In Proceedings of the Sixth International Conference on Cognitive Modeling, 267-272.
Mahwah, NJ: Lawrence Earlbaum.

Toward an ACT-R General Executive for Human Multitasking

Dario D. Salvucci (salvucci@cs.drexel.edu)
Yelena Kushleyeva (yk45@drexel.edu)
Frank J. Lee (fjl@cs.drexel.edu)
Department of Computer Science, Drexel University
3141 Chestnut St., Philadelphia, PA 19104

Abstract

While cognitive models of complex tasks have begun to incorporate increasingly sophisticated models of human multitasking, most models have utilized customized executives (Kieras et al., 2000) that fine-tune specific multitasking mechanisms for particular applications. This paper proposes a general executive for multitasking that facilitates the integration of separate task models and subsequent prediction of the effects of multitasking and task interaction. Developed in the ACT-R cognitive architecture, the general executive is specified as a new goal module that orders goals by urgency and their own requested running times. The paper demonstrates the predictive power of the general executive in the driving domain by integrating separate models of control and monitoring and predicting drivers' gaze distributions across various regions of the driving environment.

Introduction

Whereas much of traditional cognitive modeling has focused primarily on cognitive processes in small-scale laboratory tasks, modeling efforts have been rapidly maturing to address increasingly complex task domains and phenomena. Some of the most successful models for such complex tasks have arisen in work using unified cognitive architectures such as ACT-R (Anderson et al., in press), EPIC (Meyer & Kieras, 1997), and Soar (Newell, 1990) — for example, models of driving (e.g., Aasman, 1995; Salvucci, Boer, & Liu, 2001), piloting combat aircrafts (Jones et al., 1999), and air-traffic control (e.g., Lee & Anderson, 2001). As such efforts evolve, the modeling community is facing several major new challenges in the study and modeling of complex tasks, and among the most critical of these is the study of human multitasking — the ability to integrate, interleave, and perform multiple tasks and/or component subtasks of a larger complex task.

Recent explorations of multitasking in cognitive architectures have examined the integration of two tasks, where the tasks are either discrete (i.e., short tasks of roughly 1-10 sec) or continuous (i.e., extended tasks). For instance, researchers have investigated how architectures can account for switching costs in successive discrete tasks (e.g., Sohn & Anderson, 2001), psychological refractory period (PRP) effects in concurrent discrete tasks (e.g., Byrne & Anderson, 2001; Meyer & Kieras, 1997b), and error effects in elementary continuous tasks (see Kieras et al., 2000). In addition, the models for the complex tasks listed above (driving, piloting, etc.) all incorporate some multitasking in having to perform various subtasks at various intervals (e.g., occasionally checking wind direction in the air-traffic control task). These models, however, use *customized executives* for multitasking (Kieras et al., 2000) that have been fine-tuned to the particular domain, resulting in domain-specific models of multitasking that can be difficult to generalize.

In this paper, we propose an initial formulation of a *general executive* for human multitasking in the ACT-R cognitive architecture. The general executive manages a set of current goals and dictates when each goal may proceed given ordering constraints based on desired initiation times for each goal. The proposed general executive attempts to balance the individual goals' desires for unhindered progress in each task with the overall system's need for fair resource allocation across tasks as well as achievement of higher-level goals. The executive also encourages modularity and model re-use by allowing for the integration of separate models, possibly developed in isolation of one another, into an integrated task model that interleaves multiple subtasks.

We first describe the general executive mechanism as developed in the framework of the ACT-R architecture, including a discussion of the many considerations that arise in specifying such a mechanism. We then demonstrate the predictive power and usefulness of the mechanism in an application to the domain of driving, showing how the general executive can automatically interleave the subtasks of control (steering and speed control) and monitoring (for situation awareness) and accurately model human drivers' allocation of attention to these two subtasks.

