Distract-R: Rapid Prototyping and Evaluation of In-Vehicle Interfaces

Dario D. Salvucci, Mark Zuber, Ekaterina Beregovaia, Daniel Markley
Department of Computer Science
Drexel University
Philadelphia, PA 19104
+1 215 895 2674
salvucci@cs.drexel.edu

ABSTRACT

As driver distraction from in-vehicle devices increasingly becomes a concern on our roadways, researchers have searched for better scientific understanding of distraction along with better engineering tools to build less distracting devices. This paper presents a new system, Distract-R, that allows designers to rapidly prototype and evaluate new in-vehicle interfaces. The core engine of the system relies on a rigorous cognitive model of driver performance, which the system integrates with models of behavior on the prototyped interfaces to generate predictions of distraction. Distract-R allows a designer to prototype basic interfaces, demonstrate possible tasks on these interfaces, specify relevant driver characteristics and driving scenarios, and finally simulate, visualize, and analyze the resulting behavior as generated by the cognitive model. The paper includes two sample studies that demonstrate the system's ability to account for effects of input modality and driver age on performance.

ACM Classification: H.5.2 [Information Interfaces and Presentation] User Interfaces – *evaluation/ methodology, theory and methods, prototyping*

Keywords: Cognitive modeling, driving, distraction

INTRODUCTION

As user interfaces move off the desktop into our everyday environments, one increasingly important issue arises in how we interact with these interfaces while performing others tasks — for example, taking notes on a PDA during a lively business meeting, or dialing a cell phone while crossing the street. One domain that has received a great deal of attention by the media, citizens, and researchers alike is driver distraction, or the possible inattention that arises when a driver performs some secondary task in the vehicle. Driver distraction and inattention is now the leading cause of vehicle crashes in the United States [4], motivating both an increased scientific understanding as to the origins of distraction and an increased engineering effort to design and build less distracting in-vehicle interfaces.

Tools for Predicting Driver Distraction

Because of the dangers associated with driver distraction, methods and tools that help us predict and mitigate distraction are valuable to in-vehicle interface design. However, such methods have been difficult to come by. Arguably, the best and most common method to date involves experimentation with a driving simulator, which minimizes any physical dangers while offering some sense of realism. At the same time, such empirical studies are time-consuming and expensive, and also require a working prototype of the new interface. Instead, we would prefer a rigorous predictive method that can evaluate distraction potential without the need for a physical device or driving simulator. Arguably the most well-known method, the so-called "15-second rule," states that interface tasks that require more than 15 seconds of total time are most likely to be distraction [3]. This rule, however, has been disputed as to its plausibility [19], and certainly at the very least ignores many clearly important factors such conflicting modalities between interface use and driving (e.g., a highly visual interface).

In an effort to further quantify the many intricacies of distraction, research has begun to explore the use of cognitive models in predicting distraction. Recent work [14] has utilized a cognitive model of driver behavior developed in the ACT-R cognitive architecture [1] for this purpose: the driver model, validated for basic driving maneuvers such as curve negotiation and lane changing, is then integrated with models of secondary-task behavior to predict resulting performance. This work has successfully accounted for the effects of distraction both from typical perceptual-motor tasks [12] and from primarily cognitive tasks [13]. However, this type of work requires a highly-trained cognitive modeler to analyze the task and create the model, and even such a modeler requires on the order of hours to produce a faithful model of behavior.

There have been recent strides to address the difficulties of modeling through a combination of rapid prototyping and modeling by demonstration. The most recent effort in this vein by John et al. [7] allowed designers (rather than modelers) to prototype new interfaces using HTML and demonstrate task behaviors on the resulting web pages, generating an ACT-R model that can interact with the same pages. While the original effort focused solely on graphical user interfaces on the desktop, this work is currently being extended to driving and in-vehicle interfaces [8, 16]. Because

CHI 2005, April 2–7, 2005, Portland, Oregon, USA.

