Providing User Models Direct Access to Interfaces: An Exploratory Study of a Simple Interface With Implications for HRI and HCI

Frank E. Ritter, *Member, IEEE*, Dirk Van Rooy, Robert St. Amant, and Kate Simpson

***Abstract*—Models of users are a way to understand and improve the usability of computer interfaces. We present here a model in ACT-R cognitive-modeling language that interacts with a publicly available driving simulation as a simple analog for robot interfaces. The model interacts with the unmodified Java interface by incorporating a novel use of bitmap parsing. The model's structure starts to describe the knowledge a human operator of a robot must have. The model also indicates some of the aspects of the task will be difficult for the operator. For example, the model's performance makes quantitative predictions about how robot speed will influence navigation quality, correlating well to human performance. While the model does not cover all aspects of human–robot interaction, it illustrates how providing user models access to an interface through its bitmap can lead to more accurate and more widely applicable model users.**

***Index Terms*—Graphical user interfaces, human–computer interaction (HCI), human–robot interaction (HRI), image processing, telerobotics, user interface human factors, user modeling.**

I. Introduction

In the future, some robots may become autonomous and solve practical problems completely independent of human supervision. However, such a level of independence has not yet been achieved and is in some cases simply undesirable. Many of the tasks that robots face today, such as exploration, reconnaissance, and surveillance, will continue to require supervision by humans [1].

Consider the domain of robot-assisted urban search and rescue (USAR), which involves the detection and rescue of victims from urban structures like collapsed buildings. It might be optimal for the robot to exhibit a fair amount of autonomy in some situations, for example, in navigating in a confined space using its own sensors. Other situations, however, require human intervention; for example, an operator may need to assist a robot in freeing itself because its sensors do not provide enough information for autonomous recovery. Yet other situations, such as providing medication for trapped survivors, requires human control for legal reasons. These situations illustrate the need for flexibility in the interaction between a robot and its operator. The operator may dynamically shift between directly controlling the robot, communicating with it, monitoring its actions and sensors, and diagnosing its general behavior—a demanding role, especially in the critical situations in which human–robot interaction (HRI) is often called for [1], [2]. Thus, understanding how operators use robots will remain a problem for some time, and is of interest to robot developers [2]–[7].

Despite its importance, a general theory of HRI is not yet available, although steps are being taken in this direction [4]–[6]. There are a number of reasons for the lack of theory. First, the task domains for many robot systems are complex. In USAR, a representative domain, tasks can be so difficult that neither humans nor robots can solve problems alone (e.g., [7]).

Second, human–robot task domains are diverse. The requirements for interaction with a robot chemist, a manufacturing robot, and an entertainment robot, are very different from each other. General principles for interaction with these different kinds of robots and their tasks remain to be developed. The level to which the use of robots will be integrated in society, it has been argued, will be largely dependent on the robots' ability to communicate with humans in understandable and friendly ways [8].

Third, many human–robot tasks are stressful to the operator, as exemplified by USAR tasks [3]. Stress is known to influence performance on tasks in general (e.g., [9]). Explaining how the effectiveness of HRI degrades with stress and other factors will be an important requirement for a theory of HRI.

Currently, because robots are not nearly as common as commercial software packages, their controller software is usually not built by people trained in interface design. Usability problems have already begun to surface, and will worsen with time unless steps are taken to provide guidance to designers.

One way to provide guidance is to create a model operator. We introduce a methodology in which a cognitive model representing a single user autonomously interacts with a simple robot interface analog. Below, we describe a user model based on a cognitive-modeling architecture and a dynamic simulation environment in which we have exercised the user model to explore issues relevant to HRI. Our quantitative tool is in the form of a simulated user that identifies some of the problems associated

Manuscript received September 22, 2003; revised July 2, 2004. This work was supported by the Space and Naval Warfare Systems Center-San Diego, under Grant N66001-1047-411F and by the Office of Naval Research under Contract N00014-03-1-0248. The content does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred. This paper was recommended by Associate Editor A. A. Maciejewski.

F. E. Ritter and K. Simpson are with the School of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: frank.ritter@psu.edu; kate@pinkmosstea.com).

D. Van Rooy is with the Department of Psychology, Birmingham University, Birmingham B15 2TT, U.K. (e-mail: D.VanRooy@bham.ac.uk).

R. St. Amant is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA (e-mail: stamant@eos.ncsu.edu).

Digital Object Identifier 10.1109/TSMCA.2005.853482

1083-4427/$20.00 © 2006 IEEE

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

with robot interface use and gives insight into the issues that make robot interfaces difficult to use. We also introduce a solution to a serious impediment to work in this area, broadly defined, how to provide user models access to interfaces. We next report an experiment we have run to compare human performance against model performance to validate the model and to show where the model can be improved.

We end with a list of observations about why this model suggests HRI is difficult and a list of recommendations based on this model for improving HRI. Our work indicates ways to improve such interfaces and explicates initial aspects of interaction that can serve as motivations for a general theory of HRI.

II. User Model

A cognitive model is a theory of human cognition realized as a running computer program. It produces humanlike performance in that it takes time to complete each task, commits errors, deploys strategies, and learns. It provides a way of applying cognitive psychology data and theory to HRI and human–computer interaction (HCI) problems in real time and in an interactive environment (e.g., [10]–[16]).