The ACT-R General Executive

The ACT-R general executive proposed here provides a mechanism that schedules and interleaves multiple subtasks. The goal of creating a general executive has both scientific and engineering implications. As a scientific endeavor, we desire a psychologically plausible mechanism that fits well theoretically with the existing cognitive architecture and generates sound predictions that fit available human data. As an engineering endeavor, we desire a mechanism that facilitates independent development of task models as well as model re-use. Of course, taken as a whole, the topic of multitasking spans an enormous array of phenomena and empirical literature. For the purposes of this paper, we focus specifically on the integration of two continuous tasks — that is, two tasks that must be performed continually for a long time (several minutes to hours) and must be interleaved at short intervals (hundreds of milliseconds to seconds), like the driver control and monitoring subtasks analyzed in a subsequent section. Of note, we are not currently addressing explicit discrete task switching or the one-shot behavior that arises in psychological refractory period (PRP) tasks; instead, we focus on continuous tasks in which people (and models) must schedule and interleave multiple tasks for an extended period of time, sharing cognitive and other resources to maintain execution of all tasks in a fair manner.

ACT-R, Buffers, and Goals

The ACT-R cognitive architecture (Anderson et al., in press; see also Anderson & Lebiere, 1998) is a production-system architecture that posits two types of knowledge: declarative knowledge comprising factual chunks and procedural knowledge comprising condition-action production rules. The most recent version of the architecture (5.0: Anderson et al., in press) centers on "buffers" that pass information between central cognition and various modules, such as the retrieval module for memory recall and the visual and aural modules for perceptual input. To use a module and its associated buffer, a production rule typically passes a request to the module and when the module has obliged the request, another production rule "harvests" the result by examining the buffer; for example, if a production rule requests that the visual module look at and encode an object at a specific location, the visual module performs this task and places the result in the visual buffer and a subsequent rule can examine this buffer and use the information therein.

One of the buffers posited in the architecture is the goal buffer, which stores the current goal (itself a declarative chunk) and directs the system to perform a specific task. One natural way to interpret multitasking in ACT-R is that multitasking represents the scheduling and management of what goal is currently in the goal buffer. Interestingly, although ACT-R has a goal buffer, it currently has no full-fledged module associated with this buffer as it does for other modules — production rules simply set the goal buffer explicitly without making requests in the same manner as the other buffers. Older versions of ACT-R (Anderson & Lebiere, 1998) used a goal stack that allowed for pushing onto and popping from the stack which in a sense provided an instantiation of a possible goal module, but the stack-based instantiation was difficult to justify from the standpoint of clear psychological plausibility (Altmann & Trafton, 2002). We view our ACT-R general executive as a specification for a new goal module, one that maintains a set of active goals and manages the setting of goals through the goal buffer.

Specification of the General Executive

The proposed general executive centers on the maintenance and scheduling of a *goal set* that contains all currently active goals. Maintenance of the goal set is accomplished by allowing production rules to add and remove goals to and from the goal set. To add a new goal, a rule specifies the new goal chunk and also notes the goal's *delay time* — that is, when the goal should run, specified as a time offset from the current time. The delay time, noted as a slot in the goal chunk, models the fact that some goals may not need to run immediately but rather can be scheduled for execution at a later time — for instance, the periodic monitoring of some aspect of a display, such as checking wind direction in the air-traffic control task (Lee & Anderson, 2001). For convenience, we also allow for a constant 'now' as special a value for the delay time that mandates that the goal execute immediately. For removal of goals, a rule simply notes the termination of the current goal and thus its removal from the goal set.

The implementation of this mechanism in ACT-R is fairly parsimonious. In the current ACT-R syntax, the command "+goal" sets the current goal to a new goal; we simply alter the command's semantics such that this same command *adds* a goal to the current goal set — in essence, specifying the desire that a certain goal be performed, much like requests to the retrieval or visual modules. For convenience and also to maintain consistency with current syntax, the first such command also removes the current goal from the goal set. Whenever a goal is added or removed, the goal set is reordered according to the scheduling criteria described previously, and the most urgent goal is placed in the goal buffer for execution.

Given the current set of goals, we require some method of ordering and scheduling the goals to ensure fairness across tasks. All goals on the set are ordered based on their *urgency* as indicated by their desired start time (i.e., creation time plus delay time): the executive simply orders the goals from most urgent to least urgent, that is, from earliest to latest start time. However, we also expect some amount of noise in this ordering scheme, just as many mechanisms in the ACT-R architecture and the human system itself exhibit variability in behavior. For this purpose, and to maintain consistency with the architecture, we utilize a logistic noise distribution on start times with variance $\sigma^2 = \pi^2 s^2 / 3$ as dictated by the noise parameter s; this same distribution is used for noise throughout the architecture including noise on expected gain of potential production-rule instantiations.

