Cognitive Science 29 (2005) 457–492

A Multitasking General Executive for Compound Continuous Tasks

Dario D. Salvucci

Department of Computer Science, Drexel University

Received 3 May 2004; received in revised form 31 July 2004; accepted 9 November 2004

Abstract

As cognitive architectures move to account for increasingly complex real-world tasks, one of the most pressing challenges involves understanding and modeling human multitasking. Although a number of existing models now perform multitasking in real-world scenarios, these models typically employ *customized executives* that schedule tasks for the particular domain but do not generalize easily to other domains. This article outlines a *general executive* for the Adaptive Control of Thought–Rational (ACT–R) cognitive architecture that, given independent models of individual tasks, schedules and interleaves the models' behavior into integrated multitasking behavior. To demonstrate the power of the proposed approach, the article describes an application to the domain of driving, showing how the general executive can interleave component subtasks of the driving task (namely, control and monitoring) and interleave driving with in-vehicle secondary tasks (radio tuning and phone dialing).

Keywords: Multitasking, Cognitive architectures, ACT–R, Driving

1. Introduction

As theories of cognition have matured over the last decade, they have increasingly broadened in scope beyond simple laboratory tasks to a wide variety of complex, dynamic task domains. In particular, the instantiation of cognitive theories as unified computational cognitive architectures (e.g., Anderson et al., 2004; Just, Carpenter, & Varma, 1999; Meyer & Kieras, 1997; Newell, 1990) has facilitated application and validation of these theories in real-world domains; for instance, researchers have used cognitive models to better understand human behavior in such domains as air traffic control (e.g., Taatgen & Lee, 2003), human–computer interaction (e.g., Kieras, Wood, & Meyer, 1997; Ritter, Baxter, Jones, & Young, 2000), game playing (e.g., Lebiere, Gray, Salvucci, & West, 2003), even aircraft piloting (Jones et al., 1999). Such modeling efforts for complex tasks have brought to light many new challenges for

Requests for reprints should be sent to Dario Salvucci, Department of Computer Science, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104. E-mail: salvucci@cs.drexel.edu

cognitive architectures, and one of the most critical yet elusive challenges has been the modeling of general human multitasking—how people integrate and perform multiple tasks in the context of a larger complex task.

A number of models developed in cognitive architectures have now emerged that, either explicitly or implicitly, account for aspects of human performance while multitasking. Table 1 shows a sampling of multitasking models classified into four broad categories as identified by Kieras, Meyer, Ballas, and Lauber (2000). Models of *discrete successive tasks* examine performance in alternating trials (or series of trials) of simple choice-reaction tasks, accounting for

Table 1
Examples of multitasking models developed in a cognitive architecture

Domain	Architecture(s)	Reference
Discrete Successive Tasks		
Alternating choice	ACT–R	Altmann & Gray, 2000
Alternating choice	EPIC	Kieras et al., 2000
Alternating choice	ACT–R	Sohn & Anderson, 2001
Discrete Concurrent Tasks		
Dual choice	EPIC	Meyer & Kieras, 1997
Dual choice	ACT–R	Byrne & Anderson, 2001
Dual choice	ACT–R	Anderson, Taatgen, & Byrne, in press
Elementary Continuous Tasks		
Tracking and choice	EPIC	Kieras & Meyer, 1997
Tracking and choice	EPIC–SOAR	Chong, 1998
Tracking and choice	SOAR, EPIC	Lallement & John, 1998
Compound Continuous Tasks		
Air traffic control (KA-ATC)	ACT–R	Taatgen & Lee, 2003
Air traffic control (AMBR)	ACT–R, D-COG, EPIC–SOAR, iGen	Gluck & Pew, in press
Aircraft maneuvering	ACT–R	Gluck, Ball, Krusmark, Rodgers, & Purtee, 2003
Aircraft piloting (TacAir–SOAR)	SOAR	Jones et al., 1999
Aircraft piloting	CI/ADAPT	Doane & Sohn, 2000
Aircraft taxiing	ACT–R	Byrne & Kirlik, 2005
Driving	SOAR	Aasman, 1995
Driving	QN-MHP	Tsimhoni & Liu, 2003
Driving and phone dialing	ACT–R	Salvucci, 2001
Dynamic systems	ACT–R	Schoppek, 2002
Game playing (Quake)	SOAR	Laird & Duchi, 2000
Game playing (Unreal Tournament)	ACT–R	Best & Lebiere, 2003
Radar operation (Argus Prime)	ACT–R	Gray & Schoelles, 2003
Shooting and mathematical comprehension	ACT–R, IMPRINT	Kelley & Scribner, 2003
Tactical decision making and instruction following	ACT–R	Fu et al., 2004
Tracking and decision making	EPIC	Kieras & Meyer, 1997

Note. ACT–R = Adaptive Control of Thought–Rational; EPIC = Executive-Process Interactive Control; D–COG = Distributed Cognition; CI/ADAPT = Construction-Integration/ADAPT; QN–MHP = Queuing Network—Model Human Processor.

the temporal costs of switching from one task to another. Models of *discrete concurrent tasks* analyze performance in concurrent choice-reaction tasks typically offset by a short delay, accounting for "psychological refractory period" effects of dual-task interference. Models of *elementary continuous tasks* address behavior when integrating a continuous task with occasional short discrete tasks—for instance, performing a manual tracking task (e.g., keeping a cursor on target) while occasionally responding to a choice-reaction task. These first three categories all include at least one discrete choice-reaction task that lasts at most a few seconds. In contrast, models of *compound continuous tasks* account for behavior in two or more simultaneous tasks, each of which is an ongoing continuous process or at least takes enough time to require interleaving with other tasks. Compound continuous tasks abound in real-world domains, particularly in the space of complex dynamic tasks that have become a centerpiece of cognitive modeling, and serve as the primary focus of this article.

Models of compound continuous tasks, such as those highlighted in Table 1, have begun to elucidate important aspects of management and coordination of multiple component tasks. For instance, Jones et al.'s (1999) fighter piloting model can identify new aircraft on radar while flying to intercept other aircraft; Lee and Anderson's (2000) air traffic control model can accept and land planes while monitoring landing conditions such as weather; and Fu et al.'s (2004) model can listen to and interpret "over-the-shoulder" instructions while performing a tactical decision-making task. Typically, analysis and validation of the model's behavior focuses on aggregate performance in the task, much of which can be affected by multitasking requirements; for instance, Kieras and Meyer's (1997) model of manual tracking demonstrated larger tracking errors in the presence of more difficult (i.e., more visually eccentric) choice tasks, and my own model of driving and phone dialing (Salvucci, 2001b) exhibited adverse effects of dialing on overall steering performance. In addition, a few models have been analyzed specifically for how and when participants switch between component tasks; for example, Gray and Schoelles's (2003) radar-operator model demonstrated how even models that accurately capture aggregate performance can miss important aspects of task-switching-specific measures, such as how often people switch at unit-task boundaries. All these efforts contribute to a broader understanding of multitasking through study of both overall measures of task performance and particular measures of multitasking performance.

Even with this wide diversity of models and domains, current models of compound continuous tasks share one limitation: the use of *customized executives* (Kieras et al., 2000), namely, multitasking control mechanisms that have been specialized and fine-tuned for a specific model or domain or both. Fig. 1 illustrates how a customized executive unifies two individual task models into an integrated model that performs both tasks. Given two models, A and B, the customized executive is designed to integrate these two specific models, resulting in a specialized executive for the integrated model A + B. Such an executive generally does not transfer to new models C and D, thus requiring a new customized executive for the integrated model C + D. In addition, model integration with a customized executive often requires modification to the component models themselves (illustrated by A′, B′, etc., in the figure), because one model typically explicitly passes control to the other model and must thus be aware of this model (and likely some of the declarative structures associated with it). The lack of transfer and need for component-model changes means that every domain requires a new approach and new models

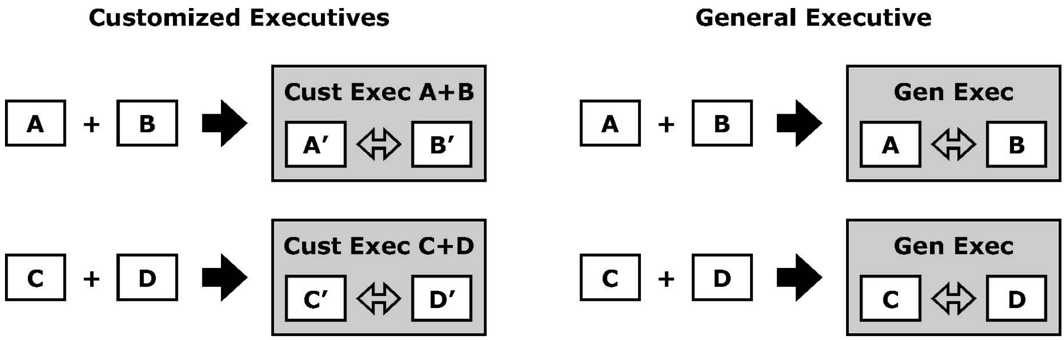

Fig. 1. Conceptual overview of customized executives and a general executive.

of multitasking, hampering both theoretical consistency among models of different domains as well as practical development through model reuse and transfer.

In the search for a general theory and model of human multitasking, we strive to move beyond customized executives to a unified *general executive*. As illustrated in Fig. 1, a general executive integrates separate models of task behavior and predicts the behavior that results from combining these models into a more complex, multitasking integrated model. Ideally, the general executive applies across all domains, providing us with a domain-independent theory of multitasking and an associated computational model of how multitasking takes place. Also, the general executive requires no modifications to the component models: Just as people can learn individual skills independently and then integrate these skills into multitasking behavior, the general executive can take models developed or learned independently of one another and unify these models into a psychologically plausible integrated multitasking model. (Subsequent learning could then specialize the component models or the executive itself for better multitasking performance; for now, we ignore such potential learning, although we address several issues related to learning later in the article.) Thus, rather than a collection of customized executives each specialized to particular domains, the general executive posits a broader theory that helps to unify our understanding and modeling of human multitasking.

This article describes a first step toward this general theory of multitasking, namely a formulation of a general executive for compound continuous tasks framed in the context of the ACT–R (Adaptive Control of Thought–Rational) cognitive architecture (Anderson et al., 2004). The general executive centers on a temporally aware queuing mechanism that manages current task goals and posits that goal representations are guided in part by reasonable heuristics of multitasking behavior. The article then demonstrates an application of the general executive to the domain of driving—a very rich, complex, ubiquitous task that millions of people perform on a daily basis. This application illustrates how the general executive accounts for driver multitasking both in integrating component subtasks of driving itself—namely, control and monitoring—and in integrating driving with a secondary task, namely, tuning a radio and dialing a cellular phone. Although the driving domain does not itself suffice as a complete validation of the general executive, it does serve as an excellent representative of compound continuous tasks (such as those in Table 1) that require management and scheduling of multiple processes in a complex environment.

2. A multitasking general executive for the ACT–R cognitive architecture

The development of a general executive and its instantiation in the ACT–R architecture serve as both a theoretical and practical endeavor. The primary motivation for the general executive is a cognitively plausible theory of multitasking that, like all aspects of a cognitive architecture, obeys the limitations and constraints of human cognition and performance. A secondary goal of the general executive is the facilitation of model integration and prediction—that is, the integration of independent individual task models into unified models and subsequent prediction of the effects of one component task on the others. This section begins with a description of the guiding theoretical principles on which the proposed general executive is based. It then continues with the specification of the executive in the ACT–R architecture, previously outlined in a preliminary report (Salvucci, Kushleyeva, & Lee, 2004) and generalized here.

2.1. Guiding principles for the general executive

Although our understanding of a multitasking general executive is very much incomplete, empirical and modeling research in this and related areas have begun to flesh out guiding principles for a possible executive model. This effort incorporates three guiding principles that helped to shape the general executive proposed here, as outlined in the following sections.

2.1.1. The general executive is an architectural mechanism

One possible approach to modeling multitasking in a cognitive architecture views multitasking as a learned cognitive skill like any other, and thus the mechanism should be instantiated as a model that abides by all architectural constraints rather than as a specialized mechanism. Such an approach requires that the component models in the integrated model somehow pass execution from one to another; one model may serve as a type of executive and help schedule and manage tasks, or all models may act as sibling processes and share execution time through a predetermined scheme. In either case, this view has at least two problematic issues that make it less plausible for a multitasking general executive. First, because models are explicitly shifting execution to others, each model must be aware of the other processes and must have special rules for passing control to these models. If models are learned or developed independently of one another, it is difficult to imagine how such specialized rules would evolve. Second, a general executive must have some form of interruption to ensure that running models are given a fair share of time, especially because, again, independently learned models would not necessarily know to give up control in a fair manner. If the general executive operates at the level of all other acquired skills, it is not clear how such a model would be able to interrupt other component models and shift execution to other models.