We have developed a model user consisting of the ACT-R cognitive architecture [17] and a simulated-eyes-and-hands system, SegMan [18]–[20]. SegMan allows the model to interact with many types of interfaces because it recognizes objects based on parsing the screen's bitmap. This approach is more general than modifying interfaces to work with models directly or instrumenting interface toolkits [15], [21], [22]. We start the explanation of the modeling system by describing the cognitive architecture.

A. *ACT-R Cognitive Architecture*

The ACT-R cognitive architecture [17, information, including a tutorial, is also available at act.psy.cmu.edu] implements and integrates many regularities of human cognition, including theories of cognition [17], visual attention [23], and motor movement [10], [15]. It has been applied successfully to higher level cognitive phenomena, such as modeling driving [12], differences in working memory [24], and skill acquisition [25]. Recently it has been used successfully to explore a number of interface issues including visual search on menu selection [15] and the effect of interface design on strategy choice [14], [26]. We chose ACT-R because it is based on a theory of human cognition, allows the prediction of behavior time, and can be extended to include further aspects of human behavior, both by adding task knowledge as well as modifying the architecture.

ACT-R makes a distinction between two types of long-term knowledge, declarative and procedural. Declarative knowledge is factual and holds information like "$2 + 2 = 4$." The basic units of declarative knowledge are chunks, which are schema-like structures, effectively forming a propositional network. Procedural knowledge consists of production rules that encode skills and take the form of condition–action pairs. Production rules apply to specific goals or subgoals, and mainly retrieve and change declarative knowledge.

Besides the symbolic procedural and declarative components, ACT-R also has a subsymbolic system that modifies the use of the symbolic knowledge. Each symbolic construct, be it a production or chunk, has subsymbolic parameters associated with it that reflect its past use. In this way, the system keeps track of the usefulness of the knowledge. The knowledge available in the declarative memory module is determined by the probability that a particular piece of knowledge will be used in a particular context.

An important aspect of models created in the ACT-R system is that they predict human behavior including the timing of actions. Each covert step of cognition (e.g., production firing, retrieval from declarative memory) or overt action (mouse click, moving attention) has a latency associated with it that is based on psychological theories and data. For instance, firing a production rule takes about 50 ms (modulated by other factors such as practice), and the time needed to move a mouse to a location on a computer screen is calculated using Fitts' law [27]. In this way, the system provides a way to apply summaries of human behavior from psychology to generate behavior as predictions and for use in applications.

A schematic of the current implementation of the ACT-R 5.0 architecture which we used is shown in Fig. 1. At the heart of the architecture is a production system that includes a number of buffers. These buffers represent the information that the system is currently acting on: The goal buffer contains the present goal of the system; the declarative buffer contains the declarative knowledge that is currently available; and the perceptual and motor buffers indicate the state of the perceptual and motor modules (busy or free, and their contents). The communication between central cognition and the buffers is performed by applying production rules.

Production rules are condition–action pairs: The first part of a production rule, the condition side, typically tests if certain declarative knowledge (in the form of a chunk) is present in a given buffer. The second part, the action side, then sends a request to a buffer to change the current goal, retrieve knowledge from a buffer such as declarative memory, or perform some action. The production system and buffers run in parallel, but each component is itself serial. The shaded areas indicate the novel functionality provided by SegMan that overrides the original perceptual–motor functionality of ACT-R 5 (ACT-R/PM), which is indicated by the dashed lines.

The SegMan vision and motor modules (explained below) allow the model to "look" at an interface and manipulate objects in that interface. The vision module builds a representation of the display in which each object is represented by a feature. Productions sent to the visual buffer can direct attention to an object on the screen and can create a chunk in declarative memory representing that object and its screen location. The production system can then send commands, initiated by a production rule, to the motor module that manipulate these objects. Central cognition and the various buffers run in parallel, but the perceptual and motor modules are serial (with a few rare exceptions) and can only contain one chunk of information. Thus, the production system might retrieve a chunk from declarative memory, while the vision module moves visual attention and the motor module moves the mouse.

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

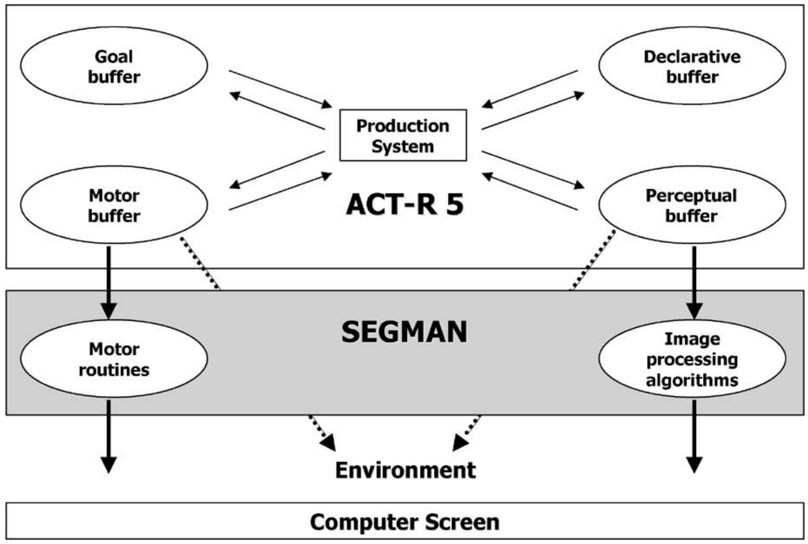


Fig. 1. SegMan+ACT-R 5 system.