Figure 1 illustrates an example of how two task goals might be scheduled. In the first frame, Task 1 has a delay time of 400 ms while Task 2 is scheduled to execute right away. Even with noise added to these times, it is most likely that Task 2 will be allowed to execute. In the second frame, Task 2 (specifically a new goal that continues the task) has a new start time of 250 ms, and again this most likely wins out over Task 1. In the third frame, Task 2 has a later start time than Task 1, and thus the first task most likely proceeds; note that when the goals come closer together in time, as in the third frame, the noise increases the chance of a later goal superceding an earlier goal.

Theoretical and Practical Considerations

Kieras et al. (2000) have reviewed a number of theoretical issues and considerations for the development of a general executive within a cognitive architecture, many of which are related to basic concepts used in today's computer operating systems. One of the most critical issues is that of task

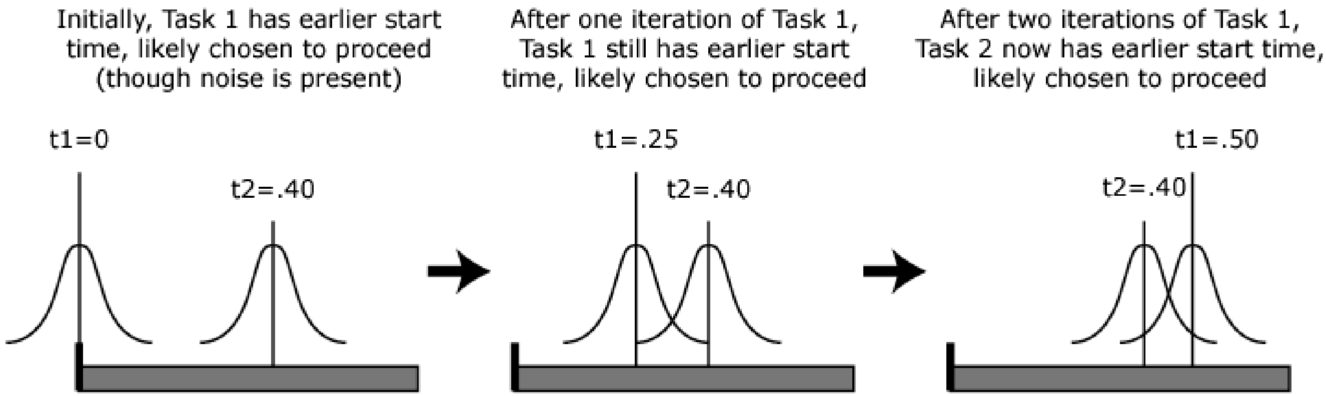


Figure 1: Example of scheduling and execution of two goals in the *goal set* as a function of time.

scheduling. Our proposed general executive uses a variant of a "first come, first served" scheduler that also appears in operating systems. However, the executive is more flexible than a strict first-come, first-served algorithm in two ways: first, it allows for time delays in the start time of processes (i.e., the goal delay times); and second, it incorporates noise into start times to enable randomness in task scheduling.

As a related point, another critical issue is that of conflict resolution, where the executive must avoid deadlocks within processes and ensure that no process experiences "starvation" with respect to needed resources. In a broader context, the executive must strike a balance between allowing processes to utilize necessary resources as much as possible but at the same time interrupting processes to allow others to proceed. Our executive interrupts processes *between* goals — corresponding roughly to the level of unit tasks — without interrupting *within* goals. We believe that this approach provides a good balance of fairness between tasks as desired by a robust general executive. It does, however, rely critically on the structure of declarative knowledge in that different declarative goal structures (e.g., as might arise from different instructions) may greatly affect when and how people perform multiple tasks. However, this reliance on declarative structure is not necessarily a disadvantage; in fact, a recent empirical study has demonstrated such a dependence (Vera, personal communication) and thus might be best modeled with an executive like that proposed here.

