and hand). This allows, for example, a procedure to initiate a vision action from a future template before a procedure initiates a hand action from the current template. To ensure the ability to block productions from future templates, the action slots are filled with intended actions appended with the current template number. This prevents, for example, moving to the next target while the hand resource is free between moving to the current target and clicking on the current target. The template number cannot be contained in a separate goal slot because that would not allow productions from the next template to execute before the productions of the current template have finished.

Perceptual-motor buffers are also used in sequencing. Productions that interact with the perceptual-motor buffers can fill or empty the buffers and can check the status of the buffers before using them.

These goal slots and buffers could be extended to include resources such as a left hand and buffers such as memory retrieval in future template development.

Empirical Validation

ATM Task

ACT-Stitch was applied to the automated teller machine task used by John et al. (2002) to test their automation of CPM-GOMS. The task was to make an $80 withdraw from a checking account on a simulation of an automated teller machine. Users interacted with the ATM by using a mouse to click on simulated keys or slots. The users were instructed to follow the following steps:

Insert card (click on the card slot)
Enter PIN (click on the 4, 9, 0, and 1 keys in turn)
Press OK (click on the OK button)
Select transaction type (click on the withdraw button)
Select account (click on the checking button)
Enter amount (click on the 8 and 0 keys)
Select correct/not correct (click on the correct button)
Take cash (click on the cash slot)
Select another transaction (click on the No button)
Take card (click on the card slot)
Take receipt (click on the cash slot)

This task was repeated 200 times by the users, and results were analyzed using the means of trials 51-100. This level of practice is comparable to that used by both Card, Moran, and Newell (1983) in a text editing task and Baskin and John (1998) in a CAD drawing task when they explored the effects of extensive practice on match to various GOMS models. As in John et al. (2002), Slow-Move-Click templates were used for clicking on targets that were difficult to select because of size and distance (e.g., the thin card slot) and Fast-Move-Click templates were used for easier targets (e.g., keypad keys). These templates were originally developed for the simple task of clicking on lit circles by Gray & Boehm-Davis (2000) and were successfully reused by John et al. to explain subject performance. The Fast-Move-Click template is made up of operators which find the target location, move the eyes to that location and perceive the target, verify the target location, move the cursor to the target location, and click the mouse button. The Slow-Move-Click template contains the same operators as the Fast-Move-Click template but in addition has operators to perceive the cursor and verify it is at the target. In order to determine eye movement durations in ACT-Stitch, the EMMA (Salvucci, 2000) extension to ACT-R was used.

To get an idea of what a template looks like after being compiled into ACT-R productions, the following shows pseudo-code for the Fast-Move-Click template. Each instance of a template in the task sequence list would have its own set of productions labeled by the position of the template in the list (x).

```
Tx-Init-Move-Cursor
IF
        right hand action goal is to move the cursor in this template
        right hand target goal is this template's object
        motor preparations have completed
THEN
        move cursor
        empty right hand target goal
        set right hand action goal to click the mouse in this template

Tx-Attend-Targ
IF
        vision action goal is to attend target in this template
        vision target goal is this template's object
        visual location and object buffers are empty
        vision is available
THEN
        fill visual location buffer with location where
          this template's object should be

Tx-Init-Eye-Move
IF
```

```
        vision action goal is to attend target in this template
        vision target goal is this template's object
        visual object buffer is empty
        visual location buffer holds object location
THEN
        fill visual object buffer with object at location
        empty visual location buffer

Tx-Verify-Targ-Pos
IF
        vision action goal is to attend target in this template
        vision target goal is this template's object
        right hand target goal is empty
        visual object buffer holds object at location y
        location y is the expected location of this template's object
THEN
        empty visual object buffer
        set visual action goal to attend in the next template
        set visual target goal to next template's object
        set right hand target goal to this template's object

Tx-Init-Click
IF
        right hand action goal is to click the mouse in this template
        right hand target goal is this template's object
        motor preparations have completed
THEN
        click mouse
        set right hand action goal to move the cursor in next template
        set right hand target goal to next template's object
```

Productions that initiate motor movements (Init-Move-Cursor and Init-Click) first check that the motor preparations from previous motor movements have completed. Since motor preparations can happen in parallel with motor executions and finishes in ACT-R, this means that preparations can start during previous executions and finishes. Productions could be written to wait for the previous executions and finishes to complete before starting preparations, but they would not be as efficient.