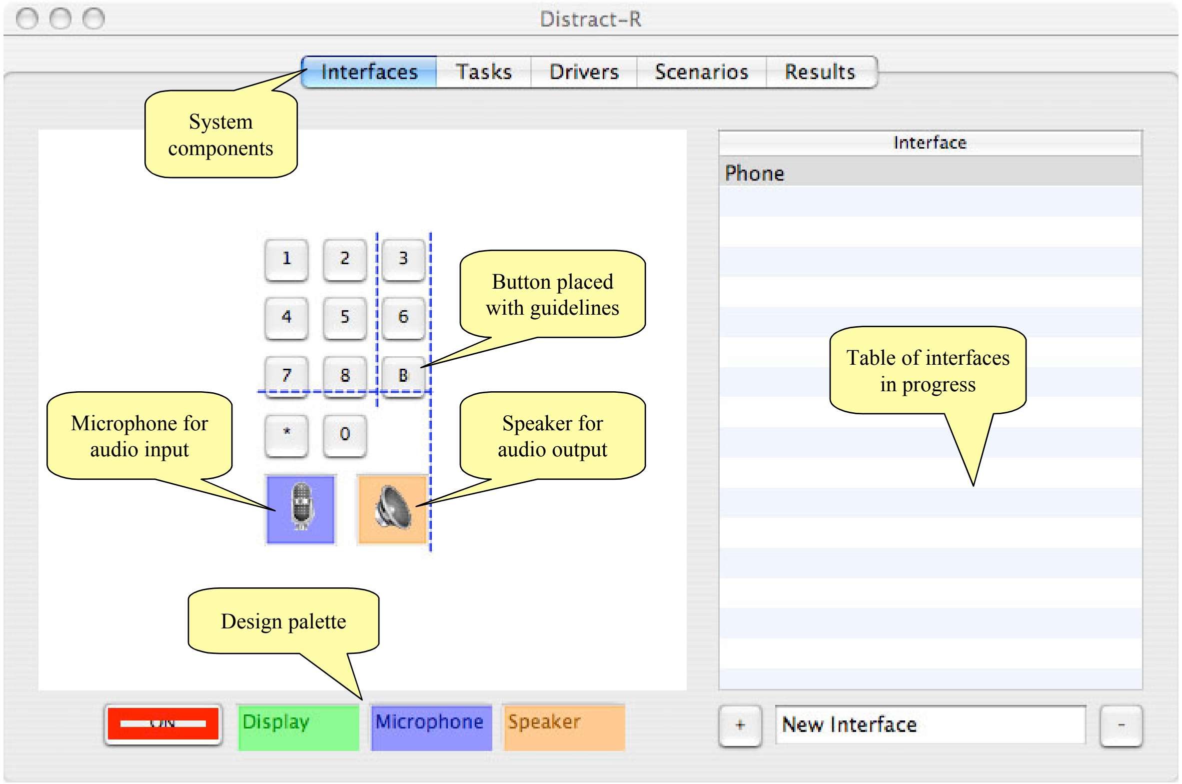


Figure 1: "Interfaces" panel for rapid prototyping of device interfaces.

these new approaches view driving as one possible application, though, they do not necessarily result in the most convenient system package for driving in particular — for instance, they require three separate applications (web browser, Java recorder application, and LISP environment) and run the driving simulation in real time, thus requiring the users wait many minutes to hours for reliable results.

An Integrated Tool: Distract-R

In this paper we describe a new system, Distract-R, intended to build on the successes of the above work and at the same time address some of their usability limitations. To this end, Distract-R at its core is a tool for prototyping and evaluating in-vehicle devices using cognitive models. However, instead of requiring users to build and test models, Distract-R uses cognitive models solely as an internal engine for generating predictions — no cognitive modeling knowledge or experience is necessary on the part of the user. In addition, Distract-R is a single integrated application that incorporates all the necessary components for prediction, including a prototyping tool, modeling engine, and driving simulator. The tight integration and optimized simulation helps Distract-R generate extremely fast predictions of performance — roughly 100 times faster than previous tools — and thus allows designers to iterate on the design cycle much more quickly and easily.

THE DISTRACT-R SYSTEM

The ultimate goal of the Distract-R system is to provide a predictive tool for understanding driver distraction and building less distracting devices, accounting as much as possible for the wide range of potential interfaces, tasks, driver characteristics, and driving scenarios. To this end, Distract-R utilizes an underlying cognitive architecture, ACT-R [1], that helps ensure psychological plausibility and at the same time provides a computational framework in which to build models of behavior. The system also incorporates recent work on simpler modeling frameworks for rapid modeling, namely the ACT-Simple framework [17] that uses ACT-R as its computational engine. In addition, Distract-R incorporates the ACT-R driver model [14] that has been independently validated over a number of studies [e.g., 12, 13, 14, 15]. Nevertheless, Distract-R abstracts over the details of the cognitive models and makes them (almost) transparent in the system, such that a user can, if desired, ignore many of the underlying modeling details and still accomplish a great deal with the system. Thus, Distract-R is *not* intended for cognitive modelers, but rather is intended for any designer or engineer who is part of the in-vehicle design process, particularly those (in the majority) with no prior experience in cognitive modeling.

Distract-R has five major components that allow a designer to specify various aspects of the interface and task and then analyze the resulting behaviors. The next five sections describe each of these components, following a logical progression through the steps a typical user might take in a single iteration through the evaluation process. Distract-R is currently a Macintosh-based application; development of a web-based system is also underway.[1]

---

[1] See http://hcil.cs.drexel.edu/distract-r/

Interfaces: Rapid Prototyping

The first component of Distract-R involves a straightforward rapid prototyping system in which a designer can quickly specify new device interfaces. This component aims to provide a “quick-and-dirty” way to sketch a new interface with as little overhead as possible, with a primary emphasis on the different modalities used in typical in-vehicle interfaces. Figure 1 shows the “Interfaces” panel of the application as a user prototypes a typical phone interface. On the left side of the window, the user clicks one of the four buttons on the lower palette to select different components for building the interface. The four possible components include:

- A **Button** (highlighted on the far left) as might appear on a phone, radio, etc.
- A **Display** that shows visual information (e.g., a radio display showing the current station)
- A **Microphone** that allows the user to provide speech input to the interface
- A **Speaker** that allows the interface to provide sound or speech output to the user

When the user clicks on one of the palette items and moves the cursor to the interface panel (center left), the cursor changes to the selected component and the user can place the component anywhere on the panel. To facilitate rapid layout, Distract-R provides guidelines that visually align a new component with others on the interface, at the same time allowing the new component to “snap” into proper alignment. The figure shows a user placing a new phone button and illustrates the guidelines that appear in the process.