B. Extending ACT-R 5 With Segman

ACT-R 5 interacts with interfaces using ACT-R/PM, its perceptual and motor components which includes tools for creating interfaces and annotating existing interfaces created in Macintosh Common Lisp (MCL) so that models can see and interact with objects in the interface. This allows most models to interact in some way with most interfaces that are written in MCL and lets all models interact with all interfaces written with the special tools included with ACT-R/PM.

We extended ACT-R/PM to provide it with direct access to any interface under Windows, thus removing dependencies on interface creation tools. This was done by including the SegMan suite (www.csc.ncsu.edu/faculty/stamant/cognitive-modeling.html). SegMan takes pixel-level input from the screen (i.e., the screen bitmap), runs the bitmap through image-processing algorithms, and builds a structured representation of the screen. This representation is then passed to ACT-R. ACT-R/PM can then moderate what is visible and how long it takes to see and recognize objects.

SegMan can also generate mouse and keyboard inputs. This functionality is integrated with the ACT-R/PM theory of motor output. Including SegMan extends ACT-R/PM to pass output to any Windows interface.

These simulated eyes and hands have been used to explore interfaces [20], [28] and to dial on-screen telephone displays written in Tcl/Tk [16] and images found on the Internet [18]. In each of these cases, the model used the interface directly—the modeler did not have to modify the interface to allow the model to interact with it.

C. The Task

We chose a driving task as a simple example of an HRI task to test our approach. Models of driving (e.g., [12], [29]–[31]) have many similarities with HRI, particularly teleoperation, a common HRI task [32] (this choice of type of task as a simple analog task to explore human–robot interfaces was based on discussions with two prominent robotics researchers, R. Murphy and L. Parker). While this task is not as complex as full robot interaction, progress with it is a subset of progress on HRI.

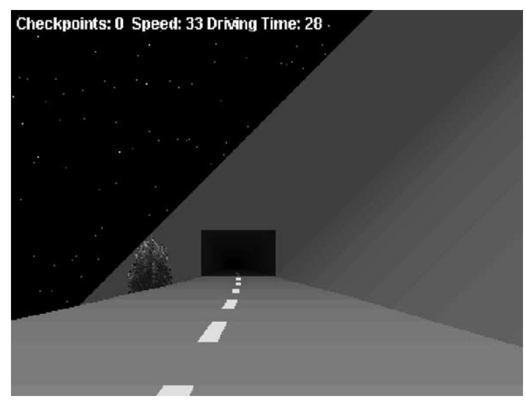


Fig. 2. A snapshot of the 3D Driver environment.

The driving environment, 3D Driver, a Java program, was downloaded from www.theebest.com/games/3ddriver/3ddriver.shtml. No changes were made to it to accommodate the model, but we did modify it to record behavior (acs.ist.psu.edu/driver-model). Fig. 2 shows an example display.

This task provides a first-person view as task perspective and the environment changes dynamically in response to the actions of the user and task environment, as is the case for many human–robot systems. It is, we acknowledge, a simplified version of a robot interface, but it allows us to illustrate our approach. The simulation we are using is comparable to many robot interfaces in that it relies heavily on perceptual–motor skills and involves decision making under time pressure and interaction with a dynamically changing environment.

D. The Driving User Model in ACT-R and Segman

Our system, Driver User Model in ACT-R and SegMan (DUMAS, pronounced "doo 'maa"), drives a car in the 3D Driver simulation. DUMAS incorporates ACT-R's assumptions about how cognition works and interacts with perception and

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

motor movement. Using SegMan incorporates that system's capabilities to see and interact with displays without instrumenting the display to support the model. DUMAS, although primitive, provides an initial computational information-processing user model of HRI that interacts with an unmodified interface. The model, like human users, starts the program by clicking the mouse on its window, accelerates by pushing the "A" key, brakes by pushing the "Z" key, and steers by using the left and right arrow keys.

The perceptual processing strategies in the model are based on observations from the literature on human driving, as is common for other driving models [12]. Land and Horwood's [33] study of driving behavior describes a "double model" of steering, in which a region of the visual field relatively far away from the driver (about 4° below the horizon) provides information about road curvature, while a closer region (7° below the horizon) provides position-in-lane information. Attention to the visual field at an intermediate distance, 5.5° below the horizon, provides a balance of this information, resulting in the best performance.

The model computes position-in-lane information by detecting the edges of the road and the stripes down the center of the road. The stripes are combined into a smoothed curve to provide the left boundary of the lane, while the right edge of the road directly gives the right lane boundary. The model computes the midpoint between these two curves at 5.5° below the horizon. This point, offset by a small amount to account for the left-hand driving position of the car, is treated as the goal. If the center of the visual field (the default focal point) is to the right of this point, the model must steer to the right, otherwise, to the left.

Perceptual processing in the model has limitations. For example, it is not entirely robust—determining the center of the lane can break down if the road curves too fast. It also only returns some of the information that it has extracted. For example, it can determine road curvature from more distant points, as is done in models of human driving [34]; however, this has not led to improved performance in this simulation environment.