An alternate approach, and the one followed here, views the general executive as a specialized mechanism implemented within the cognitive architecture. This approach more easily addresses the issues mentioned previously: An architectural general executive would not require component models to be aware of one another, and the executive could certainly be made to interrupt the component models (leaving the deeper issue of when and how to interrupt). In addition, this approach seems more in line with neuropsychological evidence that suggests that general executive processes are located in different regions of the brain than procedural rule-based processes: Whereas prefrontal brain regions have been associated with maintenance of goal state (Koechlin, Corrado, Peitrini, & Grafman, 2000; Smith & Jonides, 1999) and the dorsolateral prefrontal cortex with goal memory and planning (Fincham, Carter, van Veen, Stenger, & Anderson, 2002), the basal ganglia have been associated with procedural memory and rule-based processes (Anderson et al., 2004). Thus, the general executive proposed here follows this view of the general executive as an architectural mechanism.

2.1.2. The general executive is dependent on time

When given a decision about what active task may proceed, the general executive must strike a delicate balance in allocating a fair amount of time to each active task. One way to achieve this balance is by selecting tasks on the basis of urgency: As a task is "starved" for resources (i.e., not allowed to proceed) for some period of time, its urgency steadily rises, thus making it more likely to be selected to proceed. To complicate this straightforward view, tasks in fact differ with respect to their desired running time—that is, not all tasks may wish to proceed immediately but rather can afford to wait some time before proceeding. As one example, in an air traffic control task where the wind direction changes every 30 sec, F. J. Lee (personal communication, April 19, 2005) found that participants were most likely to examine wind direction right around the 30-sec mark. As another example, Kushleyeva, Salvucci, and Lee (2005) showed that participants tended to switch from one visual search task to another at the halfway point of the full 30-sec task period to ensure they were not penalized on the other task. In both cases, people tended to perform the timed tasks not as soon as possible, but rather approximately at the time they deemed appropriate, given the temporal characteristics of the task.

Some of the best examples of the temporal dependence of multitasking arise in the many studies of driver distraction. Wierwille (1993) summarizes results from several studies, such as one by Dingus, Antin, Hulse, & Wierwille (1989), examining glance durations for a wide array of secondary tasks, including checking fuel, changing the radio station, and setting the vehicle defroster. In this study, the total number of glances inside the vehicle ranged from approximately one to seven glances, but the average duration per glance consistently fell between 0.6 and 1.6 sec. Thus, drivers exhibited an acute sense of how long they looked inside the vehicle, knowing they could safely look for a short time (approximately 1 sec) but that, as time passed, the urgency to return to driving steadily increased. Such temporal dependence appears in many real-world dynamic environments in which the passage of time corresponds with increasing uncertainty about the world. Thus, the general executive requires some formulation that can incorporate different task urgencies as part of its task management and scheduling.

2.1.3. The general executive is sensitive to goal representations

In addition to temporal issues of when to switch tasks, we can also ask how task and goal representations may influence multitasking—that is, are there "strongly connected components" of task procedures such that people are less likely to switch within a component and more likely to switch between components? If so, are these strongly connected components tied to representations of the goal structure? Empirical studies of extreme dual tasking, such as psychological refractory period studies mentioned earlier, suggest that with practice people can interleave tasks at the proposed smallest unit of cognitive cycle time, namely 50 msec (see, e.g., Card, Moran, & Newell, 1983; Meyer & Kieras, 1997; Newell, 1990). However, under general conditions of integrating multiple task processes in compound continuous tasks, studies suggest that task switching can indeed be highly influenced by goal representations and the "strongly connected components" of task procedures within these representations. For instance, in a radar operator task, Gray and Schoelles (2003) found that task switching most likely occurred between "unit tasks"; switching also occurred within unit tasks, but with lower probability. As another example, later in this article we examine the case of phone dialing while driving, where the data suggest that drivers update steering between the (3- and 4-digit) chunks of a phone number. Thus, a general executive should be flexible enough to allow goal and other declarative representations to influence task interleaving.

The principle of switching at natural representational boundaries has already been followed explicitly or implicitly by some existing models of compound continuous tasks with customized executives. For instance, Gluck et al.'s (2003) aircraft maneuvering model establishes new control settings under two specific conditions—at the beginning of a trial, and whenever the assessment of a control instrument shows large deviations from desired values—and thus implicitly defines the most natural switching points for aircraft maneuvering. Moreover, some modeling architectures themselves incorporate mechanisms to provide "robustness against interruption" to discourage or prevent preemption at critical points in processing (e.g., APEX: Freed, 1998). Thus, specific models and architectures have already recognized and begun to address the need for influences of goal representations on multitasking, and the general executive proposed here attempts to incorporate these ideas into a domain-independent mechanism for the ACT–R architecture.

2.2. An ACT–R general executive

The previously mentioned principles do not specify a general executive in their own right, but instead provide guidance for specifying a computational mechanism and for instantiating the executive within a cognitive architecture. The following section describes one possible instantiation of these guiding principles in the ACT–R cognitive architecture, with the overall goal of specifying a straightforward mechanism that accounts for several aspects of multitasking observed in real-world complex tasks, particularly for compound continuous tasks and the driving domain in the next section. Although this article focuses on the ACT–R architecture as the context for the general executive, the guiding principles and even aspects of the ACT–R instantiation that follows should generalize well to other architectures and modeling frameworks.

2.2.1. ACT–R and multitasking

The ACT–R cognitive architecture (Anderson et al., 2004; see also Anderson, 2005, this issue) is a production system architecture with a declarative memory store for factual chunks and a procedural memory store for condition–action production rules. The architecture incorporates a number of modules that interact with cognition by passing information through *buffers*—for instance, a visual buffer that holds the result of the vision module's visual encoding, or a motor buffer that passes movements to the motor module. One such buffer is the *goal buffer,* which stores this goal (itself a declarative chunk) and guides the production system to work on its associated task. Multitasking thus has a very straightforward interpretation in the architecture as the switching of goals in the goal buffer, thus allocating a period of cognitive processing time to one goal or task and then, after some time, switching to another goal or task.

Interestingly, the current ACT–R does not have a full-fledged goal module associated with the goal buffer; instead, new goals are managed and set directly through the actions of production rules. However, the previous version of the architecture (ACT–R 4.0, described in Anderson & Lebiere, 1998) incorporated a goal stack onto which new goals could be "pushed" and from which completed goals could be "popped." In essence, this goal stack served as one instantiation of a possible goal module, implicitly positing that people maintain goals and subgoals in a hierarchical stack representation. The goal stack was abandoned in the subsequent architecture due primarily to the psychological implausibility of such a stack because of memory limitations (Altmann & Trafton, 2002). The general executive described here can be viewed as a proposal for a new goal module, taking on the responsibility of maintaining and scheduling multiple goals and offloading this responsibility from the individual task production rules.

2.2.2. Architectural module for the general executive

The proposed ACT–R general executive represents a new goal module that maintains and schedules a set of active goals. In the current ACT–R, production rules set new goals by creating a declarative goal chunk and storing this chunk in the goal buffer, effectively replacing the old goal with the new one—for instance (in pseudocode approximating ACT–R production rules):

Do-Next-Task
IF the goal is to perform Task0
and this task is complete
THEN replace the current goal with a goal to perform Task1

In contrast, the proposed ACT–R general executive allows for addition or removal of goals to or from a set of active goals—for example,

Do-Multiple-Next-Tasks
IF the goal is to perform Task0
and this task is complete
THEN remove the current goal
and add new goals to perform Task1 and Task2

In adding both new goals to the active set, the goal module is now charged with managing both goals and overseeing their execution. This policy of addition and removal still allows for goal replacement by simply removing this goal and adding a new one; at the same time, the new policy provides the additional feature of allowing a model to have and manage multiple goals.

Even with multiple active goals, the ACT–R architecture still posits only one goal buffer and thus can only perform one goal at a time. This being the case, how does the module determine which active goal should proceed next? For this purpose, the proposed general executive employs a *goal queue* that operates on a first-come, first-served (FCFS) basis: Whenever rules add new goals, the goals are placed on the goal queue and "served" (i.e., allowed to proceed) in the order in which they arrive at the queue. Queues have been studied in detail in the applied context of computer and networking systems and the more theoretical context of general queuing theory (see, e.g., Nelson, 1995). However, queues have also been studied as psychologically based constructs for modeling visual information processing (Ellis, Goldberg, & Detweiler, 1996) and general behavior (Miller, 1993), along with conceptual relatives such as cascade models (e.g., McClelland, 1979). Of note, Liu (1996) provided a detailed discussion of queuing networks of elementary cognitive processes, including the assumption of FCFS single-channel processing. The general executive proposed here follows, albeit in a simplified way, in this tradition with its integration of a queue as a basic psychological mechanism.

To expand on the production rule example previously mentioned, Task1 and Task2 are added simultaneously, and thus let us assume that Task1 proceeds by random chance (to be clarified in a later section). Also, let us assume that these goals are each associated with a single production that simply iterates one instance of this goal after another for purposes of illustration:

Iterate-Task1
IF the goal is to perform Task1
and this task is complete
THEN remove the current goal
and add a new goal to perform another iteration of Task1

Iterate-Task2
IF the goal is to perform Task2
and this task is complete
THEN remove the current goal
and add a new goal to perform another iteration of Task2

When Task1 proceeds by firing the *Iterate-Task1* rule, the rule creates a new Task1 goal and adds this goal to the queue. At this point, the queue contains the older Task2 goal added in the first firing and this newer Task1 goal added at the second firing, and thus the executive chooses the older Task2 goal to proceed. When Task2 proceeds by firing the *Iterate-Task2* rule and creating a new Task2 goal, the active Task1 goal now takes precedence and proceeds. Thus, the goals alternate positions in the goal queue, resulting in alternating firings of *Iterate-Task1* and *Iterate-Task2*.

This example represents a simplistic case in which each goal is associated with a single production firing before creating a new goal; in general, a goal will proceed for several rule firings before creating another goal. For this general case, the general executive selects a new goal and alters the goal buffer only when a rule firing creates a new goal or removes this goal—that is, whenever a rule firing modifies the goal queue in any way. A consequence of this is that a goal can proceed unhindered so long as it does not create or remove goals (forcing us to further consider the granularity of goal representations, as discussed shortly). Consider the example illustrated in Fig. 2(a), which assumes that Task1 requires two rule firings for a total of 100 msec on every iteration, and Task2 requires three rule firings for a total of 150 msec on every iteration. Assuming Task1 proceeds first, the Task1 rules fire and complete the first iteration, with the final rule firing creating a second Task1 goal. At this point, the general executive interrupts Task1 and allows the first Task2 goal to proceed until completion. This Task2 goal creates a second Task2 goal, allowing the second Task1 goal to intercede. Thus, like the earlier example,

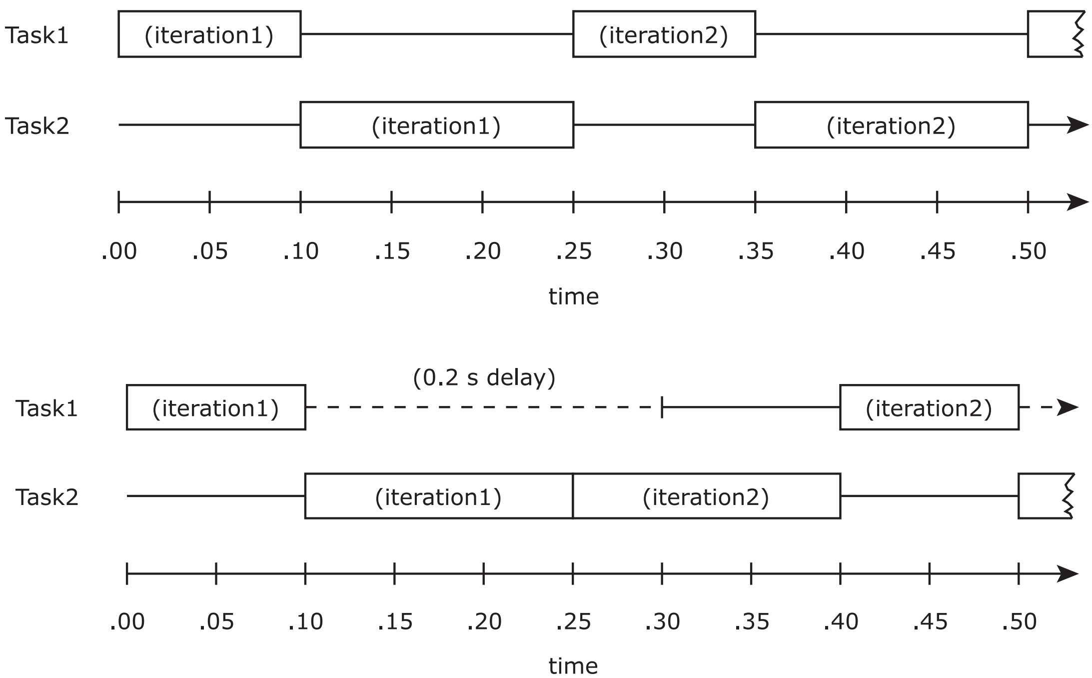

Fig. 2. Timeline for two tasks with (a) no delay on either task, and (b) a 0.2 sec delay for *Task1*. Each box indicates the time span during which the given goal would be executed.

the general executive alternates between goals for the two tasks, but only interrupts a task when it creates a new goal.