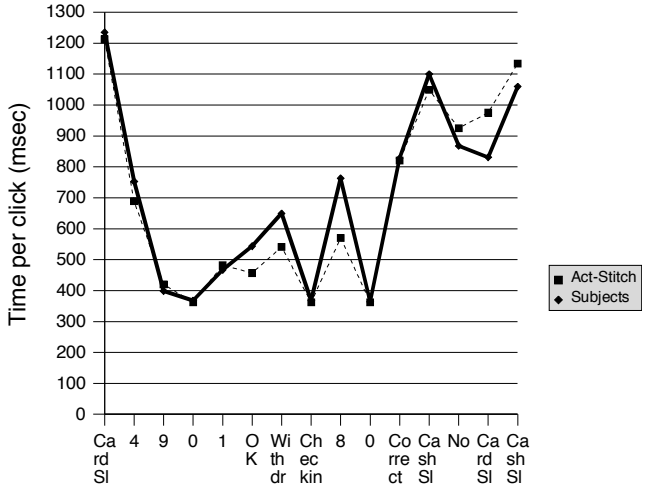

Figure 1: Average subject performance compared to ACT-Stitch predictions

Figure 1 compares ACT-Stitch predictions of mouse click times to average subject mouse click times of trials 51-100. The results are highly correlated (r=.96) with a low average absolute difference of 57msec.

The effect of interleaving on resource use is shown in PERT chart form in Figure 2. This output is from the Sherpa visualization tool developed by John et al. (2002) in their work to automate CPM-GOMS. The top row shows vision execution, the second shows vision preparation, the third cognition, the fourth shows motor preparation, and the bottom shows motor execution and finishing. Resource use is indicated with colored boxes, and instances of resource use in the same template are shown with the same color.

The figure is centered on the template for performing a Fast-Move-Click on the zero key (the lightest colored boxes), which is one of the fastest behaviors in the task for the subjects. ACT-Stitch explains this speed with an anticipatory eye movement to the zero key before the preceding nine key is clicked.

Sequential Response Task

To test the anticipatory eye movement prediction of the interleaving theory in ACT-Stitch, the sequential response task used by Wu & Remington (2004) was modeled. In this task, subjects viewed a series of five letters and responded to each individually. Subjects made sequential fixations to each of the five stimulus characters randomly drawn from the set T, D, and Z, and made choice responses mapped to three response keys (V, B, and N) on a PC keyboard and assigned to the first three digits of the right hand. Eye movements and key presses were recorded, and the stimulus letters were small enough and separated enough so that identification of stimulus letters required separate saccades and fixations. In Experiment 1 of Wu & Remington (2004), the effect of brightness of stimuli was investigated with dim and bright stimuli conditions, but no statistically significant differences were found. The predictions of the ACT-Stitch model will be compared to the results from the bright condition. Subjects were given 24 practice trials with the bright condition, then 120 trials divided into two blocks, one for each brightness condition.

The task was modeled in ACT-Stitch by creating a template for responding to a letter with an appropriate key press and applying this template to each stimulus letter.. The response template consisted of operators for finding the location of a letter, moving the eyes to that location and perceiving the letter, deciding a response, and pressing a key. As with the ATM task, the EMMA (Salvucci, 2000) extension was used to determine eye movement durations. The interleaving theory of ACT-Stitch predicts anticipatory eye movements where productions representing vision initiation operators from a future template can execute during the execution of productions representing operators in the current template.