A useful aspect of ACT-R models is that they can generate a protocol of behavioral output, illustrating how separate parts of a complex behavior like driving unfold over time. Table I illustrates this with an example run of the model, starting with the "Go" production and ending with a crash. We chose an unusually short run for illustrative purposes. As you can see, the protocol indicates what behavior (e.g., steering, cruising) is taking place at specific points in time. These predictions can be compared to the sequential behavior of human subjects as a further validation of the model, or to gain further insight into a complex behavior such as navigation.

TABLE I
TRACE OF THE DUMAS MODEL USING THE 3-D DRIVING INTERFACE

```
Time  0.000: Go Selected
**********************GO!!**********************
Time  0.050: Go Fired
Time  0.050: Perceive-Environment Selected
Time  0.100: Perceive-Environment Fired
Time  0.100: Decide-Action Selected
Time  0.150: Decide-Action Fired
Time  0.150: Steer-Right Selected
GOING TO THE RIGHT >>>>>>>>>>>>>>>>>>>>>>>>
Time  0.200: Steer-Right Fired
Time  0.200: Perceive-Environment Selected
Time  0.250: Perceive-Environment Fired
Time  0.250: Decide-Action Selected
Time  0.300: Decide-Action Fired
Time  0.300: Steer-Right Selected
GOING TO THE RIGHT >>>>>>>>>>>>>>>>>>>>>>>>
Time  0.350: Steer-Right Fired
Time  0.350: Perceive-Environment Selected
Time  0.400: Perceive-Environment Fired
Time  0.400: Decide-Action Selected
Time  0.450: Decide-Action Fired
Time  0.450: Cruising Selected
   CRUISING
Time  0.500: Cruising Fired
Time  0.500: Perceive-Environment Selected
Time  0.550: Perceive-Environment Fired
Time  0.550: Decide-Action Selected
Time  0.600: Decide-Action Fired
Time  0.600: Steer-Left Selected
<<<<<<<<<<<<<<<<<<<<<<<<GOING TO THE LEFT
Time  0.650: Steer-Left Fired
Time  0.650: Perceive-Environment Selected
Time  0.700: Perceive-Environment Fired
Time  0.700: Decide-Action Selected
Time  0.750: Decide-Action Fired
Time  0.750: Cruising Selected
   CRUISING
Time  0.800: Cruising Fired
Time  0.800: Perceive-Environment Selected
Time  0.850: Perceive-Environment Fired
Time  0.850: Crashing Selected  [model saw crash effect on screen]
***********************CRASH****************
<<<<<<<Writing data to data.txt>>>>>>
Time  0.900: Crashing Fired
```

performance, and we used multiple runs at each speed because the ACT-R architecture includes stochastic elements.

III. RESULTS

To test our model and to understand its performance and correspondence to human behavior, we compared the model's performance with human performance data gathered on the same task and interface. We ran the model with the driving simulator set at three speeds to see the effect of speed on

A. The Model's Predictions

The model performed the driving task at three different simulated speeds: 25, 45, and 65 mi/h. There were ten runs at each speed. Lane deviation (the distance from the optimal driving position, measured in degrees) and total driving time before crashing (as noted by the simulation) were recorded. Lane deviation is commonly used in driving studies to measure the influence of a variety of factors such as multitasking [12] and drug use [35]. Table II shows the results for the model.

B. Human Data

Twenty-four undergraduate and graduate students at Penn State (11 women, 13 men, mean age 22.5) volunteered for the

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

TABLE II
MEAN LANE DEVIATION AND TIME TO CRASH AS A FUNCTION OF SPEED FOR THE DUMAS MODEL

	Lane Deviation (degree)		Time to Crash (second)	
Speed	Mean	SD	Mean	SD
25 mi/h	4.95	1.36	11.56	5.25
45 mi/h	6.72	3.93	9.44	6.67
65 mi/h	9.37	2.73	5.85	3.88

TABLE III
MEAN LANE DEVIATION AND TIME TO CRASH AS A FUNCTION OF SPEED FOR HUMAN SUBJECTS

	Lane Deviation (degree)		Time to Crash (second)	
Speed	Mean	SD	Mean	SD
25 mi/h	3.97	1.63	14.39	7.58
45 mi/h	5.64	2.33	11.22	5.37
65 mi/h	8.73	3.78	8.03	5.04

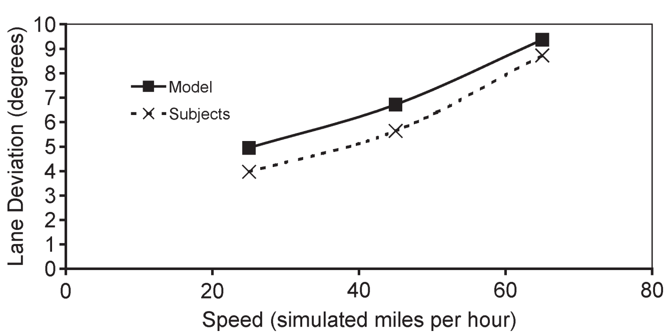


Fig. 3. Average lane deviation for the model and subjects as a function of speed.

study. Subjects drove at three different simulated speeds (25, 45, and 65 mi/h) until they crashed. The speed order varied randomly for each subject to eliminate ordering effects.