2.2.3. Incorporation of temporal dependence

The FCFS queue already incorporates one type of temporal dependence, namely, that goals are allowed to proceed in the temporal order in which they are created. However, as noted earlier, it is desirable that the general executive also addresses the fact that some tasks do not require immediate processing but rather wish to or can afford to be delayed for some period of time. To this end, the basic goal queue is augmented by allowing goals to specify a desired *delay* relative to the current time—for instance,

Iterate-Task1
IF the goal is to perform Task1
and this task is complete
THEN remove the current goal
and add a new goal to perform another iteration of Task1
after a delay of 0.2 sec

The delay defaults to a value of 0 if not specified (i.e., wishes to run immediately), as in the *Iterate-Task2* production earlier, and also allows for a special *now* value that triggers immediate execution of this goal regardless of the state of the goal queue. For our purposes, we assume that the delay can be up to a few seconds into the future; prospective goals that are minutes, hours, or more in the future involve a number of other issues, such as retrieval failures, and are beyond the scope of this article.

To incorporate this temporal dependence into the goal queue, each goal in the queue is associated with a desired start time, specifically the goal's time of creation plus any specified delay. (Note that for goals with a default delay of 0, the desired start time is equal to creation time, indicating their desire to start immediately if possible.) When the production system adds or removes goals, and the goal module reevaluates the goal queue, the module chooses the goal with the earliest start time and allows this goal to proceed. In essence, this scheme implements a slightly modified FCFS queue: If all goals have a default delay of 0, the goal queue operates as a standard FCFS queue, but the queue also allows for nonzero delay times that reorder the queue and account for delayed goal execution. Equivalently, this scheme can be formulated as using urgency for scheduling: Each goal has an urgency defined as the negation of the remaining time before the goal wishes to run (i.e.,–*delay*), and the general executive chooses the goal with the highest urgency as the next to proceed.

Fig. 2(b) illustrates the delay mechanism used in conjunction with our earlier example: the Task1 and Task2 rules remain the same except that the Task1 final rule specifies a delay of .20 sec before the next Task1 goal. As before, the first Task1 goal proceeds until completion at time .10 sec, then creates a new goal with a delay of .20 sec , thereby requesting a desired start time of .30 sec. The first Task2 goal intercedes as before, but this time on completion, the next Task2 goal has an earlier start time (.25 sec) than the next Task1 goal (.30 sec). Thus, the second Task2 goal proceeds until completion, at which point the second Task1 goal has waited .10 sec and is finally allowed to proceed.

Given the goal queue's reliance on timing information and the fact that people have an imperfect perception of time, the mechanism also includes systematic noise in the goal queue ordering to incorporate limited stochasticity to the process. Several aspects of the ACT–R architecture (e.g., chunk activation, production-rule conflict resolution) already incorporate noise, specifically a logistic distribution of noise with variance $s^2 = p^2 s^2 / 3$ driven by a noise parameter s. To maintain consistency with the architecture, the goal queue includes logistic noise around the start times of the individual goals, along with a parameter s_{gq} (*gq* for "goal queue") to specify the amount of this noise. Thus, the previous example indicates only one possible scheduling of goals and tasks, but this scheduling may change, given larger values of temporal noise in the process. Also, the noise results in essentially a random choice between goals with the same desired start times, such as goals created by the same production rule with the same delay time.

2.2.4. Heuristics for goal representations

As mentioned, one of the key issues with any multitasking general executive arises in the delicate balance between allowing a task to proceed versus preemptively interrupting the task to allow other tasks to proceed. The proposed general executive strikes this balance by potentially interrupting and switching tasks *between* goals without interrupting *within* goals. In essence, this strategy is motivated by the earlier discussion of the influences of goal representations on multitasking, finding "natural" breaking points in which to interleave other tasks. In theory, this strategy is sufficient for interleaving any number of ACT–R models and exhibiting multitasking behavior. In practice, however, models may be developed or learned in such a way that their goal representations do not allow reasonable multitasking (i.e., a kind of "impolite" goal representation)—as a simple example, models that only execute a single goal for minutes or hours on end. To this end, this work proposes two heuristics that attempt to further constrain goal representations for facilitation of multitasking and task interleaving:

- The iterating heuristic. Models of complex tasks typically run not for a short burst of time but rather for an extended period of time, implying that at some point these models iterate or repeat sections of their procedures. (For instance, 100 sec of task execution at 50 msec per rule would require 2,000 rule firings; even complex models rarely have near this many rules, but rather iterate on a core set of rules.) The iterating heuristic posits that at any point in which the model iterates on part of its rule set (i.e., by returning to a previously fired rule), the model should allow for task switching at this point by creating a new goal to perform the next iteration of the procedure (as opposed to resetting the state of the current goal).
- The blocking heuristic. Most models of complex tasks involve some amount (if not a great deal) of perceptual–motor activity, and in some cases the production system is forced to wait for a perceptual–motor action to complete before proceeding. The blocking heuristic states that when a model expects to wait a "significant time" for the completion of an action—for example, a hand movement from keyboard to mouse—the model should create a new goal for the next action, thus allowing another task to intercede during the waiting period. At this time, our work unfortunately cannot provide a clear definition of "significant time"; it is very possible that time periods deemed "significant" depend closely on the task at hand and the urgency of all active tasks. Nevertheless, this heuristic has been used in previous work on driver distraction—for example, allowing steering to intercede during a hand movement from the steering wheel to a secondary task device (Salvucci, 2001b)—and, even as an underspecified heuristic, can provide guidance for the granularity of goal representations while modeling.

To emphasize, these heuristics are not intended to be comprehensive or exhaustive, nor do they fully specify a goal representation for a given task. Instead, the heuristics are intended to constrain the space of possible goal representations for tasks such that individual task models integrate well with the general executive. In other words, the proposed approach to multitasking can be viewed as an integration of both an executive control module and a set of general guidelines for modeling individual tasks. The studies of driver multitasking in the next section will illustrate these ideas and also contain several examples of the use of these heuristics to guide modeling of individual tasks.

2.2.5. Summary of the general executive

In essence, the general executive can be characterized by four core ideas:

1. Multiple goals can be "in play" at one time, all competing for execution time on the cognitive processor (unlike current ACT–R theory).
2. However, only one goal can be executed by the cognitive processor at any given time (like current ACT–R theory).
3. Two heuristics guide when to switch away from this goal: (a) switch after completing an iteration of the procedures that achieve the goal, and (b) switch after requesting a perceptual or motor action that requires substantial waiting time.

4. When switching away from this goal, the general executive switches to and executes the most urgent goal (i.e., the goal most due or overdue). Goals can specify when subsequent goals become due on creation of these new goals.

Although this article centers on an instantiation of the general executive in the ACT–R cognitive architecture, the core ideas should generalize well, in whole or at least in part, to other architectures and cognitive theories.

3. Studies in driver multitasking

As described earlier, our previous successful efforts in modeling driver behavior in the ACT–R architecture all relied on customized executives for driver multitasking. Given the proposed general executive, this section describes three studies in which the mechanism is applied to integrating the driving subtasks of control and monitoring (Study 1), integrating driving with the secondary task of tuning a radio (Study 2), and integrating driving with the secondary task of dialing a phone number (Study 3). These studies have two overarching goals. First, the studies aim to replicate some of the findings in our previous work to ensure that the general executive can account for some of the same aspects of driver behavior as did the customized executives in previous models. Second, the studies attempt to extend previous work by accounting for new results related more directly to driver multitasking—specifically, results that elucidate when and how drivers switch tasks.

3.1. Study 1: Control and monitoring

The task of driving actually comprises a number of component subtasks, all critical to safe driving. Perhaps the most obvious and recognizable component subtask is that of *control*: lateral control (i.e., steering) to maintain a central position in the lane, and longitudinal control (i.e., acceleration and braking) to maintain a safe speed, distance, or time headway from nearby hazards. Control clearly requires attention and encoding of the visual environment, with the focus of attention centered on the lead vehicle or upcoming segments of the lane or both. However, safe driving requires additional situation awareness of surrounding vehicles (or other obstacles), bringing up a second critical subtask of *monitoring*. Although monitoring is not as critical as control in an immediate sense, it provides the driver with knowledge of her or his surroundings, thus enabling decision making or emergency maneuvers when necessary (e.g., normal or emergency lane changes); thus, the more time that can be devoted to monitoring, the more accurate the driver's mental model of the immediate surroundings. This first study explores the integration of control and monitoring in a multilane highway environment, such as that shown in Fig. 3, allocating as much attention as possible to monitoring without detracting from the immediate task of safe control.

3.1.1. Empirical study

The human driver data with which to validate the model come from the empirical validation of the original driver model (Salvucci, Boer, & Liu, 2001). This study was conducted in a

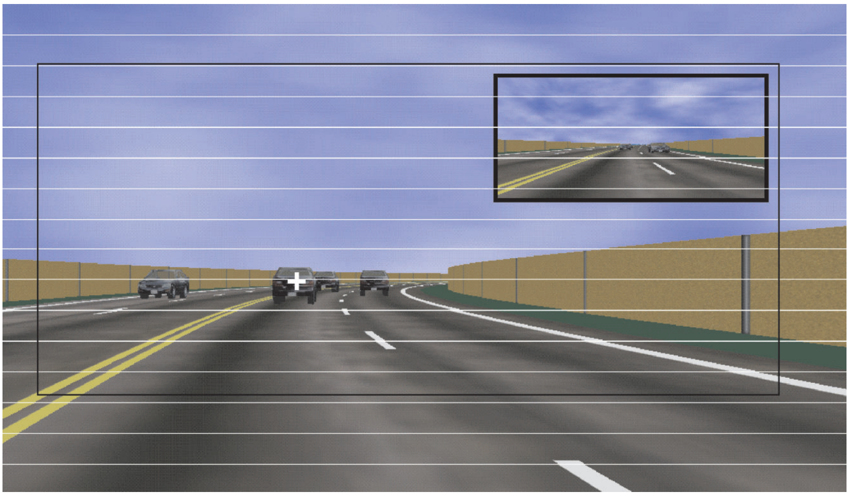

Fig. 3. Sample screen shot of the multi-lane highway driving environment.

fixed-base driving simulator with a simulated multilane highway environment and moderate (automated) traffic. The human data comprise a total of 11 participants driving in this environment and generating a total of 311 km of driving data. The resulting data protocols contain a detailed snapshot of the environment sampled at roughly 13 samples/second; each sample includes information about the driver's controls (steering angle, pedal depression, etc.), the driver's vehicle position (lateral position in lane, etc.), and the position of all other vehicles in the environment. Critically, the study included the collection of eye-tracking data that sampled the driver's point of gaze during navigation; these gaze data are crucial to elucidating the interaction of control and monitoring and the switching between them.