The table on the right side of the “Interfaces” panel lists all interfaces currently in progress. As might be expected, the “+” button adds a new interface to the list, and the “–” button removes the currently selected interface. The user can thus rapidly switch back and forth between many interfaces, sketching several interfaces in minutes to prepare for downstream evaluation.

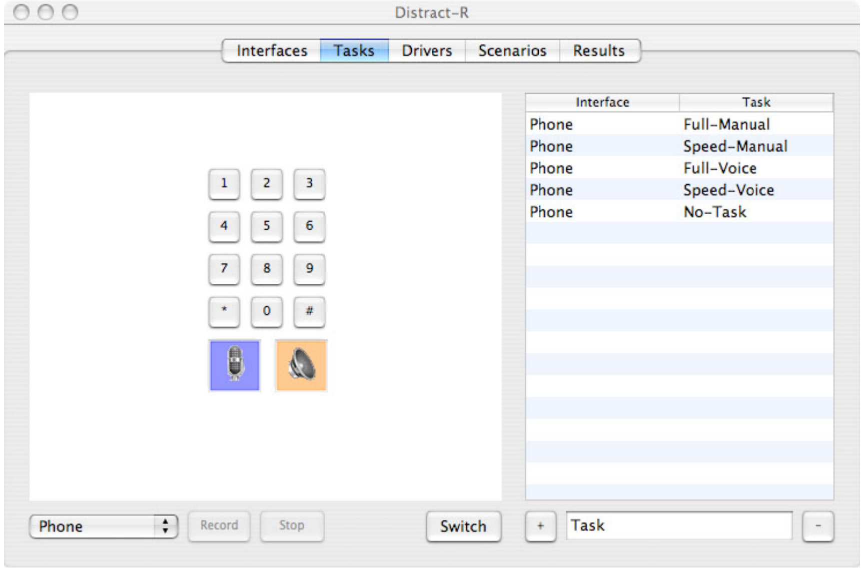
The prototyping tool as currently designed is fairly minimal, and does not offer the many features of some other prototyping systems that utilize HTML [7] or pen-based sketching [9]. However, it does serve its purpose as a simple prototyping tool for in-vehicle interfaces in two ways. First, the system allows for a very fast working interface on which a user can demonstrate sample tasks (described next). Second, the system focuses on one of the most critical aspects of the distraction potential of new devices, namely what modalities are utilized by the interface (manual, visual, speech, and aural) and how these might be combined to build less distracting device interfaces.

Tasks: Modeling by Demonstration

The second component of Distract-R allows a user to demonstrate task behaviors on the prototyped interfaces, resulting in the creation of cognitive models for these task behaviors. As noted earlier, however, the user does not need to understand the underlying models to manage the system (though such an understanding may of course help); instead, all “modeling” is done through task demonstration, an approach similar to that used in recent tools [5, 7; see 10, 18 for other approaches]. Figure 2 shows the “Tasks” panel of the application as a user demonstrates possible task behaviors on the prototyped phone interface. On the left, the user selects which interface the task is associated with, then adds a new task on the right-hand table and clicks the “Record” button. Thereafter, the user can click on the components of the interface, and the system stores these actions until the user clicks the “Stop” button.

To translate user-demonstrated tasks to predictions of distraction, we must map the demonstrated action sequence into a computational cognitive model. For model specification, Distract-R uses ACT-Simple [17], a modeling framework that “compiles” to ACT-R production rules and thus runs in simulation to generate behavioral predictions; this approach has recently been exploited by John et al. [7] in their use of the CogTool system for in-vehicle interfaces. The original ACT-Simple focused on behavioral actions in graphical user interfaces (mouse pointing, keystrokes, etc.); instead, we utilize an augmented framework for off-the-desktop interface actions (hand movement, button pressing, etc.) [see 8, 16]. In addition, to make the integrated system as fast as possible, we re-implemented a relevant subset of both ACT-R and ACT-Simple within Distract-R; the resulting (C++) code runs approximately 100 times faster than the original (LISP) code. This simplification was intended to strip out parts of the modeling systems less relevant to our immediate driving application (e.g., ACT-R’s rigorous account of memory recall and learning) and focus on building a useful approximation of the most relevant features (e.g., the basic production system with 50 ms rule firings). The resulting framework actions include:

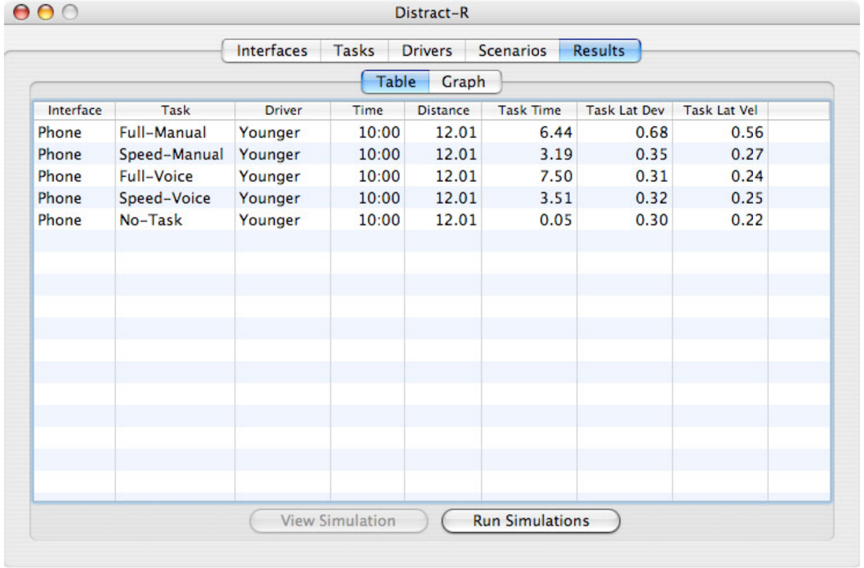
- (*press-button* <loc>): initiate a button press as a hand movement to the target with a pointed index finger, requiring 100 ms preparation time plus execution time determined by Fitts’s law (using ACT-R defaults [1])
- (*move-hand* <loc>): initiate a hand movement to or from the steering wheel, with execution time of 610 ms [6]
- (*look-at* <loc>): find and initiate visual encoding of a location, with assumed execution time of 150 ms
- (*speak*): initiate a speaking action, with assumed execution time 300 ms (as in previous models [12])
- (*listen*): initiate listening to a phrase, with assumed duration time 300 ms (same as for speech)
- (*think*): cognitive operator, with assumed duration of 1250 ms (= 1350 ms [2] – 100 ms visual)



Interface	Task
Phone	Full-Manual
Phone	Speed-Manual
Phone	Full-Voice
Phone	Speed-Voice
Phone	No-Task

Figure 2: “Tasks” panel for task demonstration.

Figure 3: Window for viewing real-time simulation.



Interface	Task	Driver	Time	Distance	Task Time	Task Lat Dev	Task Lat Vel
Phone	Full-Manual	Younger	10:00	12.01	6.44	0.68	0.56
Phone	Speed-Manual	Younger	10:00	12.01	3.19	0.35	0.27
Phone	Full-Voice	Younger	10:00	12.01	7.50	0.31	0.24
Phone	Speed-Voice	Younger	10:00	12.01	3.51	0.32	0.25
Phone	No-Task	Younger	10:00	12.01	0.05	0.30	0.22

Figure 4: “Results” panel with table of interfaces, tasks, drivers, and simulation results.

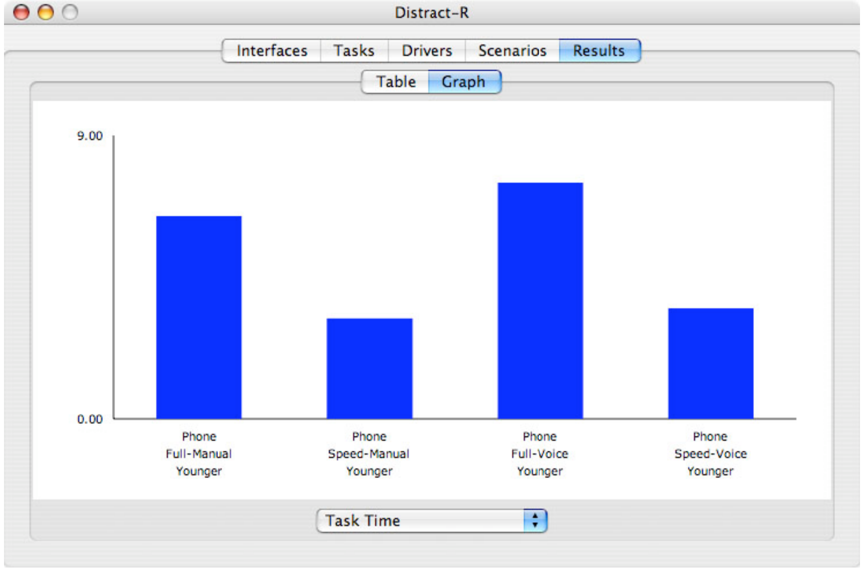


Figure 5: “Results” panel with graphical display of simulation results.

The system maps the demonstrated actions to a model as follows. A *move-hand* operator first brings the driver's hand to the device interface. Then, any click on an interface button is mapped to a *look-at* + *press-button* operator sequence; any click on a microphone is mapped to a *speak* operator (i.e., the user is speaking into the microphone); and any click on a speaker is mapped to a *listen* operator (i.e., the interface is creating sound output to the user). A sample ACT-Simple model resulting from task demonstration is shown in Table 1 in the upcoming section on application Study 1.