A version of the DUMAS model (called mini-DUMAS) ran concurrently in the background, undetectable to the subject except that it controlled the speed. As the subject drove, mini-DUMAS computed the lane deviation and time to crash but did not act on these perceptions. This is an example of using a model to compute user performance in real time. The subjects' results are shown in Table III.

C. Comparison of Human Performance to the Model's Predictions

As speed increased, lane deviation increased and time to crash decreased for both the model and human subjects. Comparisons are shown in Figs. 3 and 4. The human data correlates with the model's predictions $r^2 = 0.998$ on lane deviation and $r^2 = 0.989$ on time to crash. These comparisons show that DUMAS's performance is similar to our subjects. Humans slightly outperformed the model; however the model simulates human performance well enough to draw conclusions from.

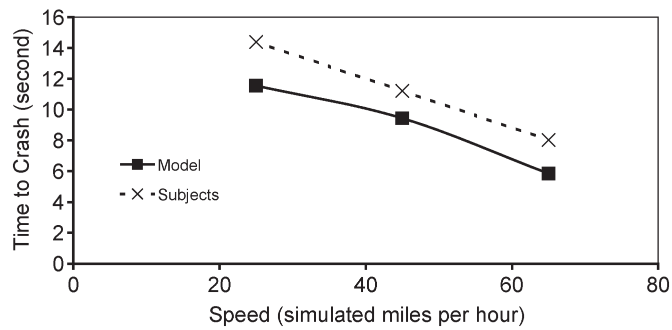


Fig. 4. Average time to crash for the model and subjects as a function of speed.

IV. DISCUSSION

DUMAS, although preliminary, illustrates many of the aspects of a theory of HRI. The use of a cognitive architecture includes many regularities of human behavior. Validating the application of DUMAS to a simple analog of many HRI tasks with human data allows identifying several issues that are relevant to robot interface design and use. While more advanced models will no doubt find further problems, the issues presented here will remain—these issues will not become easier when performing more complex tasks with more complex interfaces.

1) Visual Orientation: Visual input is undoubtedly the most important source of information in driving or teleoperation. Nevertheless, the human visual system seems badly equipped for a task like driving. We only see sharply in a small center of the visual field (the fovea); acuity drops significantly towards the periphery. As a result, eye movements (saccades), are needed to construct an integrated field of vision for larger scenes. To accomplish this, a driver needs a theory of where to look and what visual features are important.

In its current form, DUMAS's problems staying on the road arise from how it handles visual input. At the moment, the model takes snapshots of the whole visual scene to determine its actions. It detects changes by recording the locations of specific points in the visual field and then measuring the distance they move from one snapshot to the next. It turns out that this is not a good way to handle visual flow. Suppose that at time t the model analyzes the road, records the data for estimating visual flow, and determines that steering one direction or another is appropriate. At time $t + 1$, a steering command is issued, and the simulated car moves in that direction. At time $t + 1$ or later, the road is again analyzed so that flow can be computed, but at this point the action of the model resulted in changes in the visual field, independent of changes that would have occurred otherwise. This contribution needs to be accounted for, or the car would end up braking every time it steers. Thus, because DUMAS has difficulties with perception and is sensitive to the update rate and complexity of the display, it predicts that perception of this interface is something that makes this task difficult for humans, particularly until they learn where to look.

The domain of visual orientation is probably the place where collaboration between human and robot will be most intense. Robots continue to be poor at high-level perceptual functions, such as object recognition and situation assessment [2], [36], which means the human operator will still play an important

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

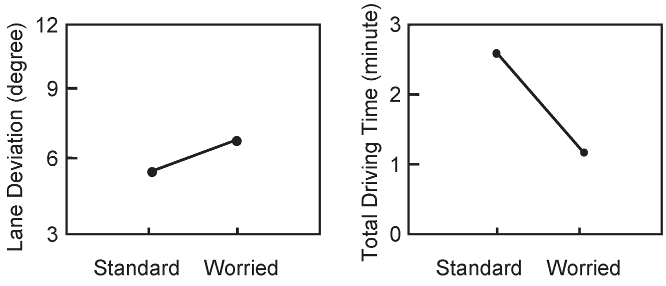


Fig. 5. Lane deviation and total driving time of the driving model as a function of standard or worried condition.

role. However, USAR applications have illustrated that it is often not an easy task for even a human operator to infer complex features in certain environments. For instance, when maneuvering through a small and dark shaft, it is difficult for both humans and machines to discern features. As a result, it becomes hard to perform tasks like identifying victims [37]. This suggests two areas worth exploring: a) how the human operator can remove ambiguities that arise from the limited visual capabilities of robots by providing information that enables the machine system to adapt to the situation at hand, for example, what the targets are; and b) how machine algorithms can make the displays richer to help human interpretation, for example, by providing information that the machine vision system can compute, such as distance to an object, some measure of object size, and frame of reference with respect to the robot.

2) Speed Control and Steering: The model has to decide which speed would be appropriate based on information coming from the visual system. This means the user model has to have a theory that determines optimal speed in a given situation. Thus, the process of controlling speed will be determined to a great extent by the constraints of other parts of the system.

An interesting and unexpected parallel that appeared between our simple driving task demonstration and USAR applications is that of course corrections. In the driving task, situations would occur in which an early version of the model would overreact to a change in the environment, usually a curve, and would brake to a complete halt. Subsequently, the model would go into a recurring sequence of overcorrections, turning too far in each direction as it attempted to straighten, and the run either had to be terminated or the model had to be helped by the analyst. The same has been seen to occur in USAR applications when the operator turns too far towards a desired location and then has to correct, leading to a series of oscillations, performing the same type of repetitive cycle of overcorrections with a robot [37].