3.1.2. Model development

The component models of control and monitoring come from the most recent driver model (Salvucci, in press; an update of the original model in Salvucci et al., 2001). The control model is based on a straightforward perception–action control law (Salvucci & Gray, 2004) that updates steering based on the perceived visual angle to two points: a *near point* directly in front of the vehicle that guides position in the lane, and a *far point* in the distance (e.g., the vanishing point of a straight road or the tangent point of a curved road) that guides steering into the upcoming roadway. This control law works well for both straight and curved roads and also generalizes easily to lane changes. A similar control law updates the depression of the accelerator and brake pedals based on time headway to the lead car. These control laws are incorporated into a set of ACT–R production rules, shown in Table 2, that visually encode the near and far points, compute the necessary values, and generate the motor actions for the steering wheel and pedals. Although many other control models use continuous mathematical functions or control-theoretic approaches (e.g., Hess & Modjtahedzadeh, 1990; Hildreth, Beusmans, Boer, & Royden, 2000), the ACT–R control model, due to its implementation as a production system, necessitates discrete updates of control. Thus, the model updates control at a periodic rate dictated by the number of productions fired (given ACT–R's 50-msec firing rate), and the re-

Table 2
Control and monitoring goals and production rules

Control
Attend-near: locate near point (if necessary)
Process-near-attend-far: note near point information, locate far point
Process-far/car: steer and accelerate for road point or lead car as appropriate
Done-unstable: if unstable, locate near point and set immediate goal of Control
Done-stable: if stable, set new goal of Control with delay $D_{control}$
Monitor
*Monitor-lane**: choose random lane and locate if vehicle present
Done-process-car: note vehicle information, set new goal of monitor
Done-no-car: set new goal of monitor

Note. Special-case rules that do not affect multitasking have been omitted.

sulting rule set generates updates at a rate of one update per 150 to 200 msec (200 msec for the first iteration after switching to control, 150 msec for subsequent iterations until switching away).

The monitoring model randomly samples the visual environment, which in this case is a two-lane highway with moderate traffic. On each iteration, the model randomly chooses, with equal probability, a lane (left or right) and direction (forward or backward) in which to glance for other vehicles; forward glances are simply through the main view, and backward glances are directed to the rearview mirror. If a vehicle is found, the model notes its lane and direction as well as distance from the driver's vehicle; this information can then be used when deciding whether to change lanes and can, for instance, help to recall a vehicle in the "blind spot" even if the vehicle cannot presently be seen. One iteration of this monitor goal requires 100 msec for the firing of two production rules, one that selects and finds a visual object for a chosen lane and direction, and one that notes the presence or absence of a vehicle in a declarative chunk. The monitoring model's goals and production rules are included in Table 2.

The integration of the control and monitoring models is a straightforward process with the general executive. To initiate driving, the model includes one production that starts both the control and monitor goals. After this point, each task is scheduled and executed by the general executive with no explicit knowledge of the other task. Not surprisingly, there is some communication between the tasks with respect to the monitoring information in declarative memory: When the driver's vehicle approaches the lead vehicle within a certain time headway, the control goal initiates an attempt at a lane change, and this decision cycle can retrieve memories of past monitor goals to check for other vehicles that may hinder the lane change. Nevertheless, the processes do not communicate with respect to multitasking, leaving the work of switching between tasks completely to the general executive.

For both the control and monitoring models, the iterating heuristic applies to the representation of goals: In both cases, the model iterates by creating a new goal of the same type to repeat the cyclic process. By default, the integrated model then would alternate between control and monitoring, interleaving one iteration of one goal with one of the other as illustrated in Fig. 2(a). However, we might expect that the switching between tasks would be at least somewhat dependent on the situation: In situations of difficult control, the control task would dominate and perhaps keep control for a longer time; in situations of easy control or a stable environment, the control task could allow monitoring to occupy more time, given that the more time taken by monitoring, the better the overall situation awareness (e.g., longer looks to estimate velocity rather than simple position). To this end, the model introduces two ways to quantify these factors. First, it includes a *control stability threshold* that indicates how "stable" the external environment should be before the control goal gives up control. The determination of stability, as defined in the most recent driver model (Salvucci, in press), measures whether the vehicle's lateral position is "close enough" to the lane center and whether its lateral velocity is "stable enough" and not moving too quickly side to side; specifically, the three control values that determine lateral position and velocity (near-point position, near-point velocity, and far-point velocity) have constant-value thresholds that, when below all thresholds, define the vehicle as stable. When the environment is not stable, the model performs the next control iteration immediately by setting the delay time to the special *now* value. When the environment is stable, the model includes a *control delay time* that indicates how long the vehicle can go without control until it would be necessary to return. Thus, the interleaving of control and monitoring in fact resembles Fig. 2(b), with control behaving like *task1* in the figure, except that the control delay time will be estimated to best account for driver behavior.

A total of three parameters were estimated for this integrated model. One parameter is associated with the general executive, namely the s_{gq} noise parameter; this was estimated at .075 and kept constant for this study and the two studies that follow. The other two parameters relate to the control model: the control delay time $D_{control}$, estimated at 500 msec and kept constant across studies; and a control stability threshold F_{stable} that scaled the threshold in the original driver model by a constant factor, estimated at .71. For analysis, model predictions were collected by running three 10-min simulations in the same multilane highway environment. Because the same environment was used for human data collection and model simulation, the model generates behavioral protocols identical to those of human drivers, and thus its behavioral data can be analyzed in exactly the same way. The model, like human drivers, also generates gaze data through ACT–R's vision module (see Byrne & Anderson, 2001) integrated with the EMMA module that translates movements of visual attention to observable eye movements (Salvucci, 2001a).

3.1.3. Results

3.1.3.1. Aggregate measures of task switching. The most informative indicator of task switching between control and monitoring arises in the form of driver gaze, noting where drivers direct their overt visual attention as eye movements to visual objects in the environment. One aggregate measure of task switching, then, examines the proportion of gaze time spent on different visual regions in the environment. Fig. 4 shows this measure for several regions in the environment: the near region of the road, the far region of the road, the lead car, and other cars (forward only) in this lane; these same regions in the other lane; the rearview mirror; and vehicles in the oncoming lanes. Gazes serving the function of control are represented by gazes to the current lane in the near region of the road, far region of the road, and lead car. The figure

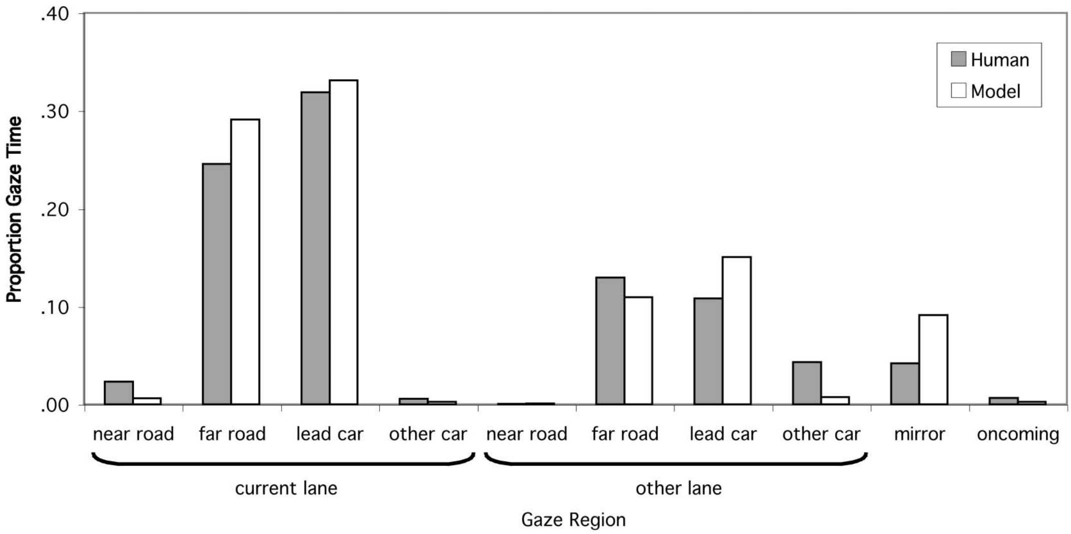

Fig. 4. Study 1: Aggregate proportion gaze times for visual regions in the multi-lane highway environment.

clearly shows that drivers directed the majority of their gazes to these regions, particularly the far road and the lead car. The model, using these points for control as well, exhibited very similar gaze times on these regions. Gazes serving the function of monitoring are represented by gazes to the other lane and the mirror. Drivers spent approximately half the time on these monitoring regions as compared to the control regions. Also, the drivers gazed at the rearview mirror roughly 5% of the time to monitor cars in back of the vehicle. Again, the model predicted very similar gaze times on these regions, albeit with slightly high predictions for the mirror. The default model used here assumed equal probabilities for forward and backward monitoring, so the addition of an extra parameter for weighting forward versus backward gazes could have more closely estimated the mirror gaze time. Nevertheless, the overall model fit to the human data remained quite good, $R^2 = .95$, $RMSE = .03$. Thus, the model nicely accounts for the aggregate effects of task switching on overall gaze time, specifically the amount of time spent on control versus monitoring gazes.

3.1.3.2. Detailed measures of task switching. Although these aggregate measures show some evidence of proper task switching, they do not speak to exactly when drivers switch between tasks; as pointed out by Gray and Schoelles (2003), models that capture aggregate measures of task switching do not necessarily correctly predict switching behavior at a lower level. To this end, we can examine when drivers switched between control and monitoring by measuring the probability of switching over time, where a "switch" is defined by a shift of gaze from the control visual regions to the monitoring visual regions and vice versa. To perform this analysis, all gazes were first classified as control or monitoring gazes, based on the regions necessary for their respective goals: Gazes on the same lane (forward only) were classified as control gazes; gazes on other lanes or in the rearview mirror were classified as monitoring gazes; and all other gazes were classified in a catch-all "other" category. Next, all one-sample gazes interrupting a continuous gaze at a particular region ("blips" in the eye-movement data, often from eye blinks or other data noise) were included in the larger gaze. Finally, the resulting gazes were grouped together in subsequences by task (control, monitoring, or other), and switch probability distributions over time were computed from these subsequences at ¼-sec intervals. This analysis excluded any samples that were part of a lane-change maneuver to remove any ambiguity of the meanings of "current" and "other" lane.

Fig. 5(a) shows the resulting switch probability distributions for monitoring, including the human data (solid line) and model predictions (dashed line). Human drivers were most likely to switch after 250 to 500 msec of monitoring, with a sharp drop-off thereafter and very few switches after 1 sec. It is important to note that the peak in this distribution is not simply a function of the time needed to visually encode another car's position (which can be done peripherally and would require 100 msec for ACT–R); drivers attend to the monitoring task as long as possible, knowing that longer looks can provide more information (e.g., velocity) and more accurate information. The model reproduced this distribution, $R^2 = .95$, $RMSE = .04$, primarily as a function of the control delay time in the control model: Because a stable control goal requests a delay time of 500 msec (as estimated in parameter fitting), the integrated model can run several iterations of the monitoring goal until this delay time expires and the model returns to control. Thus, the model may monitor several vehicles during this delay, or may even monitor the same vehicle for an extended period of time. The control delay time also results in very few switches after 1 sec, because the urgency to switch back to control increases as monitoring runs longer past the 500 msec delay time.

Fig. 5(b) shows the analogous distribution for control, and here we see a very different pattern: The highest switch probability for human drivers fell in the 0 to 250 msec range, and the probabilities dropped steadily from this point in a smooth manner. It was not uncommon for drivers to perform control for 1 to 2 sec before switching, indicating points of difficult driving during which the driver maintained interrupted control. The model nicely reproduced the overall trends in the data, $R^2 = .97$, $RMSE = .02$. In the model, the control goal relinquishes control only when the vehicle achieves the stability threshold, thus generating a number of long control times in the 1 to 2 sec range. However, usually the model required only a short period of control during stable driving, and thus the model, like human drivers, exhibited the highest switch probability in the shortest duration range.

3.2. Study 2: Radio tuning and driving

Although driving itself can be decomposed into subtasks, another common aspect of driver multitasking involves the integration of driving with some secondary task, such as the use of an in-vehicle device like a cellular phone or navigation device. The burgeoning use of such in-vehicle devices has begun to pose a serious safety risk: A recent study determined that driver distraction and inattention is now the leading cause of crashes, ahead of even speeding and alcohol (Hendricks, Freedman, Zador, & Fell, 2001). This second study explores the driver multitasking that occurs when integrating the primary driving task with the secondary task of tuning a radio device. The study is based on empirical data collected by Sodhi, Reimer, and Llamazares (2002) that examined driver eye movements while performing various secondary tasks.

