MODELING TOOLS FOR PREDICTING DRIVER DISTRACTION

Dario D. Salvucci
Drexel University
Philadelphia, PA

In contrast to the vast amount of modeling work focused on desktop user interfaces, recent work has increasingly focused on "off-the-desktop" interfaces, one prime example being in-vehicle interfaces used while driving. This paper highlights four recent approaches to predicting driver distraction from in-vehicle interfaces as secondary tasks: hand-crafted modeling with the full-fledged ACT-R architecture, hand-crafted modeling with the much less complex ACT-Simple framework, modeling-by-demonstration using the new CogTool, and simplified modeling-by-demonstration using the integrated Distract-R system. While all four use an integrated-model approach and a rigorous driver model, each approach illustrates different advantages and disadvantages of simplifying cognitive modeling for purposes of rapid prototyping and evaluation of in-vehicle interfaces.

INTRODUCTION

While a vast majority of modeling work in human-computer interaction has focused on desktop systems, the recent industry and research trends involving PDAs, mobile systems, ubiquitous systems, and the like have emphasized the importance of "off-the-desktop" systems. One domain in particular that has received a fair bit of attention is that of interface use while driving, namely the issue of driver distraction that can result from in-vehicle device interaction. Driver distraction and inattention has become the foremost cause of crashes in the United States (Hendricks et al., 2001), and researchers and practitioners alike have worked to explore and alleviate the sources of this distraction.

Along with my collaborators, I have been exploring how to predict the effects of driver distraction through the use of cognitive models. Specifically, we have developed and utilized four different approaches to predicting distraction: one that uses the ACT-R architecture (Anderson et al., 2004) in conjunction with a rigorous model of driver behavior (Salvucci, in press); one that uses the ACT-Simple framework (Salvucci & Lee, 2003) in conjunction with this driver model; one that integrates the new CogTool (John et al., 2004) with the driver model; and finally one, Distract-R (Salvucci et al., 2005), that emphasizes very rapid prototyping and evaluation with an optimized driver model in a single standalone package. The following sections summarize each approach and draw comparisons among them, noting the advantages and disadvantages of each.

MODELING TOOLS

ACT-R + Driver Model

The first (and original) approach utilizes the full-fledged ACT-R cognitive architecture (Anderson et al., 2004) as a framework for cognitive modeling. Given a new in-vehicle interface, the modeler creates hand-crafted ACT-R production systems that interact with the interface and produce the desired task behavior. This model is then integrated with a rigorous model of driver behavior (Salvucci, in press), also implemented as an ACT-R production system. The integrated model runs in simulation and occasionally uses the interface as a secondary task while driving, thus producing a host of behavioral predictions about driver performance (e.g., deviation from lane center, or braking response time for a sudden event). The original study using this approach (Salvucci, 2001) modeled phone dialing with two possible modalities — by hand or by voice — and compared the distraction potential of the two modalities. Later studies offered additional performance studies of phone dialing (Salvucci & Macuga, 2002) and radio tuning (Salvucci, 2005).

Creating ACT-R models by hand clearly offers the greatest flexibility and predictive power for a modeler, since it allows for the full functionality of the architecture. However, it also involves a serious commitment: hand-crafted models can take hours, days, or weeks to develop, especially considering that they must interact with a working interface (typically implemented in LISP, as is ACT-R itself). Thus, model development is generally reserved for only highly trained modelers (i.e., at least several weeks of training), and requires hours of work even for such experts to complete.

ACT-Simple + Driver Model

The second approaches uses the ACT-Simple framework (Salvucci & Lee, 2003), which was originally designed as a rapid modeling tool for desktop user interfaces; it includes simple operators such as (*look-at*) and (*press-key*) that are translated to full ACT-R production rules, thus offloading the rigor of writing full rules from the modeler to the automated system. Because ACT-Simple models are "compiled" to ACT-R rules, the resulting models easily integrate with the ACT-R driver model just as a hand-crafted model would. For instance, we have modeled phone dialing in ACT-Simple (with the numeric keypad as an analog to a phone) and used the resulting models to compare the distraction effects of dialing on younger versus older drivers (Salvucci, Chavez, & Lee, 2004).

Table 1: Ratings for four approaches to predicting driver distraction.