A constraint that typically arises in teleoperated navigation is that of communications time lag, which can occur as a result of navigating quickly. In USAR, the time between operating the controls and the robot actually reacting can create an additional time lag. The impact of time lags could be added to our system, and would lead to poorer performance. This effect of time lag on control tasks has been found in HRI systems [37, p. 12] and is seen as a general problem in human factors [38]. For example, Fig. 5 shows how performance decreases when the model worries, including a validated worry component from another project [39]. Time lags would have a similar effect.

3) Multitasking: Most process models of driving use very efficient and continuous processes to perceive the environment and control the car (e.g., [40]). In contrast, our implementation uses a discrete updating mechanism, reflecting the production rule nature of the ACT-R model. This has important consequences. First, our model does not produce optimal behavior but rather tries to simulate human behavior. Specifically, its performance is determined by the speed and accuracy with which it can react and adapt to the environment. Second, our model allows assessing the influence of a secondary task, in other words, the effects of multitasking (this measure is also of interest to those studying driving [12] and working robot teams [5], [6]). If the model has to divide its attention between two tasks, it is likely that each of those tasks will suffer, and this is consistent with previous work attempting to measure the effect of neglecting a robot because of a secondary task [5]. Specifically, the time between updates increases, leading to poorer performance. We have only preliminary results, but Fig. 5 shows how a secondary task (of simply having a simple rule fire to represent active worry) appears to disrupt performance as badly as a 20-mi/h speed increase in the range we have used. Extensions of this model could support an analysis of how many robots can be driven at the same time, or, more specifically, predict how performance quantitatively degrades with increases in workload and stress.

Some interaction aspects have not yet been directly addressed in this work, but can be in the future. These are:

4) Navigation: Our driving simulation has a direct interface—the operator directs the car using the keyboard looking at a simulated world. This perspective is often referred to as inside-out driving or piloting, because the operator feels as if he is inside the robot or vehicle and looking out. (This can be contrasted with a plan-view or God's-eye-view display.) This interface is a common method for vehicle or robot teleoperation [1]. A problem that arises with direct interfaces is that they provide poor contextual cues, which lead to less situation awareness. Work here could be extended by incorporating a model of spatial reasoning (e.g., [41]), and investigating how map building by the user is related to aspects of the user interface and characteristics of the user (expert or novice, high or low working-memory capacity, and so on).

5) The Influence of the User Interface on Performance: The present approach is now suited to explore how changes in the interface will affect performance. The model can provide quantitative predictions of the effects of interface changes. As such, the user model starts to indicate at what point robot autonomy could effectively compensate for human-operator constraints. It also becomes possible to study how individual differences (e.g., attention, visual, and motor capabilities) of the human operator can influence performance on different interfaces, that is, how to support different kinds and levels of operators. Work like this on optimizing cell phone menu design has found that 30% improvements in keying time are possible [18].

6) The Level of Expertise of the User: By varying the task knowledge in the user model, one could vary the skill of the operator and see how this affects performance. Knowledge can also affect the design of the interface as well, as one would like experts to have great flexibility and control, while novices

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

should be guarded from making large and costly mistakes. The application of the current user model can predict aspects of the relationship between user interface designs and performance in a direct and quantitative way.

V. Conclusion

Based on the results of this simple model, we can see several general lessons for the area of HRI design. We first note the limitations of this model and then examine the implications of what we have learned. The lessons were prepared with USAR in mind.

A. Limitations of DUMAS

These results are limited in several ways. The limitations arise because the current version of the model does not interact with a very complex interface and does not do very complex tasks. HRI has grown to include many types of interfaces and many types of tasks. The results reported here are thus only a subset of what makes robot interfaces difficult to use. The difficulties this model has, such as difficulties seeing, will exist for more complex interfaces, but that there will be further difficulties discovered as this approach goes forward. For example, this model only uses one sensing mode (vision) and does not examine situations in which users can see robots. The issues noted in Section IV will still apply when using additional sensing modes, particularly if these additional sensing modes are displayed to the operator through a visual interface, but our results of problems and pitfalls will be a subset of the larger problems.

This model is also of a single user. It would have to be extended to explore the problems that more complex interaction and mixed teams would create, such as a robot as a co-worker or of multiple users working with the same robot. Also, a role in a team other than operator may typically not include visual feedback from the robot's viewpoint but may involve observing the robot directly (e.g., [37]).

The model has less to say about more complex tasks, such as those relying on multiple sensors and those with mixed-initiative control, which we know are important aspects of HRI. For example, to examine the use of multiple sensors, we would have to include knowledge about how to choose between the appropriate sensor modes to use for a given task, or how to use the information from multiple modes. Nonetheless, we can note some of the important problems that models, and thus operators, have to address that we can see at this point.

Clearly, an obvious place for improvement and a next step is to have the model interact with a more realistic interface from a robot or a robot simulation. Examples of such simulations include Cornell's RoboFlag [42], the standard RoboCup [43], and simulation of the NIST USAR testbed [44], [45].