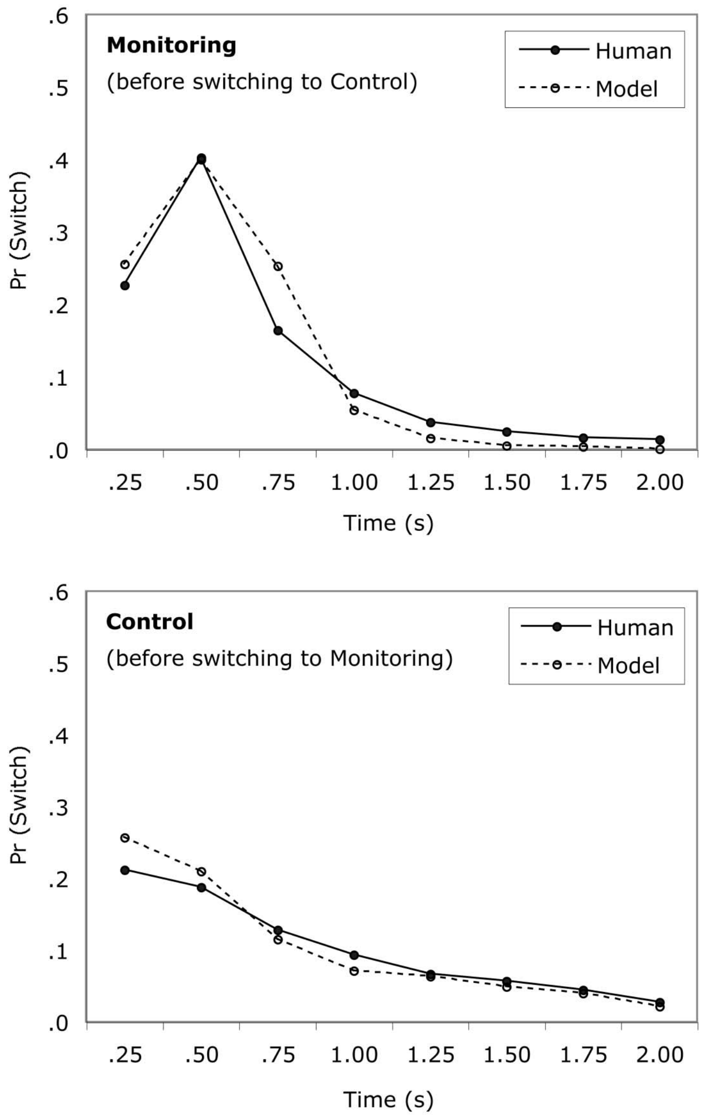

Fig. 5. Study 1: Probability of switching after a given time for (a) monitoring and (b) control.

3.2.1. *Empirical study*

In the Sodhi et al. (2002) study, participants drove an instrumented vehicle on a semirural road at a self-selected natural speed. Although driving, participants were occasionally asked to perform a variety of secondary tasks, including the one of interest to this study, namely, the radio-tuning task in which the participant turned on the radio and tuned the station to 1610 AM. Throughout the experimental drive, participants' eye movements were monitored using a head-mounted eye tracker sampling gaze direction at 50 Hz. These eye movements were subsequently processed and classified as glances either to the roadway (for the driving task) or the radio (for the secondary task). Sodhi et al. (2002) provided aggregate analyses as well as individual protocols illustrating drivers' behavior during the secondary tasks; this article focuses on the results that best illustrate the task-switching aspects of driver behavior, namely, time spent on one task before switching to the other as elucidated by the eye-movement protocols.

3.2.2. Model development

The component models for this study include the primary task of control and the secondary task of tuning a radio to a desired station. The control model comes, not surprisingly, directly from Study 1, with production rules and declarative representations simply ported over from the previous study into this study. However, because the driving environment differed in this study, the model was placed on a single-lane roadway with a single lead vehicle to help the model drive at a constant speed. It is important to note that the monitoring portion of the driver model became irrelevant in this environment—with no vehicles in other lanes and no possibility of lane changing—and thus the model only required the control component of the driving model used in the first study.

The radio-tuning model, derived from a task analysis of tuning a standard car radio, is outlined in Table 3, which lists the goals that make up the model as well as the individual production rules associated with each goal. Unfortunately, the original study did not specify the initial state of the radio, so the model assumes that drivers select a station in two steps: (a) holding down the station-advance button for 2 sec to jump approximately to the station, and (b) press-

Table 3
Radio-tuning goals and production rules

Tune-Move
Find-radio: locate radio (peripherally)
Encode-radio: fixate and encode radio
Move-to-radio: move right hand to radio, set new goal of Tune-Power-On
Tune-Power-On
Find-power-button: locate power button
Encode-power-button: fixate and encode power button
Press-power-button: press power button, set new goal of Tune-Start-Jump
Tune-Start-Jump
Find-advance-button: locate advance button
Encode-advance-button: fixate and encode advance button
Hold-advance-button: press advance button, set new goal of Tune-Monitor-Jump
Tune-Monitor-Jump
Find-display: locate station display
Encode-display: fixate and encode display
Hold-advance-button: before 2 sec, set new goal of Tune-Monitor-Jump
Release-advance-button: after 2 sec, release advance button and set new goal of Tune-Adjust
Tune-Adjust
Find-display: locate station display
Encode-display: fixate and encode display
Press-advance-button: press advance button, set new goal of Tune-Return
Tune-Return
Home-hand: move right hand to steering wheel, terminate goal

ing the advance button one last time to adjust for error in this jump and select the exact station. (These assumptions were made a priori before any simulations were performed and were not modified thereafter.) Referring to the table, the Tune-Move goal visually locates the radio and moves the right hand to its general location, then Tune-Power-On locates the power button and presses this button to activate the radio. Next, Tune-Start-Jump presses the station-advance button and Tune-Monitor-Jump iterates to watch the display and complete the jump after 2 sec. Finally, Tune-Adjust presses the advance button one last time, and Tune-Return moves the right hand back to the steering wheel.[2] The model incorporates the iterating heuristic in that the Tune-Monitor-Jump goal iteratively creates the same goal until the appropriate time. The model also incorporates the blocking heuristic in that each hand movement or button press is considered a significant event (requiring visual fixation and then movement), which must wait for the physical action to complete and thus allows another goal (i.e., control) to intercede.

As in Study 1, the creation of the integrated model for control and radio tuning becomes a straightforward task, given the general executive. On starting up, the model initially runs only a control goal to navigate the construction-zone environment. To initiate tuning, the model simply adds the tuning goal to the goal queue, thus allowing tuning to interleave with driving as dictated by the general executive. The general executive here uses the same noise value estimated in Study 1, and the control model uses the same control delay time (500 msec) as in Study 1. However, because the studies used different driving environments, we expect that human drivers exhibited different abilities or tolerances in the environments. To account for this, the model incorporated constant scaling factors for three sets of parameters, namely for the amount of steering change, the amount of pedal-depression change, and the control stability threshold; these scaling factors were estimated to produce the best fit to the empirical data for both Studies 2 and 3, with final values of 0.7, 0.4, and 2.5, respectively. The integrated model was run in three driving simulations with eight tuning trials per simulation, spaced 20 sec apart.

3.2.3. Results

In Study 1, the measure most illustrative of drivers' task-switching behavior was the probability of switching between tasks as indicated by eye-movement data. Fortunately, Sodhi et al. (2002) reported these same switch probabilities for drivers performing the radio-tuning task, and thus we can validate the integrated model's predictions through comparison with these data. Fig. 6(a) shows the switch probability distributions for radio tuning, including the human data (solid line) and model predictions (dashed line). The human drivers switched most often in the 0.6- to 1.0-sec range, with fewer switches in the first 0.0 to 0.4 sec, even fewer switches after 1.0 sec, and almost no switches after 1.6 sec. As before for monitoring, these distributions indicate a balance between keeping a gaze on the radio (primarily to confirm the changing display) and ensuring that control is able to intercede at regular intervals. The model nicely reproduced this switch probability distribution, $R^2 = .92$, $RMSE = .03$; the control delay time (kept constant between studies) allowed the model to concentrate on tuning initially, thus leading to a few quick switches before 0.4 sec, but also forcing the model to interrupt and switch to control as time passed. Switches around 0.6 to 1.0 sec were primarily due to the initial Tune-Move and Tune-Power-On goals running in sequence, without a switch in between, for a total duration of approximately 0.8 sec; switches around 1.4 sec were primarily due to the Tune-Adjust

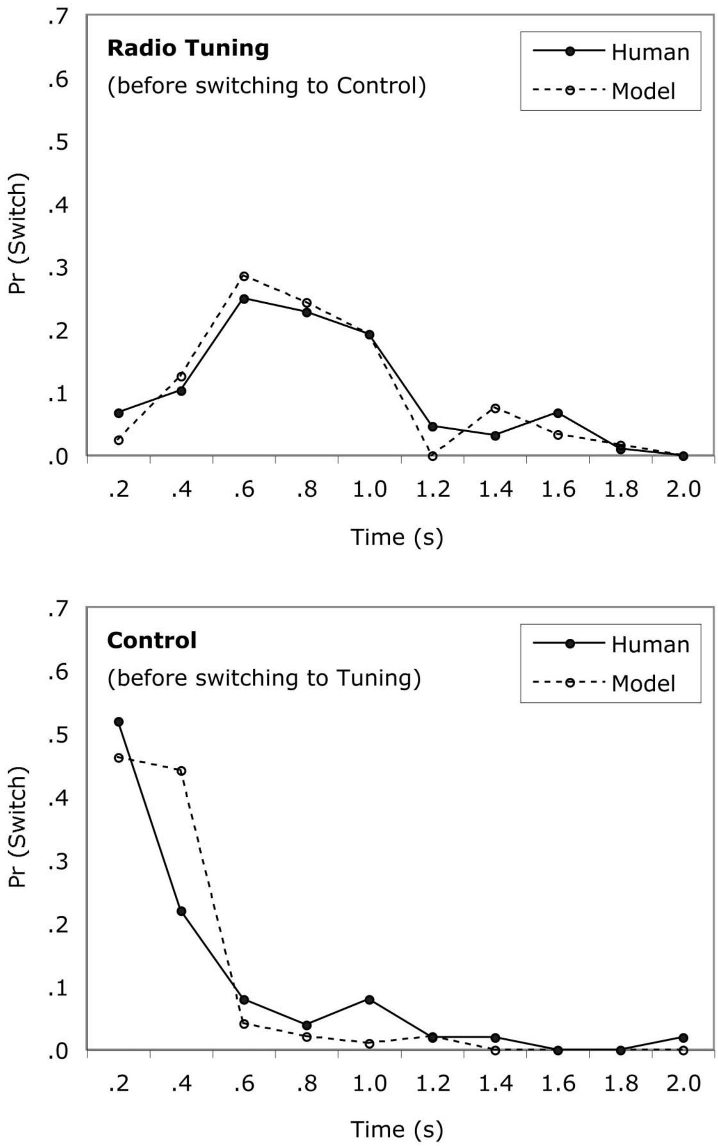

Fig. 6. Study 2: Probability of switching after a given time for (a) radio tuning and (b) control.

and Tune-Return goals running in sequence and waiting for the hand to return to the steering wheel before terminating the goal.

Fig. 6(b) shows the analogous distribution for control, with a pattern very similar to that in Study 1—namely, that the switch probability was rather high for short times and tailed off for higher times. The model again fit the data well, $R^2 = .81$, $RMSE = .08$, albeit with a discrepancy in the 0.2- to 0.4-sec range. Again like Study 1, the model under normal conditions often produced stable control with single control updates, leading to frequent short switch times; under less frequent, difficult driving conditions, or when the vehicle became unstable, the model focused on control and exhibited longer switch times.

3.3. *Study 3: Phone dialing and driving*

Study 3 also explores driver multitasking in the context of driver distraction. However, whereas Study 2 examined a highly visual radio-tuning task, Study 3 examines a primarily nonvisual task of dialing a familiar 10-digit phone number on a numeric keypad with no visual feedback. Several of our previous efforts have shown how the ACT–R driver model can predict various effects of phone dialing on driver performance (Salvucci, 2001b, 2002; Salvucci & Macuga, 2002). This study presents new empirical results that delve deeper into exactly when drivers interleave driving and dialing, and it also demonstrates how the general executive can reproduce these results by integrating separate models of the two tasks. The study uses the same highway environment as Study 2, namely, a realistic one-lane highway with a lead vehicle, with one exception: The lead vehicle accelerates and decelerates in an abrupt random manner similar to what might occur in a construction zone (see Salvucci & Macuga, 2002). As a result, we can analyze both the aggregate effects of multitasking on driver performance, and also the detailed effects of multitasking that elucidate when drivers switch between driving and dialing.