Along with such theoretical considerations, a general executive can have practical considerations in providing a robust workable mechanism that facilitates modeling in the cognitive architecture. One advantage of our proposed executive is that the task models can be developed independently of one another and later merged. Such modularity greatly facilitates model re-use and, in turn, increases the predictive power of newly integrated models (see, e.g., Salvucci, 2001). The executive also does not affect the behavior of existing ACT-R models, since existing models' explicit setting of goals is equivalent to a goal set with exactly one goal throughout execution.

Case Study: Driver Control and Monitoring

To illustrate how the proposed general executive facilitates the modeling of human multitasking, we now describe an application to the domain of driving. As a complex yet ubiquitous task, the task of driving is actually an integration of several subtasks performed fluidly by the driver to successfully navigate the environment. Of these various subtasks, the two (arguably) most critical subtasks are control and monitoring: control of steering and acceleration that maintains the car's safe position, speed, and heading; and monitoring that maintains a consistent, up-to-date awareness of the current situation as is critical for normal maneuvers (e.g., lane changes) or emergency maneuvers (e.g., swerving to avoid an obstacle). We now explore how the general executive facilitates the integration of basic models of driver control and monitoring into a single multitasking model that switches between these tasks. Of course, the evaluation of any general mechanism requires application not in a single domain but in many domains; nevertheless, we consider the driving application as an initial foray into evaluating and testing the benefits and limitations of our general executive in a complex, dynamic, real-world domain.

Component Models

To build a model of driver behavior in the spirit of the general executive, we started with models for the component subtasks involved in driving and then integrated these using the general executive. We extracted the component models from an existing ACT-R model of driver behavior (Salvucci, Boer, & Liu, 2001) that navigates a highway environment. The component models studied here are:

- **Control model**: The control model updates the positions of the steering wheel and the accelerator and brake pedals. This update is based on observed positions and changes in salient visual points in the driving environment (e.g., the vanishing point of the road or the position of a lead vehicle).

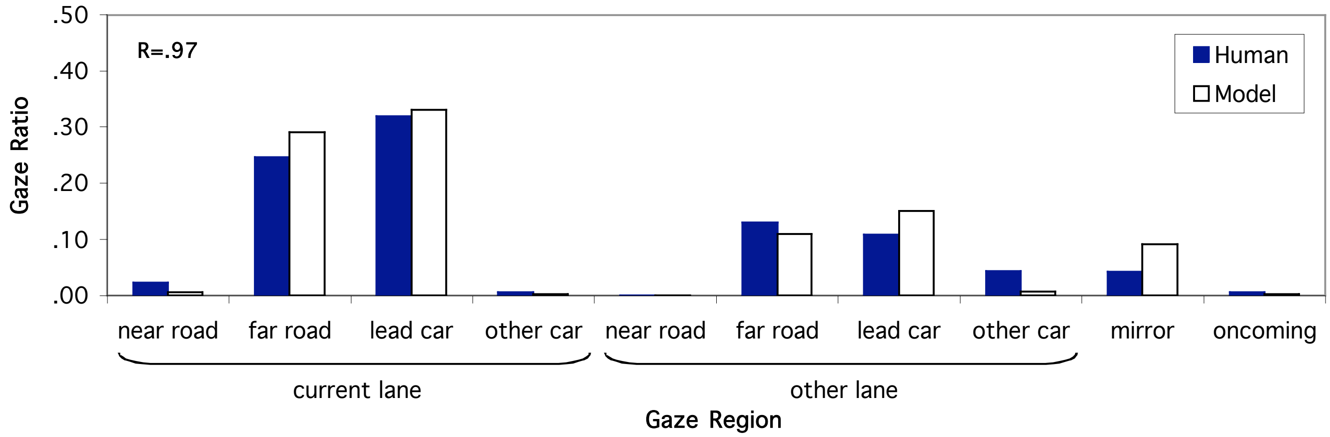


Figure 2: Ratio of gaze dwell time in various regions of the driving environment.

- **Monitoring model**: The monitoring model randomly samples the environment — with equal probability, it chooses a lane (left or right in a two-lane highway) and direction (forward or backward in the mirror) and looks there for the presence of a vehicle; if a vehicle is present, the model notes its position in declarative memory.