	Required Effort			Theoretical Fidelity		Task Generality	Prediction Accuracy
	Creating Interfaces	Creating Models	Running Simulations	Perceptual-Motor	Cognitive		
ACT-R	hours	hours	hours	√√	√√	√√	√
ACT-Simple*	hours	minutes	hours	√	–	√√	√
CogTool	minutes	minutes	hours	√	–	√√	√
Distract-R	minutes	minutes	seconds	√	–	√	√

The most important feature of an ACT-Simple model is the high-level specification that modelers — even those with very little training — can quickly type out in a matter of minutes. Because of the translation to ACT-R, this simple model also benefits from the rigor of the underlying cognitive architecture which helps guarantee a degree of psychological validity. However, because there is no mock-up of the actual interface, most of the ACT-Simple operators cannot specify arguments; for instance, the (*look-at*) operator cannot specify where to move, since there is no specification of interface objects. Thus, the framework is forced to make general assumptions about these movements in the average or canonical case, much like Card, Moran, and Newell's (1983) keystroke-level models.

CogTool + ACT-Simple* + Driver Model

CogTool (John et al., 2004) is a new system for rapid prototyping and evaluation of graphical user interfaces. The tool allows a user to mock-up new interfaces in HTML and create models by demonstration — that is, clicking through the interface to demonstrate sample task processes. When a user demonstrates a task sequence, the user's actions are translated into an ACT-Simple model that runs in ACT-R. However, it actually incorporates a modified ACT-Simple that includes arguments for the operators, such as specifying the target of mouse movements (not possible with the original ACT-Simple). In doing so, the model can interact with the mockup: CogTool provides a connection between ACT-R and the HTML mock-up such that model actions are executed on the interface and cause it to be updated appropriately.

John, Salvucci, Centgraf, and Prevas (2004) describe the integration of CogTool with the driver model to predict distraction as in the earlier ACT-Simple studies. In this case, however, CogTool allows an extra level of cognitive fidelity in incorporating the interface objects into the simulation, thus providing a more powerful tool for prediction. At the same time, the system required no fewer than three applications running together (CogTool in Dreamweaver, ACT-R, and Netscape); also, the simulations, due to limitations of the driver model implementation in LISP, ran approximately in real time and thus hindered rapid evaluation.

Distract-R

Distract-R (Salvucci et al., 2005) is a system developed to address the speed and system integration issues of the integrated CogTool approach. First, it includes a complete re-implementation (in C++) of the driver model and a minimally sufficient ACT-R to maximize speed: the new implementation runs approximately 100 times faster than the original model, thus allowing, for instance, 10 minutes of virtual driving simulation in approximately 2-3 seconds of real time. Second, Distract-R incorporates all the basic elements of CogTool, ACT-Simple, and ACT-R in a single stand-alone application, thus facilitating download and running of the full system; in fact, we have also developed an applet-based version of the system (http://cog.cs.drexel.edu/distract-r) that users can interact with over the web with no download overhead and no (or limited) compatibility issues.

Nevertheless, because Distract-R focuses on very rapid prototyping and evaluation, it does not include the full power of the integrated CogTool system. For instance, the Distract-R prototyping tool cannot specify the "state" of the interface, only the layout of buttons and displays; CogTool, through its use of HTML web pages, allows for multiple interface states as multiple pages and state transitions as hyperlinks from one page to another. While such functionality could feasibly be built into Distract-R, we purposefully designed Distract-R to be as simple as possible without state transitions, and in our (albeit limited) experiences thus far, this has been sufficient for many real-world secondary-task interfaces.

DISCUSSION

Table 1 attempts to summarize our experiences with the four approaches in terms of required effort, theoretical fidelity, task generality, and prediction accuracy. (We have included the modified ACT-Simple* with operator arguments instead of the original ACT-Simple as a more interesting basis for comparison.) For required effort, clearly the full-fledged ACT-R architecture requires the most time in all aspects, namely coding the interface, writing the models, and running the simulations. ACT-Simple* allows for faster model creation but still requires a working interface, whereas

CogTool speeds up both interface specification (in HTML) and model creation. All three of the methods, however, rely on the original LISP simulation that runs in real time. Distract-R, by borrowing concepts from CogTool for interface design and model demonstration, and then augmenting these tools with a highly optimized model, allows the fastest overall model development cycle.