B. Robot Interfaces are Hard to Use Because They Contain Many Difficult Tasks

The model explored here suggested that it is not an artifact or coincidence that current robot interfaces are difficult to use. Several of the tasks are quite difficult for this model and will be for users generally. The difficulties range from the perceptual issues that must be addressed, to the relatively high-level cognition and problem solving in new situations that must be supported, and the sensitivity of the task to response time, as well as the several kinds and large amount of knowledge that is required. These difficulties of the model are not artifacts, rather they are consistent with what is known in psychology and HCI.

The model of vision that interprets the bitmap makes direct suggestions that a reason the task is difficult is because the vision-recognition problem of the real-world bitmaps is difficult. This result suggests that changes that help operators see the world better, such as better display hardware and augmented reality, would make robot interfaces easier to use.

Robot interfaces are typically complex interfaces. The model can predict how the usability of robot interfaces are restricted by many factors, including the quality of the display, communication issues such as bandwidth and response lag, the processing speed of the user relative to the robot they are manipulating, the knowledge and processing in the robot and user, and the user's eye–hand coordination. There are thus many ways to improve interaction, but good interaction will require getting most or all of these factors correct.

Improving an interface that relies on this many factors is difficult without tools to help keep track of these factors and their relationships. Keeping track of these factors in complex environments, such as operating multiple robots, understanding how their control structures will be robust to perturbations, understanding tradeoffs in designs, and predicting how many robots an operator can easily manipulate, will be particularly difficult.

C. Changes to Robot Interfaces Can be Important

The results of the model suggest changes to robot interfaces can lead to large changes in the user's performance with the interface. Applications of the model suggest that changes to an interface can reduce the time to perform a task and can reduce errors. The model also shows that even changes that only lead to better perception of the interface and decreased worry (as implemented as a secondary task) by the user would be helpful.

D. Models Can Use HRI Interfaces Directly

Having the model use SegMan to interact using the interface's bitmap means that models can interact with a much wider range of interfaces, including other robot interfaces. Other interfaces that run under the Windows operating system can be examined directly. For example, this approach has been used by models to interact with Minesweeper, and multiple telephone interfaces [16], [18], which are shown in Fig. 6. Models using SegMan can also now interact in simple ways with the Evolution Robotics' interface shown in Fig. 7, which is similar but much simpler than the teleoperation of real vehicles such as Dragon Runner [46] or USAR robots [1]–[3].

Creating the model within a cognitive architecture provides several further advantages when creating a large model. It provides a source of additional model components, as there

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

Fig. 6. Three example phone interfaces that SegMan+ACT-R/PM models interact with [16].

Fig. 7. The Evolution Robotic's interface that SegMan+ACT-R/PM interacts with.

are many people creating models of human behavior using the ACT-R architecture (act.psy.cmu.edu/papers.html provides a list). There are several models we can already point to as being useful for studying interfaces in general and HRI in particular that could be included. These include Freed's (explained in [16]) and St. Amant's [18] models of telephone dialing (for button pushing and visual search in an interface), and Salvucci's [12] model of telephone dialing while driving (for multitasking). Young's [47] and St. Amant's [2] models of interface exploration may be extendable to include search in the environment, a common task for USAR robots. This is theoretically pleasing because it offers an additional audience of the models, other users, and scientists to test the models in formal and informal ways through reuse. There are enough models that for some projects we have been able to build upon and combine existing models rather than create the model entirely from scratch.

Working within a common cognitive-modeling architecture provides resources from the architecture for people interested in understanding the model as well. For ACT-R, this includes an online tutorial, programming interfaces, a mailing list to get help with technical problems, and a manual.

E. User Models Can Provide Insights to Robot Interface Design

The model presented here shows that models of users can already provide insights and summaries useful for creating better robot interfaces. The various levels of the model provide suggestions for improving the interface we studied, and by analogy provides suggestions for many USAR robot interfaces.

This approach of creating a model user results in a description of human behavior that may be useful for teaching to interface designers. To a certain extent, this knowledge of users and of how to support them with interfaces can be passed to designers through books on HCI (and soon, HRI). Model users like the one presented here may provide better summaries of users by providing a system-level summary of users, which may be more interesting, manipulatable, and informative to HRI designers because it is more concrete. The model's design can be examined (and explained in teaching materials) by students and designers, forming the basis for their mental model of users.

The model's parameters can be varied to represent individual difference in the robot operator, such as working-memory capacity and knowledge. The impact of these differences on performance can be examined. These preliminary analyses predict how individual differences in users will be important. Differences in interfaces, broadly defined, can also start to be examined without running subjects. For example, these results suggest that the speed of the robot can influence its drivability, with increased speed not always helpful. Finally, reuse contributed to creating the model as we were able to take insights and sometimes code from other models [12], [39].

F. Recommendations for Future User Models

In addition to the limitations of the model noted, there are several recommendations for further work that are now apparent.

1) Reusable Simulated Eyes and Hands Should Continue to be Developed: The simulated eyes and hands used in this work open up the application of user models to test robot interfaces and computer interfaces generally. These simulated eyes and hands also provide a way for user models to interact with other pieces of software, significantly increasing the range of tasks that are feasible for user models to perform. Similar reuse can be obtained using this approach to apply agents to interfaces; indeed, the simulated eyes and hands were originally built to support agents' abilities to interact with interfaces [28].