3.3.1. *Empirical study*

The human driver data for the driving and dialing task were collected from drivers navigating the construction-zone environment in a small-scale driving simulator, specifically a desktop system with a force-feedback Logitech Wingman® steering wheel and integrated pedals. In the experiment, each participant was first given general information about the driving environment and allowed 5 min of practice driving. Then, the participants wrote down four of their most familiar 10-digit phone numbers along with a name or phrase associated with each number. The experiment continued with three phases: pretest, main, and posttest. In the main phase of the experiment, participants drove in the environment and were occasionally asked to dial a phone number: The experimenter asked the participant to dial and simultaneously hit a special start key on the keyboard; the participant then dialed the number on the numeric keypad of the keyboard, and finally terminated the number by pressing *Enter* on the keypad. In the pretest and posttest, the same protocol was followed, but the simulation was turned off, and the participant had only to dial the number (without driving). Each participant performed a total of 32 dialing trials (8 for each of the four phone numbers) in the main driving test and 32 trials combined in the pretest and posttest (with 16 trials each). In all, data from 10 participants were collected for the study. Of these, 3 participants failed to dial 90% of the numbers correctly and were omitted from further analysis. The data from the remaining 7 participants include 51 km of driving data.

3.3.2. *Model development*

The component models for this study include the primary task of control and the secondary task of dialing a phone number. The control model was taken from Study 2, and because both studies use essentially the same driving environment, the model could be directly imported for this study with no changes to the model, including parameter settings.

Table 4
Phone-dialing goals and production rules

Dial-Move	
	Find-keypad: locate numeric keypad
	Look-at-keypad: fixate and encode keypad
	Move-to-keypad: move right hand to keypad, set new goal of Dial-Prepare
Dial-Prepare	
	Recall-first-block: retrieve first block
	Start-dial: set new goal of Dial-Block
Dial-Block	
	Recall-first-number: retrieve first number of block
	Type-number-recall-next: type current number, recall next number in block
	Recall-next-block: retrieve next block
	Do-next-block: if there is a next block, set new goal of Dial-Block
	Done-blocks: if there is no next block, continue
	Press-enter: type *Enter* termination key, set new goal of Dial-Return
Dial-Return	
	Home-hand: move right hand to steering wheel, terminate goal

The dialing model derived from a task analysis of dialing a 10-digit phone number, resulting in the goals and production rules outlined in Table 4. First, the Dial-Move goal locates the numeric keypad and moves the hand from the steering wheel to the keypad, assumed here to be equivalent to a move from the home row of a standard keyboard to the numeric keypad as implemented in ACT–R. Next, the Dial-Prepare goal prepares the dialing procedure by recalling the declarative chunk representing the first block of numbers; the model segments the phone number into blocks of 3, 3, and 4 digits corresponding to the typical segmentation *xxx-xxx-xxxx*. Then, the Dial-Block goal iteratively dials each block by recalling and typing each digit and recalling the next block. When the retrieval of the next block fails and thus dialing is complete, the model presses the *Enter* key and moves the hand back to the steering wheel. Finally, the Dial-Return goal moves the right hand back to the steering wheel and terminates the dialing goal. The iterating heuristic applies for the Dial-Block goal, which repeats for each of the three blocks in the phone number. The model assumes that the keystrokes that compose the actual dialing in the Dial-Block goal are not "significant" pauses, and thus the blocking heuristic does not apply. However, the model does consider hand movements between steering wheel and keypad significant pauses, and thus the blocking heuristic applies for the Dial-Move and Dial-Return goals.

Typing times are particularly critical for the dialing model's performance, but also susceptible to large variability in the human data, and thus it was important to ensure that the model's typing matched reasonably well with that of the human participants. To this end, the execution time for keying a digit on the keypad was set to the average keystroke time found in the empirical study, 260 msec, minus the 50-msec cognitive initiation time needed for a production firing, resulting in a 210-msec motor time. In her TYPIST model, John (1996) found that a keystroke motor time of 230 msec worked well for typists at speeds of 30 gross words per minute (gw/min), but this time can decrease significantly for faster typists—for example, to 170 msec for a 60-gw/min typist. Thus, the average keystroke motor time found here falls nicely within her reported range of typical times. In addition, the model uses John's (1996) assumption that the cognitive processor waits for the completion of each keystroke before firing a new production rule.

As in Studies 1 and 2, the creation of the integrated model for driving and dialing becomes a straightforward task, given the general executive. As in Study 2, the model initially runs only the control goal to navigate the environment, then adds the dialing goal to initiate dialing. The general executive again uses the same noise value as the other two studies. The control model again uses the same control delay time, and because the Study 3 driving environment is identical to that in Study 2, this study imports all other control parameters from Study 2. The integrated model was run in five driving simulations with eight dialing trials per simulation spaced 20 sec apart, totaling 20 km of driving data. To avoid the possibility of the driver model overtaking the lead car (e.g., if it braked too late to avoid a crash), the model's vehicle was constrained to a minimum following distance of 10 m behind the lead vehicle. The model was also run in five simulations without driving to generate baseline performance on the dialing task.

3.3.3. Results

3.3.3.1. Aggregate effects of task switching. The human and model data arising from this study include a host of information elucidating the multitasking behavior with driving and dialing. Before moving to a detail analysis of task switching, it is important to ensure that the general executive produces the same effects observed in previous studies of driver distraction—namely, aggregate effects of dialing on driving, and also effects of driving on dialing. First we consider aggregate effects of driving on dialing. Fig. 7(a) shows the mean dialing times in the baseline condition, representing the data collected during the pretest and posttest with no driving task, and in the driving condition, representing the data collected for dialing while driving. For the human drivers, driving had a significant effect on dialing time, $t(6) = 3.40$, $p < .05$; perhaps surprisingly, this difference is not large, with driving adding only 0.55 sec to the mean dialing time (4.46 sec baseline, 5.01 sec driving). The model produced a very close fit to these data, including the observed increase due to driving. In essence, driving does take some time away from execution of the dialing sequence, but at the same time, the control updates occur quickly and can be smoothly interleaved with dialing such that the total time added remains relatively small. The model exhibited almost no variability in the baseline condition but a small amount in the driving condition, where vehicle stability played a role in allowing or disallowing the model to switch away from control.

Next we consider aggregate effects of dialing on driving. Fig. 7(b) shows two measures of driver performance: lateral deviation, computed as a root-mean-squared error between the vehicle's current lateral position and the center position of the lane (see, e.g., Salvucci, 2001); and speed deviation, computed as a root mean squared error between the vehicle's current speed and the speed of the lead vehicle. As in such analyses in previous studies, these measures were computed both during normal driving and while dialing, where the latter included a 5-sec window after completion of dialing to account for subsequent corrections in control. For lateral deviation, human drivers exhibited a significant effect of the dialing task, $t(6) = 2.99$, $p < .05$. The model also exhibited this effect; the effect size was larger

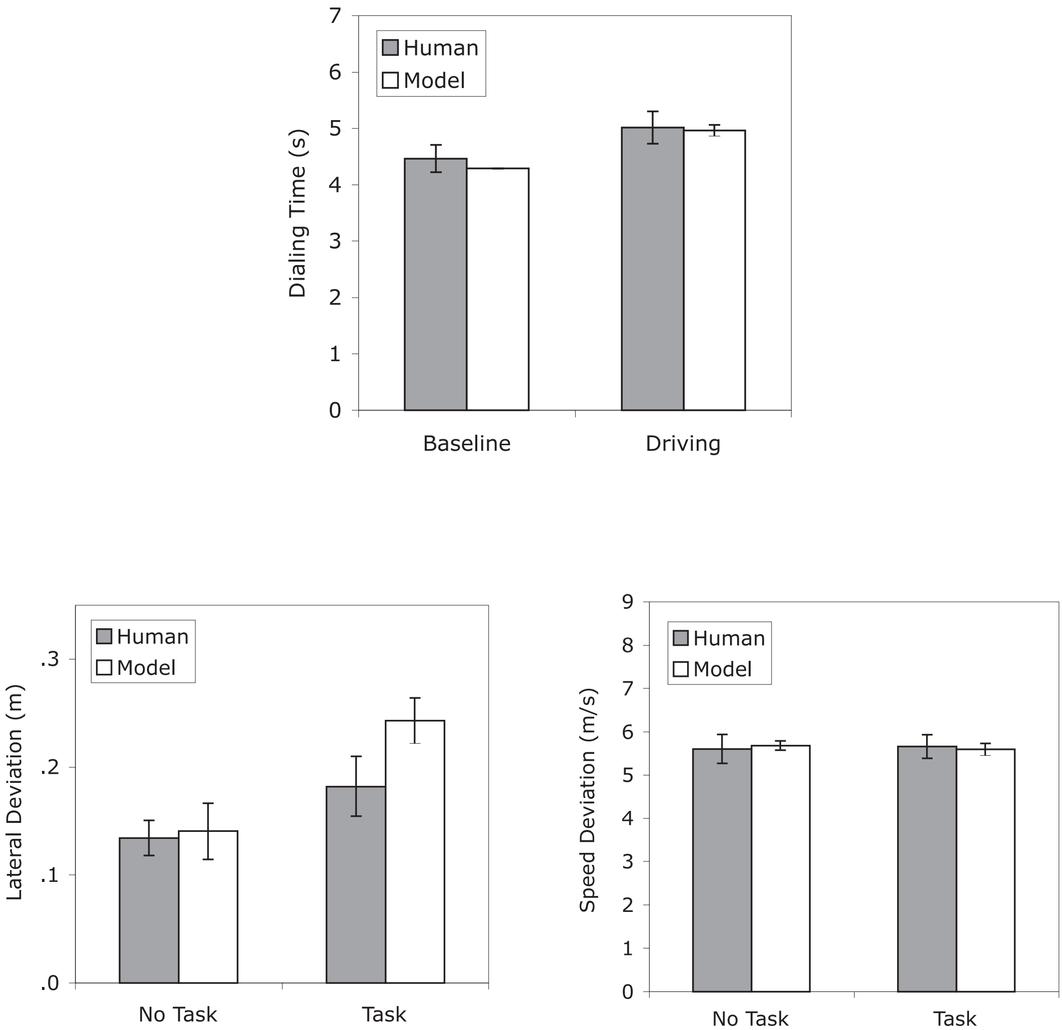

Fig. 7. Study 3: Aggregate effects of (a) driving on dialing as measured by dialing time, and (b) dialing on driving as measured by lateral deviation and speed deviation. Errors bars represent standard errors.

for the model than for human drivers, although there was also considerable variability in both model and human data. Interestingly, there was no significant effect of driving on the speed deviation measure for the human drivers, $t(6) = .35$, $p > .5$; the car-following aspect of the driving task turned out to be quite difficult for the drivers, leading to a fairly high mean speed deviation (over 5 m/sec or roughly 11 mph), and thus the presence of the secondary task did not add significantly to this difficulty. The model, experiencing the same difficult car-following as the human drivers, also showed no effect of the secondary task. This highlights an important finding for the model and for the general executive: Multitasking may show distraction effects for some measures and not for others, and the general executive nicely predicts the significant effects and, just as importantly, the lack of significant effects for appropriate measures.

3.3.3.2. Detailed measures of task switching. Although this overall picture of dialing provides some clues as to the efficient interleaving of driving and dialing, perhaps the most important validation for the general executive arises in analysis of exactly when drivers switched between dialing and driving. In the radio-tuning task, we could analyze driver eye movements to find when drivers switched, because the task necessitated visual fixation on the radio for successful execution. In the phone-dialing task, drivers had no need to take their eyes off the road because the interface (i.e., the numeric keypad) gave no visual feedback. Thus, the phone-dialing task required a different approach to analyzing task-switching behavior. For this purpose, our study uses *key delays* to elucidate driver task switching, namely, the time between one key press and the next during the course of dialing a phone number. The baseline condition (dialing alone without driving) provides baseline key delays without interference from driving; the driving condition (dialing while driving) provides delays that include interference from driving. By comparing these baseline and driving conditions, we can determine where extra time was needed in the driving condition and thus where, presumably, drivers interleaved control with dialing.

For the human drivers, Fig. 8(a) shows the mean key delays computed as the time preceding each key press, including the 10 key presses for the phone number and the final *Enter* key press. The first key press showed a very significant effect of driving, $t(6) = 4.17$, $p < .01$, and the delays for both baseline and driving were much longer than the other delays due to initiation of the dialing (e.g., hand movement to the keypad) and because the driving condition included extra driving time to ensure stability of the vehicle before starting to dial. Also, the key presses that initiate new blocks within the phone number (Keys 4 and 7, for a phone number of the form *xxx-xxx-xxxx*) showed a significant effect of driving, $t(13) = 3.12$, $p < .01$; again, this difference indicates the extra time needed to control the vehicle to an acceptable stability. However, both the intermediate key presses (Keys 2–3, 5–6, 8–10) and the final *Enter* key press (Key 11) showed no significant effects of driving, $p > .30$. Thus, the human drivers seemed to interleave iterations of control at the block boundaries, but not within blocks, except for the final key press, which required more time than the intermediate key presses but still showed no significant difference while driving.