Theoretical fidelity rates how well the methods maintain the details of what is known about the various psychological abilities and limitations of human behavior, or at least as much as has been integrated into typical cognitive architectures. ACT-R, as a flagship cognitive architecture, incorporates numerous rigorous theories of human cognition, perception, and action. The other methods, by relying on the ACT-Simple framework, maintain some of this theory for perceptual-motor aspects — for example, they require preparation time for a motor movement and execution time based on the specific motor action; at the same time, the strictly sequential nature of an ACT-Simple model does not provide the full power of ACT-R in that ACT-R models can more highly parallelize (or avoid parallelism) depending on task domain and expertise. For cognitive fidelity, the ACT-Simple models rely on an aggregate mental operator and thus do not attempt to model more detailed cognitive processes (e.g., memory retrievals). Thus, only the ACT-R architecture can allow for models of "cognitive distraction" (e.g., Salvucci, 2002) in which distraction arises from a primarily cognitive task instead of a primarily perceptual-motor task.

Task generality rates the breadth of possible secondary-task interfaces that could be modeled using the approach. The ACT-R and ACT-Simple approaches both require a full-fledged working interface written in LISP; while tedious, this does allow for arbitrary complexity of the interface. CogTool provides a very flexible approach using HTML, which suffices to represent a wide variety of devices; the designer can build prototypes using HTML components (e.g., buttons), or can include images and "hot spots" on the image to provide functionality. As mentioned, interfaces designed in Distract-R are stateless, which limits its flexibility; again, however, we have found that even a stateless interface can adequately represent a variety of interfaces for purposes of evaluating potential for distraction.

Predication accuracy reflects how accurately the various approaches have accounted for human behavioral data. Perhaps surprisingly, all four have exhibited very similar prediction accuracies. The quantitative fits for the four approaches have been good for certain measures (e.g., time to complete secondary task) but not as good for other measures, primarily driving-specific measures (e.g., vehicle lateral deviation) that often depend on details of the simulation. However, the qualitative fits and rank-order predictions have generally been excellent for all the approaches. In one direct comparison, we (Salvucci et al., 2005) compared the predictions of the two extremes, namely ACT-R versus Distract-R; we found that both produced very good correlations to human data (R^2>.80) for three measures of performance, and also reasonable quantitative fits that could be improved by tuning only one (steering-related) parameter.

Given that their accuracies are roughly equivalent, one might wonder why we would use the more complex approaches like direct modeling in ACT-R. We suspect that for purposes solely of predicting driver distraction, the easiest approaches, especially Distract-R, serve well and provide the most benefit for the least cost. At the same time, for purposes of better understanding driver distraction and developing more rigorous theories of the underlying causes, clearly we desire as much theoretical "power" as is available — thus modeling with more complex methods such as ACT-R provide a more detailed picture of behavior. Overall, we believe that one tool does not suffice for all our modeling needs and desires; instead, we have found that a set of related tools representing different points along the speed – fidelity/generality continuum to be most useful for predicting driver distraction.

ACKNOWLEDGMENTS

This work was supported by Office of Naval Research grant #N00014-03-1-0036.

REFERENCES

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.

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

Hendricks, D. L., Freedman, M., Zador, P. L., & Fell, J. C. (2001). The relative frequency of unsafe driving acts in serious traffic crashes. Report for Contract No. DTNH22-94-C-05020, National Highway Traffic Safety Administration, U.S. Department of Transportation.

John, B. E., Prevas, K., Salvucci, D. D., & Koedinger, K. (2004). Predictive human performance modeling made easy. In *Human Factors in Computing Systems: CHI 2004 Conference Proceedings* (pp. 455 – 462). New York: ACM Press.

John, B. E., Salvucci, D. D., Centgraf, P., & Prevas, K. (2004). Integrating models and tools in the context of driving and in-vehicle devices. In *Proceedings of the Sixth International Conference on Cognitive Modeling* (pp. 130-135). Mahwah, NJ: Lawrence Erlbaum Associates.

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

Salvucci, D. D. (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.

Salvucci, D. D. (2005). A multitasking general executive for compound continuous tasks. *Cognitive Science*, *29*, 457-492.

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

Salvucci, D. D., Chavez, A. K., & Lee, F. J. (2004). Modeling effects of age in complex tasks: A case study in driving. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society*.

Salvucci, D. D., & Lee, F. J. (2003). Simple cognitive modeling in a complex cognitive architecture. In *Human Factors in Computing Systems: CHI 2003 Conference Proceedings* (pp. 265-272). New York: ACM Press.

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

Salvucci, D. D., Zuber, M., Beregovaia, E., & Markley, D. (2005). Distract-R: Rapid prototyping and evaluation of in-vehicle interfaces. In *Human Factors in Computing Systems: CHI 2005 Conference Proceedings* (pp. 581-589). New York: ACM Press.