The original driver model has a third component model for decision making that decides whether and when to change lanes based on its current situation awareness; due to space constraints in this paper, we ignore decision making here and focus solely on control and monitoring. Readers can find further details about the component models in the full description of the model (Salvucci, Boer, & Liu, 2001).

The component models have associated goals *control* and *monitor* that perform the respective subtasks. It is critical to note that both of these goals perform incremental work of approximately 100-300 ms and then pass control to a new goal of the same type: one *control* goal will update steering and acceleration and then create a new *control* goal to perform the next iteration, and one *monitor* goal will attempt one glance in a chosen lane/direction and then create a new *monitor* goal to continue this task. As a result, both models operate independently (in fact, they could each run alone without the other) and continually refresh their respective subtasks with a new goal.

Model Integration

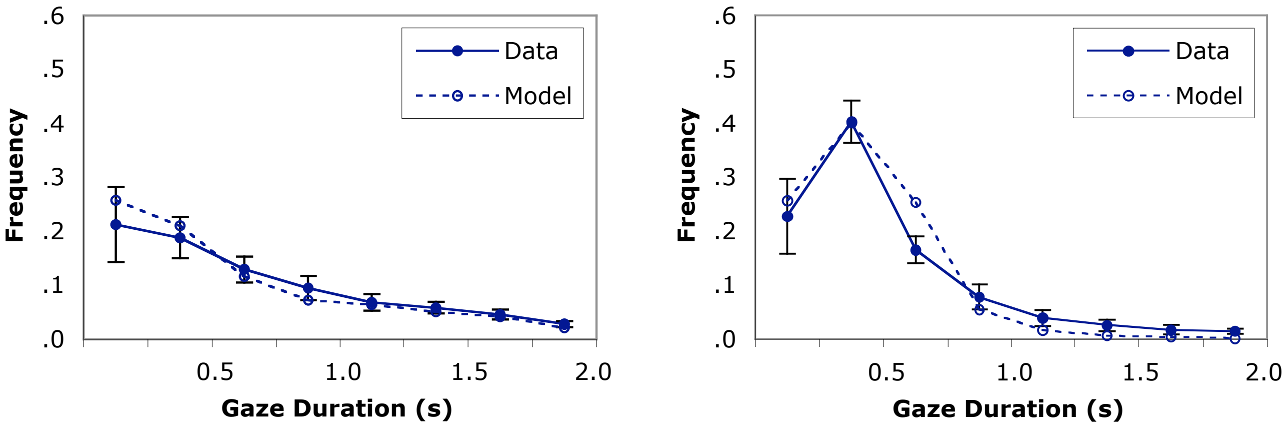
The creation of the original integrated model required a customized executive that passed control back and forth between control and monitoring. Now, with the general executive, model integration happens easily: we simply add one production rule that adds both *control* and *monitor* to the goal set, and the general executive handles the rest. As each incremental goal finishes and creates a new goal, the general executive reschedules the current goals in the set and allows the most urgent goal (as computed using start times with noise) to proceed.

The default integration above treats both subtasks equally, with one iteration of control and one of monitoring alternating one after another during execution. While this integration indeed successfully navigates a realistic environment, the resultant task switching does not necessarily match people's tolerance for when to switch between tasks. Thus, we explored two possibilities for changing the default scheduling:

- *Control stability*: The original driver model contains a stability threshold that switches away from control only when the external environment is "stable," where stability measures the lateral displacement from the lane center and the lateral velocity of the vehicle. We maintained this scheme in the new integrated model: if the control goal finds that the environment is *not* stable, it dictates that the next control goal should be done immediately by means of a *now* value for the *when* slot. In addition, we define a parameter F_{stable} that scales the threshold to allow us to estimate drivers' degree of stability acceptance.
- *Control delay time*: The default integrated model specifies a delay (*when*) time of 0 for new control goals, indicating that they should execute as soon as possible. However, given a stable environment, there is the possibility that control need not execute for a given period of time. Therefore, we define a second parameter $D_{control}$ that specifies the delay time for a new control goal assuming a stable environment.

We also have a third parameter dictated by the general executive, namely the *s* parameter that controls the amount of noise in the scheduling times; note that eventually as the general executive extends to more domains, this parameter should be estimated across all domains like the other noise parameters in the ACT-R architecture.