Fig. 8(b) shows the analogous graph of key delays for the model. Overall the model reproduced many aspects of the human driver data, $R^2 = .96$, $RMSE = .10$, including: (a) the longest baseline time for Key 1 and longer baseline times for Keys 4, 7, and 11; (b) effects of driving for the first key press and for the block-starting key presses, namely, Keys 1, 4, and 7; (c) no effects of driving for the intermediate key presses and the final key press. Like the human drivers, the model exhibited effects on the block boundaries because of the extra time needed to control the vehicle; these block boundaries coincide with the goal boundaries of dialing a block of numbers, and thus this result falls directly from the goal structure in the model. Although the effect of driving for Keys 4 and 7 was slightly larger than those for the human drivers, there was a fair amount of variability in the driving condition because of the "randomness" of how these boundaries coincide with different driving situations (e.g., whether the boundary occurred during a stable straight road segment or a difficult curved

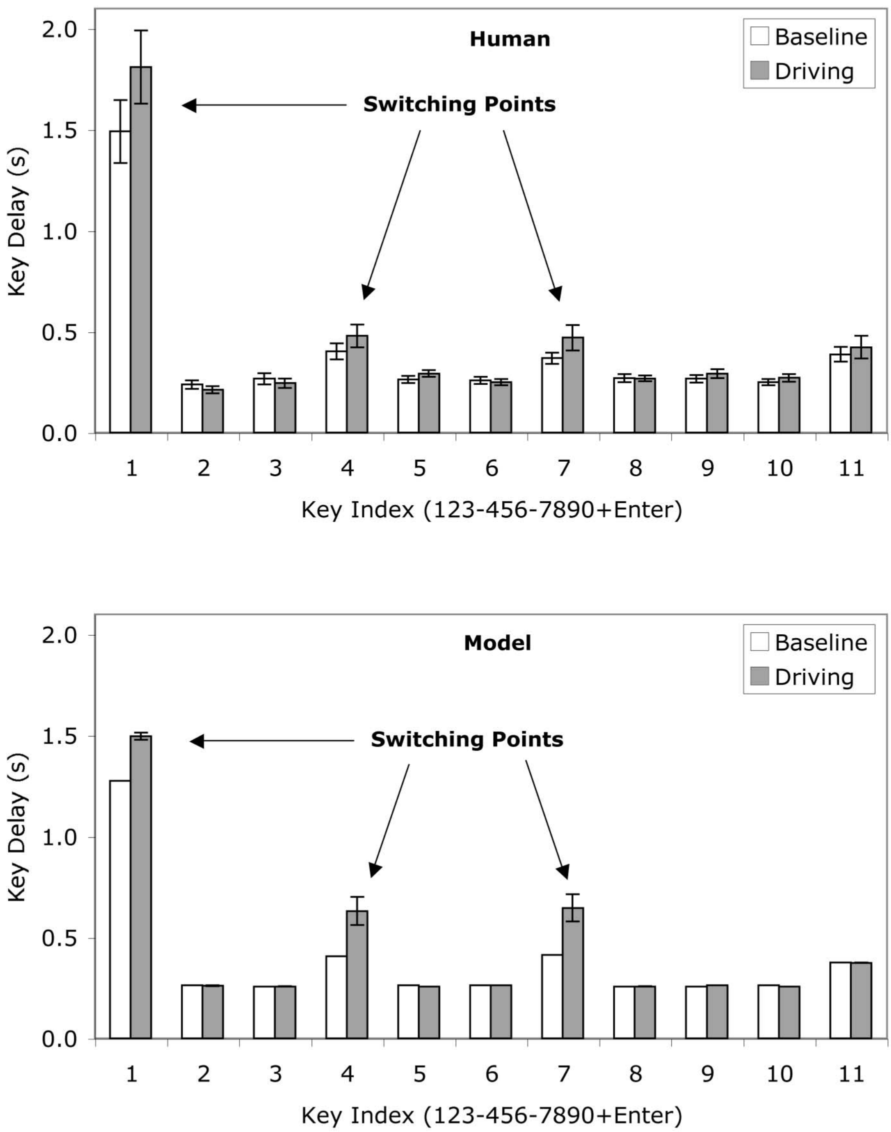

Fig. 8. Study 3: Task-switching points as illustrated by key delay times for (a) human drivers, and (b) model simulations. Errors bars represent standard errors.

segment). The model also somewhat underpredicted the delay times for the first key press; one could argue that the model has more "readiness" to start dialing immediately, whereas drivers seemed to exhibit some extra time to initiate the dialing goal. Nevertheless, the model nicely reproduces the trend in the key press data of where drivers switch back from dialing to driving, specifically at the block boundaries.

3.4. Summary and lessons learned

These three studies explored how the general executive can account for various aspects of driver multitasking. In our previous modeling efforts in this domain (e.g., Salvucci, 2001b, 2002; Salvucci & Macuga, 2002), the models used customized executives to exhibit multitasking behavior: The component models managed processes by explicitly passing execution from one to another through the setting of new goals (e.g., driving passing control to dialing, then dialing back to driving). The studies here demonstrate how the general executive provides a more rigorous and plausible description of driver multitasking in three ways as related to the general principles outlined earlier.

First, the general executive as an architectural mechanism provides a more plausible and consistent account of multitasking that more easily generalizes across studies, as compared to previous domain-specific customized executives. In the previous models, the component models could not be easily separated from each other—for instance, because the control model specifically mentioned and passed execution to the dialing model, removal of the dialing model would "break" the control model. Such a strongly unified model might be reasonable after long periods of learning (arguably the control and monitoring models presented here may achieve that level of integration), but component models for individually well-learned skills, such as driving and dialing, can clearly be decomposed and run independently without reliance on the other. Second, to achieve integration with another task model, the component models would require modification to explicitly manage execution with the new model—for instance, for driving and radio tuning, the driving model would require changes to set the goal to radio tuning, and the tuning model would need analogous changes to set the goal back to driving. This would imply that with each new integrated set of tasks, a person would need to learn new rules to manage that specific set of tasks. Again, such rules might sometimes be achievable through learning, but learning specific rules to manage all possible (or reasonable) sets of tasks would lead to an exponential explosion in rules and seems implausible as models account for a fuller scope of human cognition.

Because the general executive handles task management and scheduling, the four models used in the studies (control, monitoring, radio tuning, and phone dialing) do not refer or pass execution to one another, but instead only take care of their own processing. For this reason, the models could plausibly be learned independently of one another and can also run independently if desired. Second, the general executive mechanism remained the same across all studies, and its one parameter (temporal noise) remained constant across all studies. This further confirms our goal of having a single mechanism operate over any set of possible desired tasks. Third, the primary task of vehicle control, common to all three studies, was handled by the same control model across studies. The only caveat to this reuse arose in changes to control-specific parameters due to differences in driving environments in the empirical studies; nevertheless, even these parameters remained unchanged when using the same driving environment (namely, Studies 2 and 3 used the same driving environment and thus used the exact same control model). Most important, the model parameter most critical for multitasking, the control delay time, remained constant across all studies. Thus, unlike our previous studies with customized executives, these studies demonstrate how reuse of both the general executive and component models can greatly facilitate modeling of multitasking and lead to a more plausible account of the integration of independently learned component models.

The second way in which the general executive extends our earlier models of driving relates to the incorporation of time in the multitasking model. The previous models did not reason about time in their multitasking schemes, but rather used one of two methods for multitasking: (a) They defined an estimated probability of switching from this task to another—for instance, the probability of monitoring after one cycle of control updates; or (b) they allowed only a single iteration before switching tasks—for instance, performing a single monitoring goal and then switching back to driving. These two methods could produce only a very limited set of switch probability distributions, namely, an exponential distribution that fades over time (a) or a trivial distribution where all switches occur at the same point in time (b). These two distributions clearly do not match the switch probabilities observed in the driving studies, particularly those in Figs. 5(a) and 6(a). The proposed general executive helps to account for the temporal trends in these data by incorporating time into its task scheduling and allowing tasks to proceed until their time "expires" and they relinquish execution.

The third way in which the general executive benefits previous driving work involves the use of representational heuristics to guide model goal representations. The previous studies, in fact, used the iterating heuristic implicitly by switching only after completed iterative cycles (e.g., after a control update), and they also hinted at the blocking heuristic as part of the guidance for the development of the first dialing model for driving (Salvucci, 2001b). The two heuristics as defined here help to formalize these previous assumptions into two basic heuristics that can guide future model development. In addition, Study 3 in particular demonstrates that goal representations can play a key role in where people switch between tasks. As mentioned earlier, although these heuristics (particularly the blocking heuristic) do not fully specify the desired goal representations, they at least help constrain the space of representations in such a way as to make models more amenable to integrated multitasking.

4. General discussion

The proposed general executive consists of both a computational mechanism and representational heuristics for allowing interleaved execution of multiple task models. The application to the driving domain illustrates how the general executive facilitates modeling of compound continuous tasks and helps to account for both aggregate effects of multitasking and detailed measures of task-switching behavior. Of course, no one domain or one modeling effort can suffice in validating the general executive; further studies in both simple laboratory domains and complex dynamics tasks are needed to flesh out its strengths and weaknesses. However, driving serves as an excellent case study in launching such an effort for the ACT–R architecture: It serves as a representative example of a wide variety of domains that demand dynamic integration of cognitive, perceptual, and motor processes in the context of a complex real-world task. The driving studies here demonstrate that the general executive is at least general enough to account for a variety of aspects of driver multitasking, showing promise and opening up the theory to further validation in other domains.

The development of models for complex tasks often relies on prior development of components of these tasks, and thus model reuse is critical in facilitating both theoretical consistency and practical development. Reuse can occur in a number of ways. The cognitive architecture itself provides the first layer of reuse in that its built-in theories and mechanisms cut across all models with a baseline theory and simulation engine. At times, reuse can provide common representations of information for general tasks—for instance, representations (and perhaps corresponding models) proposed for list memory (Anderson, Bothell, Lebiere, & Matessa, 1998) and analogy (Salvucci & Anderson, 2001) offer a common way of using these general mechanisms within larger models. At a higher level, reuse can involve embedding entire models within larger models, including declarative representations and chunks as well as their associated production systems. The studies of driving presented here serve as one example: A model of radio tuning or phone dialing can be developed and run completely by itself, but can also be easily integrated with the driver model to immediately generate emergent predictions of the interactions that arise between the tasks. This type of reuse with the proposed executive should generalize in a straightforward way to a large range of possible models; for example, we might integrate the driver model with Pirolli's (this issue) model of information retrieval to predict interactions of driving with e-mail or Web access (e.g., through currently available "smart phone" devices); we might integrate the driver model with Lewis's (this issue) model of parsing to predict interactions of driving with display reading or conversation; or we might integrate Lewis's model with Anderson's (this issue) model of algebraic symbol manipulation to predict how students might encode instructions or "over-the-shoulder" help while solving algebra problems. Model reuse offers enormous predictive power to cognitive architectures: As individual researchers explore particular issues of driving, language, memory, and so forth, the entire modeling community immediately benefits from the work and can integrate the resulting models, representations, and so forth, into their own efforts.

Besides the critical issue of model reuse, another issue brought to light by the general executive is the need for awareness and reasoning about time. Time clearly serves as an essential aspect of many dynamic complex tasks, and thus it is not surprising that models of these tasks must exhibit awareness of time as part of their behavior. The proposed general executive incorporates an implicit awareness of time through its scheduling mechanism: New goals are temporally ordered on the goal queue, and a model can modify the default ordering through specification of goal delay times. Other modeling efforts have recently begun to explore explicit reasoning about the passage of time for tasks such as interval estimation (Taatgen, van Rijn, & Anderson, 2004), including accounts of the types of errors made during estimation of temporal intervals. These efforts nicely complement the proposed executive and, perhaps in the near future, could replace its now-rudimentary formulation of time with a more rigorous account of temporal reasoning.

Although the general executive focuses currently on internally driven task switching, some task switching can be driven externally by the environment—for instance, when a choice stimulus appears on screen during a manual tracking task. The general executive does not directly speak to this issue, but in fact offers a solution for dealing with such task switches: When the model does sense the external stimulus, it creates a new goal with the purpose of responding to the stimulus, which itself is then interleaved with the primary task. In ACT–R, the sensing of the external stimulus can be performed through use of the "buffer stuffing" (e.g., Fick & Byrne, 2003) in which a new stimulus immediately appears in a visual or other perceptual buffer. Subsequently, a production rule can react to the presence of a new stimulus in the buffer and create a new goal to handle and respond to the stimulus. Thus, although the general executive does not provide a method of handling the environmentally driven stimulus directly, such a method already exists in the ACT–R architecture, and the general executive provides a complementary mechanism that allows a model to respond without necessarily drawing full attention away from the primary task.