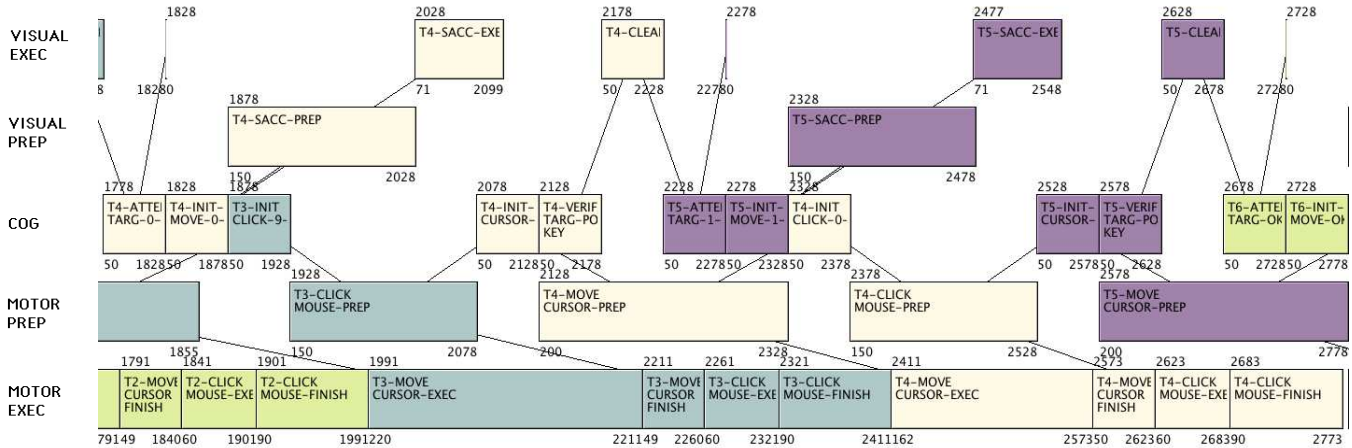

Figure 2: PERT chart of ACT-Stitch interleaving perceptual execution, perceptual preparation, cognitive, motor preparation, and motor execution and finishing resources in the ATM task

Figure 3 shows the time line of ACT-Stitch predictions for eye fixations and typing responses for the first four stimulus items (only four stimuli are presented because subjects had various strategies for where to fixate their eyes during the last stimulus response). Each horizontal bar represents the duration of response to a single stimulus, beginning with the eye fixation on a stimulus and ending with the typed response. The dark area represents eye fixation time ("fixation time") and the light area represents the time between moving the eye fixation to the next stimulus and the response to the current stimulus ("typing time"). Since the typed response for a particular stimulus occurs at the same time as fixations for the next stimulus, the figure shows that ACT-Stitch predicts anticipatory eye movements.

Figure 4 shows the time line of subject performance for eye fixations and typing responses in the bright condition of Experiment 1 of Wu & Remington (2004). The figure shows the anticipatory eye movements that the model predicts. The zero-parameter timing predictions are relatively close, with an average difference of fixation time of 126msec and an average difference of typing time of 64msec. The fixation and typing times of the model are consistently less than those for subjects, perhaps indicating that subjects are doing some processing that is not accounted for by the model. As can be seen in Figure 5 (a PERT chart representation of the model's performance), the fixation time is directly influenced by the time to decide on a mapping between letter and key. A post-hoc change of the decision time from 50msec to 150msec would reduce the average difference of fixation time between model and subjects to 26msec, while keeping the average difference of typing time at 64msec. Another feature of the data not accounted for by the model is the decreasing typing time over subsequent stimuli. These limitations of the model will be discussed later.

General Discussion

The interleaving theory of ACT-Stitch produces anticipatory eye movements that give a good account for data from two tasks, one showing quick motor response of measured mouse clicks and one showing anticipatory eye movements of measured eye fixations.

There is room for improvement, especially in the sequential response task. The fixation and typing times of the model are consistently less than subjects, perhaps indicating that subjects are doing some processing that is not accounted for by the model. Further work with the sequential response task done by Wu, Remington, and Pashler (submitted) shows that fixation times on a specific stimulus can be lengthened depending on the response of the previous stimulus, suggesting the processing of the previous stimulus is still occurring after the fixation is started. The ability of ACT-Stitch to interleave productions from different tasks will be useful in trying to develop models to explain this result. Wu et al. also replicate finding of the decrease in time between end of fixation and typed response. It is difficult to explain this result with identical templates that do not make reference to the number of stimuli remaining to be processed because the timing of perceptual and motor processing is linked by cognition. One possible solution may involve perceptual and motor processing of different durations that are decoupled from cognition by means of the motor buffers (derived from EPIC) or visual buffers (derived from EMMA). Since preparation can occur in parallel with execution in these buffers, preparations can put visual or motor actions in a queue that could decouple the start of the execution from the start of the cognitive initiation.

This paper offers only a first step of a template and interleaving theory in ACT-R. Many more templates are needed to test the robustness of the representations used for the interleaving theory. But this work is a first step to easier modeling and multi-tasking in ACT-R.