Because the tasks are performed while driving, we assume that drivers occasionally need to switch back to the primary driving task. Inference of these switch points, however, is extremely difficult or impossible given a minimal description of the interface and only an action sequence without goal structure. Thus, Distract-R allows a user to manually specify switch points where desired through use of a special "Switch" button: during demonstration, a click on this button indicates that the current point in the sequence serves as a natural switching point back to driving. This approach mimics the "Monitor Driving" button in the CogTool augmented with the ACT-R driver model [8]. These switches create a note in the resulting ACT-Simple model to "pop" back to driving after performing the previous operator.

For cognitive (or mental) operators, Distract-R does not attempt to infer where such operators would appear [cf. 7]. Instead, it inserts cognitive operators in the form of (*think*) only at switch points immediately after a button press: because the button press requires visual confirmation (we ignore the possibility of pressing "by feel" for now), the system assumes that the switch back to driving interrupts the secondary task model such that a *think* is needed to recall and resume the previous goal. Further attempts to infer the placement of cognitive operators — a well-known critical issue in keystroke-level modeling [2, 7] — is left for future work. In addition, the total *think* time of 1250 ms is actually divided into three equal-sized units of 417 ms with switch points between them, modeling the fact that cognitive time can be interleaved with the primary driving task (as derived in [16]).

Drivers: Specifying Individual Variability

The third component of Distract-R allows a user to specify characteristics of the individual driver(s) for testing of effects of individual differences on performance. The theory behind how individual differences map to cognitive models and architectures is currently very incomplete. However, we attempt to exploit two particular differences that recent efforts have identified as important sources of effects: driver age and driver "steering style."

Age can sometime have significant effects on driver performance, both from a "hardware" point of view (e.g., slowing of motor processes) and from a "software" point of view (e.g., strategic variability). In the particular context of the ACT-R driver model, recent work [15] has shown that age can be reflected in the cognitive architecture by scaling of the cognitive cycle time, namely scaling the normal cycle time up by 13%. Distract-R utilizes this result by allowing the user to specify several drivers with different ages to test differential effects on performance. The interface allows for any values between 20 and 70 years of age; however, the system currently maps any age below 45 to "younger" and above to "older," since the theory has only been validated to this extent. The system thus includes the potential for generalizing these settings in the future as more is known about architectural effects of age to that level.

"Steering style" represents a driving-specific parameter that indicates the aggressiveness with which a driver steers — that is, whether the driver prefers to steer aggressively toward the lane center after the car drifts, or whether s/he prefers to slowly bring the car back to center. This particular aspect of steering style maps nicely to the ACT-R driver model, which has three parameters that control its desire to move to lane center. The "style" control on Distract-R's interface currently scales all three parameters by a constant value: the center represents a default value of 1.0, and the left and right end points represents scalings of 0.5 and 1.5, respectively. Adjustment of this value can help users produce better quantitative fits to empirical results if desired, as shown in the applications section later in the paper.

Scenarios: Specifying Simulation Environments

The fourth component of Distract-R involves selection of the driving environment for model simulation — that is, the environment in which the model driver will navigate and occasionally perform secondary tasks on the interface. We currently have implemented only one possible simulation environment, a construction-zone environment that forces drivers onto a single lane behind a lead car. The full "Scenarios" panel offers the option for generalizing this component later to encapsulate a fuller range of possible environments, ideally from the simplest car-following task to a full-blown multi-lane highway environment with signs, heavy traffic, and perhaps additional hazards.

Results: Simulation and Visualization

The final component of Distract-R consists of the driver simulation, visualization, and analysis tools. First, when the user switches to the "Results" panel, the system takes all specified interface-task pairs and drivers and creates a table of all possible combinations. To watch the driver model navigate the environment in real time, the user can select one combination from the table and press a "View Simulation" button. The system then launches the simulation in a new window, as shown in Figure 3, and also indicates the model's current eye position on-screen. This provides users with a visualization tool such that they can examine sample protocols and get a sense of how the interface may impact the driver's performance.