The general executive as currently specified certainly does not address all the known issues related to multitasking within cognitive architectures. One significant limitation is that the executive currently has no relation to declarative memory and activation, and thus goals cannot be retrieved or forgotten like other declarative chunks. For our chosen domain of driving, such issues are arguably less relevant: When switching tasks at the subsecond level, presumably goals rarely if ever fall below a threshold of retrieval at which they would fail to be recalled (e.g., a driver forgetting the primary driving goal after switching to radio tuning), at least in part because of continual reminding from environmental stimuli. Nevertheless, I expect that as the general executive generalizes to other domains, issues concerning goals as declarative chunks, such as decay and interference (e.g., Altmann & Gray, 2002), will require modifications to this framework.

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). The initial stage of learning would include the learning of declarative instructions that, over time, shift to more procedural execution (e.g., Taatgen & Lee, 2003). In the final stages of learning, human behavior can become highly skilled such that the parallel streams of cognition, perception, and motor movements are highly optimized and interleaved; such behavior has been modeled in cognitive architectures in smaller task paradigms (e.g., Byrne & Anderson, 2001; Meyer & Kieras, 1997) and larger complex tasks (e.g., Vera, Howes, McCurdy, & Lewis, 2004), and notably Taatgen (2005, this issue) explores how such optimizations may arise directly from instruction learning. The general executive proposed here lies somewhere in between: The skills of driving, radio tuning, and phone dialing are well-learned tasks—well past the instruction stage—and thus the executive manages these production-level compiled tasks, but at the same time the executive does not predict the highly optimized interleaving demonstrated in the recent work mentioned previously. Nevertheless, my hope is that the general executive will generalize to incorporate such mechanisms and help bridge the sometimes large gap between initial learning and extremely skilled execution.

Notes

1. In the actual production-rule syntax, the general executive reinterprets standard ACT–R syntax such that “+goal” adds new goals and “–goal” removes these goals. It also assumes that the first “+goal” in a rule removes this goal before adding—thus ensuring that models written for this ACT–R with one “+goal” run exactly as before.
2. The tuning and dialing models actually have one additional production rule that clears ACT–R’s motor program when the task completes; this rule compensates for a problem with this driver model in which steering movements do not automatically clear the motor program as they would, given a complete motor module that integrates steering and other hand movements.

Acknowledgments

This work was supported by Office of Naval Research grant #N00014–03–1–0036. Many thanks to Frank Lee and Yelena Kushleyeva for guidance on the original model, and to John Anderson, Wayne Gray, Wai-Tat Fu, Mike Schoelles, Niels Taatgen, and Rob Goldstone for many comments and helpful suggestions.

References

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

Altmann, E. M., & Gray, W. D. (2000). An integrated model of set shifting and maintenance. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the 3rd International Conference on Cognitive Modeling* (pp. 17–24). Veenendaal, The Netherlands: Universal Press.

Altmann, E. M., & Gray, W. D. (2002). Forgetting to remember: The functional relationship of decay and interference. *Psychological Sciences, 13,* 27–33.

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

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science, 29,* 313–341.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111,* 1036–1060.

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language, 38,* 341–380.

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

Anderson, J. R., Taatgen, N. A., & Byrne, M. D. (in press). Learning to achieve perfect time sharing: Architectural implications of Hazeltine, Teague, & Ivry (2002). *Journal of Experimental Psychology: Human Perception and Performance.*

Best, B. J., & Lebiere, C. (2003). Spatial plans, communication, and teamwork in synthetic MOUT agents. In *Proceedings of the 12th Conference on Behavior Representation in Modeling and Simulation.*

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

Byrne, M. D., & Kirlik, A. (2005). Using computational cognitive modeling to diagnose possible sources of aviation error. *International Journal of Aviation Psychology, 15,* 135–155.

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

Chong, R. S. (1998). *Modeling dual-task performance improvement: Casting executive process knowledge acquisition as strategy refinement* (Tech. Rep. No. CSE-TR-378-98). Unpublished doctoral dissertation, Department of Computer Science and Engineering, University of Michigan, Ann Arbor.

Dingus, T. A., Antin, J. F., Hulse, M. C., & Wierwille, W. W. (1989). Attentional demand requirements of an automobile moving-map navigation system. *Transportation Research, A23,* 301–315.

Doane, S. M., & Sohn, Y. W. (2000). ADAPT: A predictive cognitive model of user visual attention and action planning. *User Modeling and User Adapted Interaction, 10,* 1–45.

Ellis, R. D., Goldberg, J. H., & Detweiler, M. C. (1996). Predicting age-related differences in visual information processing using a two-stage queuing model. *Journal of Gerontology: Psychological Sciences, 51,* 155–165.

Fick, C. S., & Byrne, M. D. (2003). Capture of visual attention by abrupt onsets: A model of contingent orienting. In F. Detje, D. Dörner, & H. Schaub (Eds.), *Proceedings of the Fifth International Conference on Cognitive Modeling* (pp. 81–86). Bamberg, Germany: Universitas-Verlag Bamberg.

Fincham, J. M., Carter, C. S., van Veen, V., Stenger, V. A., and Anderson, J. R. (2002). Neural mechanisms of planning: A computational analysis using event-related fMRI. *Proceedings of the National Academy of Sciences, 99,* 3346–3351.

Freed, M. (1998). Managing multiple tasks in complex, dynamic environments. In *Proceedings of the 1998 National Conference on Artificial Intelligence* (pp. 921–927). Menlo Park, CA: AAAI Press.

Fu, W-T., Bothell, D., Douglass, S., Haimson, C., Sohn, M.-H, & Anderson, J. R. (2004). Learning from real-time over-the-shoulder instructions in a dynamic task. In *Proceedings of the Sixth International Conference on Cognitive Modeling* (pp. 100–105). Pittsburgh, PA: Carnegie Mellon University.

Gluck, K. A., Ball, J. T., Krusmark, M. A., Rodgers, S. M., & Purtee, M. D. (2003). A computational process model of basic aircraft maneuvering. In F. Detje, D. Doerner, & H. Schaub (Eds.), In *Proceedings of the Fifth International Conference on Cognitive Modeling* (pp. 117–122). Bamberg, Germany: Universitats-Verlag Bamberg.

Gluck, K. A., & Pew, R. W. (Eds.). (in press). *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation, and validation*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Gray, W. D., & Schoelles, M. J. (2003). The nature and timing of interruptions in a complex cognitive task: Empirical data and computational cognitive models. In *Proceedings of the 25th Annual Meeting of the Cognitive Science Society* (p. 37). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Hendricks, D. L., Freedman, M., Zador, P. L., & Fell, J. C. (2001). *The relative frequency of unsafe driving acts in serious traffic crashes.* Washington, DC: National Highway Traffic Safety Administration.

Hess, R. A., & Modjtahedzadeh, A. (1990). A control theoretic model of driver steering behavior. *IEEE Control Systems Magazine, 10*(5), 3–8.

Hildreth, E. C., Beusmans, J. M. H., Boer, E. R., & Royden, C. S. (2000). From vision to action: Experiments and models of steering control during driving. *Journal of Experimental Psychology: Human Perception and Performance, 26,* 1106–1132.

John, B. E. (1996). TYPIST: A theory of performance in skilled typing. *Human–Computer Interaction, 11,* 321–355.

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.

Just, M. A., Carpenter, P. A., & Varma, S. (1999). Computational modeling of high-level cognition and brain function. *Human Brain Mapping, 8,* 128–136.

Kelley, T. D., & Scribner, D. R. (2003). Developing a predictive model of dual task performance (Tech.Rep. ARL-MR-0556). Aberdeen, MD: U.S. Army Research Laboratory.

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

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,* 391–438.

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

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.

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

Koechlin, E., Corrado, G., Pietrini, P., & Grafman, J. (2000). Dissociating the role of the medial and lateral anterior prefrontal cortex in human planning, *Proceedings of the National Academy of Sciences USA, 97,* 7651–7656.

Kushleyeva, Y., Salvucci, D. D., & Lee, F. J. (2005). Deciding when to switch tasks in time-critical multitasking. *Cognitive Systems Research, 6,* 41–49.

Laird, J. E., & Duchi, J. C. (2000). Creating human-like synthetic characters with multiple skill levels: A case study using the Soar Quakebot. In *Papers from the AAAI 2000 Fall Symposium on Simulating Human Agents* (Tech. Rep. FS-0A-03; pp. 75–79). Menlo Park, CA: AAAI Press.

Lallement, Y., & John, B. E. (1998). Cognitive architecture and modeling idiom: An examination of three models of the Wickens task. In M. A. Gernsbacher & S. J. Derry (Eds.), *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 597–602). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Lebiere, C., Gray, R., Salvucci, D., & West, R. (2003). Choice and learning under uncertainty: A case study in baseball batting. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society* (pp. 704–709). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Lee, F. J., & Anderson, J. R. (2000). Modeling eye-movements of skilled performance in a dynamic task. In *Proceedings of the Third International Conference on Cognitive Modeling* (pp. 194–201). Gronigen: University of Groningen, Netherlands.

Liu, Y. (1996). Queueing network modeling of elementary mental processes. *Psychological Review, 103,* 116–136.

McClelland, J. L. (1979). On the time relations of mental processes: An examination of systems of processes in cascade. *Psychological Review, 86,* 287–330.

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

Miller, J. O. (1993). A queue-series model for reaction time, with discrete-stage and continuous-flow models as special cases. *Psychological Review, 100,* 702–715.

Nelson, R. (1995). *Probability, stochastic processes, and queueing theory—The mathematics of computer performance modeling*. New York: Springer Verlag.

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

Reed, M. P., & Green, P. A. (1999). Comparison of driving performance on-road and in a low-cost simulator using a concurrent telephone dialing task. *Ergonomics, 42,* 1015–1037.

Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Supporting cognitive models as users. *ACM Transactions on Computer Human Interaction, 7,* 141–173.

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

Salvucci, D. D. (2001b). 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. (2002). Modeling driver distraction from cognitive tasks. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society* (pp. 792–797). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Salvucci, D. D. (in press). Modeling driver behavior in a cognitive architecture. *Human Factors*.

Salvucci, D. D., & Anderson, J. R. (2001). Integrating analogical mapping and general problem solving: The path-mapping theory. *Cognitive Science, 25,* 67–110.

Salvucci, D. D., Boer, E. R., & Liu, A. (2001). Toward an integrated model of driver behavior in a cognitive architecture. *Transportation Research Record, 1779,* 9–16.

Salvucci, D. D., & Gray, R. (2004). A two-point visual control model of steering. *Perception, 33,* 1233–1248.

Salvucci, D. D., Kushleyeva, Y., & Lee, F. J. (2004). Toward an ACT–R general executive for human multitasking. In *Proceedings of the Sixth International Conference on Cognitive Modeling* (pp. 267–272). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

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

Schoppek, W. (2002). Examples, rules, and strategies in the control of dynamic systems. *Cognitive Science Quarterly, 2,* 63–92.

Smith, E. E., & Jonides, J. (1999, March 12). Storage and executive processes in the frontal lobes, *Science, 283,* 1657–1661.

Sodhi, M., Reimer, B., & Llamazares, I. (2002). Glance analysis of driver eye movements to evaluate distraction. *Behavior Research Methods, Instruments and Computing, 34,* 529–538.

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. (2005). Modeling parallelization and flexibility improvements in skill acquisition: From dual tasks to complex dynamic skills. *Cognitive Science, 29,* 421–455.

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

Taatgen, N., van Rijn, H., & Anderson, J. R. (2004). Time perception: Beyond simple interval estimation. In Pro*ceedings of the Sixth International Conference on Cognitive Modeling* (pp. 296–301). Pittsburgh, PA: Carnegie Mellon University.

Tsimhoni, O., & Liu, Y. (2003). Modeling steering using the Queuing Network—Model Human Processor (QN-MHP). In *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting* (pp. 1875–1879). Santa Monica, CA: Human Factors and Ergonomics Society.

Vera, A., Howes, A., McCurdy, M., & Lewis, R. L. (2004). A constraint satisfaction approach to predicting skilled interactive performance. In *Human Factors in Computing Systems: CHI 2004 Conference Proceedings* (pp. 121–128). New York: ACM Press.

Wierwille, W. W. (1993). Visual and manual demands of in-car controls and displays. In B. Peacock & W. Karwowski (Eds.), *Automotive ergonomics* (pp. 229–320). London: Taylor & Francis.