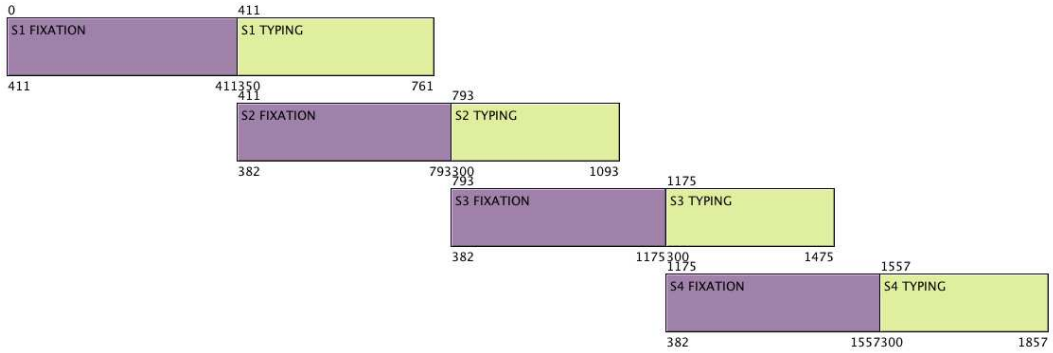

Figure 3: Time line of predictions for eye fixations and typing responses from the ACT-Stitch model

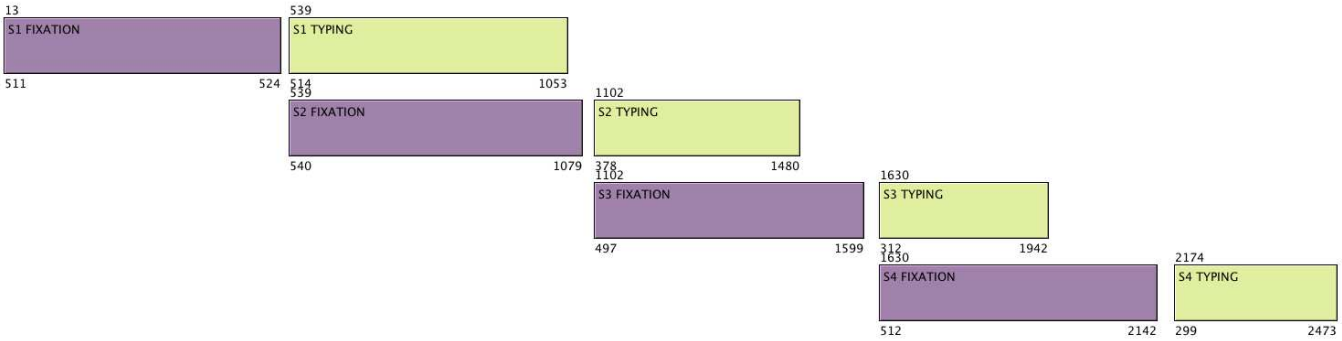

Figure 4: Time line of eye fixations and typing responses from subjects in Wu & Remington (2004)