Parameter estimation was done first informally to find reasonable settings for parameter values, then more systematically to improve the fits to the results below. The resulting values for the parameters were as follows: F_{stable} = .71, $D_{control}$ = 500 ms, s = .075.



(a) (b)

Figure 3: Frequency of gaze durations in various regions of the driving environment for (a) control and (b) monitoring. Data points represent frequencies of durations within a .25 s range; bars indicate standard deviations.

Data Collection

To validate the integrated model, we would like to compare the model's behavioral data to that of human drivers. The human driver data comes from the original study (Salvucci, Boer, & Liu, 2001) in which 11 drivers navigated a multi-lane highway for a total of 311 minutes (548 km) of driving data. The model data comes from running three 10-minute simulations in the same conditions as the human data; because the model drives in the same environment as did human drivers in the driving simulator (minus the rendering of complex graphics), a model simulation generates a completely analogous protocol that can be analyzed in an identical manner as the human data.

Results

To analyze drivers' switching between control and monitoring, we examine and compare the eye movements produced by the human drivers and the integrated driver model (through its perceptual mechanisms and EMMA model of eye movements: Salvucci, 2001b). Figure 2 shows the ratio of gaze dwell time in various regions of the driving environment: the near road, far road, lead car, and other cars in the current lane; the same regions in the other lane (given the two-lane highway); the rear-view mirror; and oncoming cars (in the opposite two lanes). For both the human drivers and the model, the majority of the gaze is directed at the far road and lead car of the current lane — both critical regions for controlling the vehicle (both laterally and longitudinally). The far road and lead car of the other lane also garner some attention, as does the rear-view mirror, as indicators of occasional monitoring glances to these areas. Also, both humans and the model rarely look at the near road, at other cars in the two lanes, and at oncoming cars. Thus, the model nicely captures the overall distribution of gaze across the visual regions, R=.97.

While the overall distribution provides some sense of drivers switching between tasks, we can also analyze more revealing finer-grain data that indicate exactly when drivers switched between these tasks. As a first step in exploring this idea, we classified the gaze regions as either control or monitoring regions based on their function in the environment: the control regions included the far and near road regions as well as the lead car, and the monitoring regions included the lead and other car in the other lane as well as the rear-view mirror. Using this classification, we extracted all single gazes at one region (consecutive eye-movement samples ignoring single-sample "blips") and computed histograms of the frequencies of gaze durations, using 1/4-second increments to alleviate data noise.

Figure 3(a) shows the human and model gaze-duration histograms for control gazes, with points representing frequencies of gaze durations within a .25 s range. The graph of the human data (solid line) clearly indicates a drop over time, with the highest frequency (and highest standard deviation) occurring in the first 1/4 second and steadily decreasing out to 2 seconds. The pattern for the model data (dotted line) arises primarily from its stability threshold for switching: often the model can switch away after controlling up to 1/4 second, but on occasions when the vehicle incurs some instability (such as steering into and out of a curve), the model must stay with control until stability is again achieved. The model thus produces a very similar histogram to that of human drivers, R=.99.

Figure 3(b) shows the analogous histograms for monitoring gazes. For the human data, we see a very different pattern compared to the control gazes: the peak of the distribution lies around 1/4-1/2 s, and drops steeply thereafter, with very few gazes lasting more than 1 s. The model nicely captures the human data, the critical aspect being the control delay time incorporated into the model. In essence, the control goal requests a new goal only after the control delay time (500 ms), thus giving monitoring time to perform several iterations before switching back to control. At the same time, after the delay time threshold has passed, the model's urgency to switch back becomes increasingly larger, thus producing very few gaze durations greater than 1 second and a good overall fit to the human drivers' gaze pattern, R=.97.

General Discussion

We have described a general executive for multitasking in the ACT-R architecture that schedules multiple tasks and interleaves them to execute reasonable multitasking behavior. One of the most significant implications for this work is the ability to develop task models *independently* and later integrate them into a fuller model. While we have demonstrated such an integration for driver control and monitoring, there are of course many real-world applications for such integration. In the driving domain itself, there is a clear application with great potential — namely, to secondary device use and driver distraction. While recent work has shown the predictive power of integrated ACT-R models to account for driver distraction (Salvucci, 2001; Salvucci & Macuga, 2002), these models have customized executives that specify how the models should interleave driving and the secondary task. Now, the new general executive offers the potential to make *a priori* predictions about when and how interleaving takes place, allowing a modeler to develop secondary-task models independently and then easily integrate them with the driver model for predictions of their distraction potential.