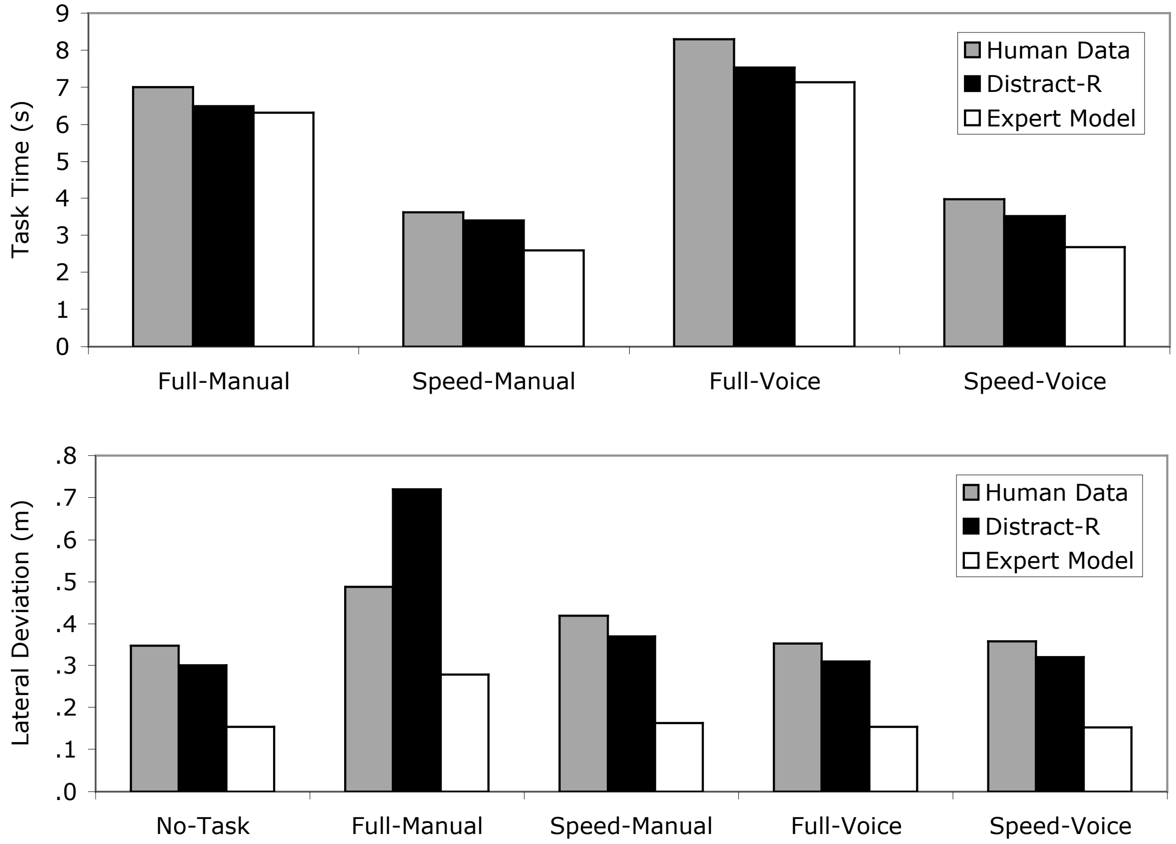


Figure 6: Task time and lateral deviation results for the Study 1 task conditions.

To gather aggregate measures without viewing, the user can press the "Run Simulations" button and run each combination sequentially to fill in a table of relevant measures, shown in Figure 4. The table currently includes the measures of average time to complete the secondary task ("Task Time"), root-mean-squared lateral deviation from lane center ("Task Lat Dev"), and average absolute lateral velocity ("Task Lat Vel"); the last two measures, common in empirical studies of driver behavior, represent indicators of vehicle stability and thus can be interpreted as steering performance. As we noted earlier, the original driver model ran in approximately real time; in stark contrast, Distract-R can run 10 minutes of simulated driving in approximately 3 seconds, greatly speeding up the evaluation process and getting near-immediate predictions of performance.

To view and compare simulation results for specific measures, Distract-R includes a graphing tool illustrated in Figure 5. From the pop-up menu on the bottom, the user selects a particular measure of performance, and the system then updates the view to display the results for all interface-task-driver combinations. Both axes scale automatically according to the number of combinations as well as the largest values in the graph.

SAMPLE APPLICATION STUDIES

We have applied the Distract-R system in two studies to demonstrate the validity and usefulness of this approach. The following two studies examine how the system can help predict potential effects of driver distraction as arise in differences in input modality (Study 1) and as arise in younger versus older drivers performing a secondary task (Study 2). The current studies are based on previous work in which the tasks were modeled by an expert modeler. In contrast, we illustrate here how Distract-R enables a designer with no background in cognitive modeling to produce the same results quickly and easily.

Study 1: Effects of Interface Input Modality

The first study explored the effects of different input modalities on driver performance. In an empirical study of phone dialing while driving in a fixed-base simulator, Salvucci [12] examined four methods of dialing a mounted phone: *Full-Manual* dialing by entering all 7 digits of a memorized phone number, *Speed-Manual* dialing by entering a 1-digit shortcut to the desired number, *Full-Voice* dialing by speaking all 7 digits and hearing audio confirmation of recognition, and *Speed-Voice* dialing by speaking a single-phrase shortcut for the number. All interfaces required drivers initially to turn the phone on with