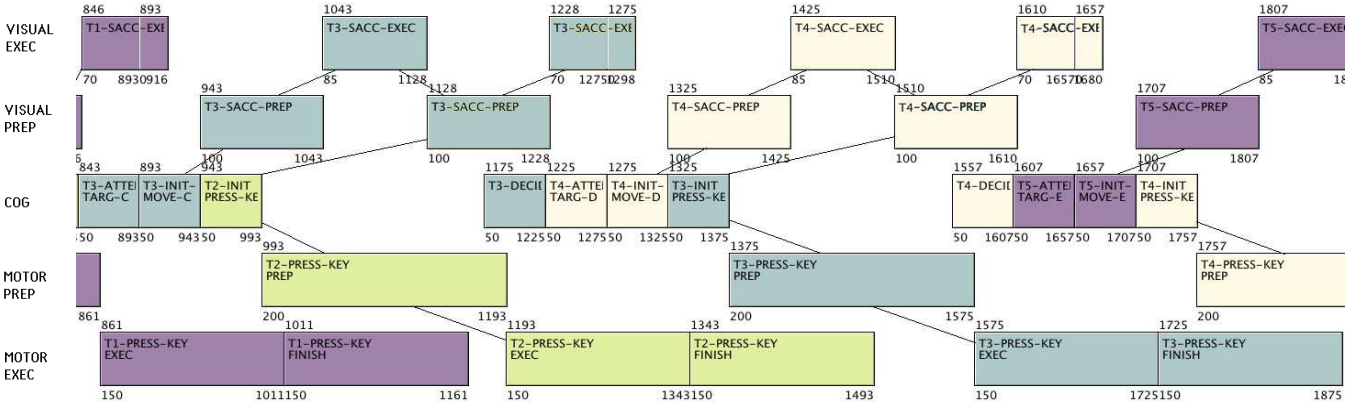

Figure 5: PERT chart of ACT-Stitch interleaving perceptual execution, perceptual preparation, cognitive, motor preparation, and motor execution and finishing resources in the sequential response task

Acknowledgments

This work was supported by Office of Naval Research grant N0001404IP2002 and by funds from the Airspace Operations System project of NASA's Airspace System program.

References

Anderson, J. R., & Lebiere, C. (1998). The atomic components of thought. Hillsdale, NJ: Erlbaum.

Baskin, J. D., and John, B. E. (1998). Comparison of GOMS Analysis Methods. Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems (Summary) 1998 v.2 p.261-262.

Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. Psychological Review, 108, 847-869.

Card, S. K., Moran, T.P. & Newell, A. (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Freed, M., Matessa, M., Remington, R. and Vera, A. (2003) How Apex automates CPM-GOMS. Proceedings of the Fifth International Conference on Cognitive Modeling, pp. 93-98. Bamberg, Germany:Universitats-Verlag.

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.

Gray, W. D., John, B. E. & Atwood, M. E. (1993) Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance, Human-Computer Interaction, 8 (3), pp. 237-309.

John, B. E. (1988) Contributions to Engineering Models of human-computer interaction. Ph.D. Thesis. Carnegie Mellon University.

John, B. E. (1990) Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. In proceedings of CHI, 1990 (Seattle, Washington, April 30-May 4, 1990) ACM, New York, 107-115.

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 (4), pp. 320-351.

John, B. E., Vera, A. H., Matessa, M., Freed, M., and Remington, R. (2002) Automating CPM-GOMS. In Proceedings of CHI'02: Conference on Human Factors in Computing Systems. ACM, New York.

Land, M. F., & Hayhoe, M. (2001). In what ways do eye movements contribute to everyday activities? Vision Research, 41, 3559-3565.

Land, M. F., & Lee, D. N. 1994. Where we look when we steer. Nature, 369, 742-744.

Lee, F.J. & Taatgen, N.A. (2002). Multi-tasking as Skill Acquisition. Proceedings of the twenty-fourth annual conference of the cognitive science society (pp. 572-577). Mahwah, NJ: Erlbaum.

Matessa, M. (submitted to 2004 Cognitive Science conference). An ACT-R framework for modeling interleaving templates of human behavior.

Matessa, M., Vera, A., John, B., Remington, R., & Freed, M. (2002). Reusable Templates in Human Performance Modeling. In Proceedings of the Twenty-fourth Annual Conference of the Cognitive Science Society.

Pelz, J. B. and Canosa, R. (2001). Oculomotor Behavior and Perceptual Strategies in Complex Tasks. Vision Research, 41, 3587-3596.

Salvucci, D. D. (2000). A model of eye movements and visual attention. In Proceedings of the Third International Conference on Cognitive Modeling (pp. 252-259). Veenendaal, The Netherlands: Universal Press.

Salvucci, D. D. (2002). Modeling driver distraction from cognitive tasks. To appear in Proceedings of the 24th Annual Conference of the Cognitive Science Society.

Salvucci, D. D., & Lee., F. J. (2003). Simple cognitive modeling in a complex cognitive architecture. To appear in Human Factors in Computing Systems: CHI 2003 Conference Proceedings. New York: ACM Press.

Wu, S.-C., & Remington, R. W. (2004). Coordination of component mental operations in a multiple-response task. Paper to appear in S.N. Spencer (Ed.), Proceedings of the Eye Tracking Research and Applications Symposium 2004. New York: ACM SIGGRAPH.

Wu, S.-C., Remington, R. W., & Pashler, H. (submitted to 2004 Cognitive Science conference). Coordination of component mental operations in sequences of discrete responses. Proceedings of the twenty-sixth annual conference of the cognitive science society. Mahwah, NJ: Erlbaum.