This general executive nicely handles the type of multitasking in driving — and we would argue in similar complex dynamic tasks — that interleave multiple tasks at roughly the level of hundreds of milliseconds to a few seconds. However, there are clearly other levels of multitasking that are not yet addressed by this general executive. On one end of the spectrum, people can multitask at the level of several minutes to hours — e.g., word processing and checking email. On the other end, people can perform discrete rapid tasks together in a highly optimized manner to produce PRP effects (Byrne & Anderson, 2001; Meyer & Kieras, 1997b). At this time, we have focused our general executive on a level in between, but we hope that in the long term, this mechanism can generalize to account for multitasking at other such levels.

Another critical aspect involves the modeling of improvement in multitasking over time (Chong, 1998), or put another way, transitions between the "stages of multitasking skill acquisition" (Kieras et al., 2000). Learning in multitasking of course involves learning of the individual tasks, but even when people are skilled at particular individual tasks, the integration adds another layer of complexity (e.g., driving and talking on a phone). Again, we view the proposed general executive as currently occupying one point on this continuum: the multitasking being represented should be sensible but not necessarily highly optimized. One of our most important long-term goals involves extending the general executive to handle learning and optimization for particular sets of tasks — for instance, using ACT-R's production compilation mechanism (Taatgen & Lee, 2003) to create new "compiled" production rules for both tasks and integrate these tasks in an increasingly automated manner.

Acknowledgments

This work was supported by ONR grant N00014-03-1-0036 to the first author. We thank John Anderson, Christian Lebiere, Niels Taatgen, and Alex Chavez for their helpful comments and suggestions regarding this work.

References

Aasman, J. (1995). *Modelling driver behaviour in Soar*. Leidschendam, The Netherlands: KPN Research.

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., & Qin, Y. (in press). An integrated theory of the mind. *Psychological Review*.

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

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

Chong, R.S. (1998). Modeling dual-task performance improvement: Casting executive process knowledge acquisition as strategy refinement. Doctoral Dissertation, University of Michigan.

Jones, R.M., Laird, J.E., Nielsen P.E., Coulter, K., Kenny, P., & Koss, F. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine*, *20*, 27-42.

Kieras, D.E., Meyer, D.E., Ballas, J.A., & Lauber, E.J. (2000). Modern computational perspectives on executive mental processes and cognitive control: Where to from here?. In S. Monsell & J. Driver (Eds.), *Control of Cognitive Processes: Attention and Performance XVIII* (pp. 681-712). Cambridge, MA: MIT Press.

Lee, F.J., & Anderson, J.R. (2001). Does learning of a complex task have to be complex? A study in learning decomposition. *Cognitive Psychology*, *42*, 267-316.

Meyer, D.E., & Kieras, D.E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review*, *104*, 3-65.

Meyer, D.E., & Kieras, D.E. (1997b). A computational theory of executive cognitive processes and multiple-task performance: Part 2. Accounts of psychological refractory period phenomena. *Psychological Review*, *104*, 749-791.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Salvucci, D.D. (2001). Predicting the effects of in-car interface use on driver performance: An integrated model approach. *International Journal of Human-Computer Studies*, *55*, 85-107.

Salvucci, D.D. (2001b). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, *1*, 201-220.

Salvucci, D.D., Boer, E.R., & Liu, A. (2001). Toward an integrated model of driver behavior in a cognitive architecture. *Transportation Research Record*, 1779, .

Salvucci, D.D., & Macuga, K.L. (2002). Predicting the effects of cellular-phone dialing on driver performance. *Cognitive Systems Research*, *3*, 95-102.

Sohn, M.-H., & Anderson, J.R. (2001). Task preparation and task repetition: Two-component model of task switching. *Journal of Experimental Psychology: General*, 130, 764-778.

Taatgen, N.A., & Lee, F.J. (2003). Production compilation: A simple mechanism to model complex skill acquisition. *Human Factors*, *45*, 61-76.