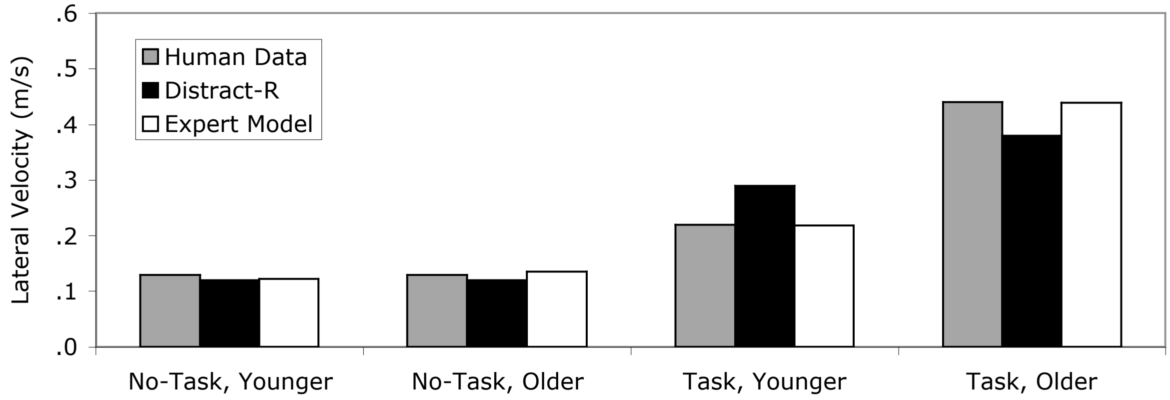


Figure 7: Lateral velocity results for the Study 2 task conditions.

one keypress; the manual interfaces required drivers to press a "Send" key after dialing, while the voice interfaces provided an audio confirmation of the sending process. The empirical study found that manual dialing interfaces produced significant effects of distraction while voice dialing interfaces produced little to no effect [see 12].

We used Distract-R to prototype the four dialing methods as four distinct tasks on the same interface. The basic phone interface comprised a standard keypad as well as a microphone and speaker for speech input and output, namely that shown in Figure 1. We then demonstrated the four methods as tasks on this interface assuming that drivers switched back to driving after turning on the phone, after speaking or listening (since they do not require visual attention), and, for the *Full* interfaces, after dialing 3 of the 7 digits; we also demonstrated a fifth place-holder null task (i.e., no operators) to make predictions for the no-task condition. Table 1 shows resulting model for the Speed-Voice condition, with asterisks indicating points at which the model switches to driving. Note that, as mentioned earlier, the *think* operator is in fact divided into three blocks of 417 ms and switches to driving between the blocks to model interleaving of cognitive operations.

Table 1: ACT-Simple model for the Speed-Voice condition as generated by task demonstration.

Operator	Description
(*move-hand* device)*	Move hand to device
(*look-at* "On")	Look at On button
(*press-button* "On")*	Press On button
(*think*)	Cognitive operator
(*speak*)*	Say callee's name
(*listen*)*	Hear callee's name
(*listen*)*	Hear send confirmation

Results were compiled with Distract-R's default settings of 10-minute simulations for each task running a task trial every 20 seconds. The entire process as performed by the authors, from prototyping to task demonstration to full simulation, required a total of **90 seconds** — 75 seconds for prototyping and demonstration, and 15 seconds for all simulation runs. Of course, in this case the design and tasks were known in advance as opposed to being invented on the fly; nevertheless, this result clearly demonstrates that Distract-R's interface itself is not an obstacle to specifying and evaluating new devices.

Figure 6 shows the human data, Distract-R predictions, and original expert model predictions [12] for two measures. To fit the empirical data, we used a younger driver (like the original study) and manipulated only the "steering style" parameter, estimated value 0.55. The task-time measure shows the time needed to complete the tasks for each dialing method. Distract-R (R^2>.99, *RMSE*=.53) and the expert model (R^2=.99, *RMSE*=1.07) are both fairly accurate albeit slightly on the low side for most conditions. (Note that Distract-R currently has no randomness in the system and thus repeated runs produce the same results.) Importantly, Distract-R predicts the same rank order of interfaces as the human data, namely that the *Full-Voice* condition requires the most time and the *Speed-Manual* condition the least time. The lateral-deviation measure shows how far the vehicle deviated from the center of the lane, calculated as a root-mean-squared error. Distract-R (R^2=.88, *RMSE*=.11) again predicts the main effects well, in particular the large effect for the *Full-Manual* condition, a smaller effect for *Speed-Manual*, and little to no effect for the voice conditions. The expert model (R^2=.83, *RMSE*=.21) does not fare quite as well; in fairness, however, the original study explored *a priori* predictions with no parameter estimation, and thus the expert model could very likely be adjusted for a better fit with such an estimation.

Study 2: Effects of Driver Age

The second study explored the effects of age on driver performance, with and without a secondary task. Reed and Green [11] asked younger (approx. age 20-30) and older (approx. age 60-70) drivers to dial a 10-digit phone number (including area code) while recording various behavioral measures. Salvucci, Chavez, and Lee [15] modeled this experiment with hand-crafted ACT-R and ACT-Simple models. The original experimental results, also achieved by the expert model, showed younger and older drivers to have roughly the same performance without a task but reduced performance with a task. In addition, older drivers exhibited a much larger effect of task than younger drivers.

Again we utilized Distract-R for rapid prototyping and demonstration, this time designing only the phone keypad (i.e., no microphone or speaker) and demonstrating the task of dialing 10 digits. For this study, the entire process required **60 seconds** — 45 seconds for prototyping and task demonstration, and 15 seconds for simulation. As in Study 1, results were compiled with Distract-R's default 10-minute simulations, and the "steering style" parameter was re-estimated to a best-fitting value of 0.65. (The small difference in parameter values between studies is not especially surprising given that the empirical studies were conducted with two different driving simulators with different simulated vehicle dynamics; in general, we expect the same parameter values to apply given the same empirical setup and similar driver populations.)

Figure 7 shows the human data, Distract-R predictions, and expert model predictions [15] for Reed and Green's lateral-velocity measure. Both Distract-R (R^2=.87, *RMSE*=.05) and the expert model (R^2>.99, *RMSE*<.01) exhibited no effect of age for the no-task condition. In the task condition, the models both produced effects of task and the corresponding interaction with age — namely, that the performance of older drivers is hampered more than that of younger drivers. The expert model more closely captures performance in the Task conditions, but nevertheless Distract-R does very well in accounting for the basic effects qualitatively and quantitatively.

FUTURE DIRECTIONS

As a predictive tool for designing less distracting in-vehicle interfaces, Distract-R is a work in progress. The application does not focus on any one component in the design process, but rather emphasizes the integration and ease of flow between components, allowing a user to quickly move from new designs to quantitative predictions. Each component of the system could be more fleshed out in the future, requiring both theoretical advances as well as new engineering tools. For instance, incorporation of other forms of driver variability, such as fatigue or drunkenness, would require new theories of how such factors map onto the cognitive architecture. As soon as such theories can be realized, they can be easily embedded into the system to provide both a more general and more rigorous account of driver behavior and distraction.

This paper has shown that Distract-R provides a good engineering approximation to several aspects of distraction as demonstrated in the two application studies. Nevertheless, validation of an application such as Distract-R may take many forms. The accuracy of the driver and task models have the most significant impact on predictions — that is, for the system to generate reliable predictions, these models must have as much psychological validity and plausibility as possible. Fortunately, the driver model has been independently validated for basic measures of performance [14], and the integrated-model approach used here has also been demonstrated successfully in a number of studies [e.g., 12, 13]. In addition, ACT-R itself has undergone many validation efforts [e.g., see 1], and ACT-Simple has proven reliable as a simpler, approximate modeling framework [17]. Applications of the modeling frameworks, driver model, and Distract-R system to additional domains will continue to provide validation for the proposed approach and/or suggest ways in which the system should be revised and improved.

ACKNOWLEDGMENTS

This work was supported in part by grants from the Office of Naval Research (#N00014-03-1-0036) and Ford Motor Company to the first author.

REFERENCES

1. Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Hillsdale, NJ: Erlbaum.
2. Card, S. K., Moran, T.P. and Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
3. Green, P. (1999). The 15-second rule for driver information systems. In *ITS America Ninth Annual Meeting Conference Proceedings*. Washington, D.C.: Intelligent Transportation Society of America.
4. 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.
5. Hudson, S. E., John, B. E., Knudsen, K., and Byrne, M. D. (1999). A tool for creating predictive performance models from user interface demonstrations. *CHI Letters 1* (1), 93-102.
6. IVIS DEMAnD Prototype Software User's Manual, pub. No. 00-136, USDOT, Aug. 2000, http://www.tfhrc.gov/humanfac/00-136.pdf.
7. 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.
8. 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*.
9. Landay, J. A., & Myers, B. A. (1995). Interactive sketching for the early stages of user interface design. In *Proceedings of Human Factors in Computing Systems: CHI 95* (pp. 43-50). New York: ACM Press.
10. Paganelli, L., & Paternò, F. (2003). A tool for creating design models from web site code. *International Journal of Software Engineering and Knowledge Engineering*, *13*, 169-189.
11. Reed, M. P & Green, P. A. (1999). Comparison of driving performance on-road and in a low-cost driving

simulator using a concurrent telephone dialing task. *Ergonomics*, *42*, 1015-1037.

12. 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.
13. 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: Erlbaum.
14. Salvucci, D. D. (to appear). Modeling driver behavior in a cognitive architecture. *Human Factors*.
15. Salvucci, D. D., Chavez, A. K., & Lee, F. J. (2004). Modeling effects of age in complex tasks: A case study in driving. To appear in *Proceedings of the 26th Annual Conference of the Cognitive Science Society*.
16. Salvucci, D. D., John, B. E., Prevas, K., & Centgraf, P. (2005). Rapid development of keystroke-level models for on- and off-the-desktop interfaces. Manuscript in preparation.
17. Salvucci, D. D., & Lee, F. J. (2003). Simple cognitive modeling in a complex cognitive architecture. In *CHI 2003 Conference Proceedings* (pp. 265-272). New York: ACM Press.
18. St. Amant, R., & Ritter, F. E. (2005). Specifying ACT-R models of user interaction with a GOMS language. *Cognitive Systems Research*. *6*, 71-88.
19. Tijerina, L., Johnston, S., Parmer, E., Winterbottom, M.D., and Goodman, M. (2000). Driver Distraction with Route Guidance Systems (Technical Report DOT HS 809 069), East Liberty, OH: National Highway Traffic Safety Administration.