2) Libraries of Models: One of the ways to encourage this work and to make it more feasible is to create libraries of behaviors for reuse. That is, it would be useful to create a series of models that perform typical tasks done with robot interfaces, and to create them in such a way that they are modular and

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

reusable. These libraries should be created in the same spirit that graphics and statistical libraries are created.

As these user models become easier to create, they will be able to more routinely provide feedback directly to interface designers. In the meantime, example models like this can summarize behavior with robot interfaces, noting what makes their interfaces difficult to use so that designers can study these problems and avoid them.

Acknowledgment

Comments from participants at ACT-R and DARPA workshops, P. Benjamin, J. Jansen, W. Stevenson, three anonymous reviewers, and the Associate Editor have improved this presentation. In particular, discussions with J. Scholtz, R. Murphy, and L. Parker have contributed to the authors' understanding. A. Reifers modified the 3D Driver to record behavior.

References

[1] T. W. Fong and C. Thorpe, "Vehicle teleoperation interfaces," *Auton. Robots*, vol. 11, no. 1, pp. 9–11, 2001.

[2] R. R. Murphy, J. Blitch, and J. Casper, "AAAI/RoboCup-2001 urban search and rescue events: Reality and competition," *AI Mag.*, vol. 23, no. 1, pp. 37–42, 2002.

[3] J. Casper and R. Murphy, "Human–robot interactions during the robot-assisted urban search and rescue response at the World Trade Center," *IEEE Trans. Syst., Man, Cybern. B*, vol. 33, no. 3, pp. 367–385, Jun. 2003.

[4] J. L. Drury, D. Hestand, H. A. Yanco, and J. Scholtz, "Design guidelines for improved human–robot interaction," in *CHI Extended Abstracts 2004*. New York: ACM, 2004, p. 1540.

[5] J. W. Cradall, C. W. Nielsen, and M. A. Goodrich, "Towards predicting robot team performance," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Washington, DC, 2003, pp. 906–911.

[6] D. R. Olsen, Jr. and M. A. Goodrich, "Metrics for evaluating human–robot interactions," presented at the Proc. Performance Metrics Intelligent Systems (PERMIS), Gaithersburg, MD, 2003. NIST Special Publication 1014. [Online]. Available: www.isd.mel.nist.gov/research_areas/research_engineering/Performance_Metrics/PerMIS_2003/Proceedings/

[7] J. L. Marble, D. J. Bruemmer, D. A. Few, and D. D. Dudenhoeffer, "Evaluation of supervisory vs. peer–peer interaction with human–robot teams," in *Hawaii Int. Conf. System Sciences (HICSS)*, Big Island, HI, 2004, p. 50130b.

[8] L. S. Lopes and J. H. Connell, "Sentience in robots: Applications and challenges," *IEEE Intell. Syst.*, vol. 16, no. 5, pp. 66–69, Sep./Oct. 2001.

[9] M. T. Allen and M. D. Crowell, "Patterns of autonomic response during laboratory stressors," *Psychophysiology*, vol. 26, no. 5, pp. 603–614, 1989.

[10] D. E. Kieras, S. D. Wood, and D. E. Meyer, "Predictive engineering models based on the EPIC architecture for a multimodal high-performance human–computer interaction task," *ACM Trans. Comput.-Hum. Interact.*, vol. 4, no. 3, pp. 230–275, 1997.

[11] F. E. Ritter, "Choosing and getting started with a cognitive architecture to test and use human–machine interfaces," *MMI Interakt. J.*, vol. 7, pp. 17–37. [Online]. Available: useworld.net/mmiij/musimms

[12] D. Salvucci, "Predicting the effects of in-car interface use on driver performance: An integrated model approach," *Int. J. Human-Comput. Stud.*, vol. 55, no. 1, pp. 85–107, 2001.

[13] F. E. Ritter and J. H. Larkin, "Using process models to summarize sequences of human actions," *Hum.-Comput. Interact.*, vol. 9, no. 3, pp. 345–383, 1994.

[14] Z. Duric *et al.*, "Integrating perceptual and cognitive modeling for adaptive and intelligent human–computer interaction," *Proc. IEEE*, vol. 90, no. 7, pp. 1272–1289, Jul. 2002.

[15] M. D. Byrne, "ACT-R/PM and menu selection: Applying a cognitive architecture to HCI," *Int. J. Human-Comput. Stud.*, vol. 55, no. 1, pp. 41–84, 2001.

[16] R. St. Amant, A. R. Freed, and F. E. Ritter, "Specifying ACT-R models of user interaction with a GOMS language," *Cogn. Syst. Res.*, vol. 6, no. 1, pp. 71–88, 2005.

[17] J. R. Anderson and C. Lebiere, *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum, 1998.

[18] R. St. Amant, T. E. Horton, and F. E. Ritter, "Model-based evaluation of cell phone menu interaction," in *Proc. CHI Conf. Human Factors Computer Systems*, Vienna, Austria. New York: ACM, 2004, pp. 343–350.

[19] K. Shah, S. Rajyaguru, R. St. Amant, and F. E. Ritter, "Connecting a cognitive model to dynamic gaming environments: Architectural and image processing issues," in *Proc. 5th Int. Conf. Cognitive Modeling*. Bamberg, Germany: Universitäts-Verlag Bamberg, 2003, pp. 189–194.

[20] R. St. Amant and M. O. Riedl, "A perception/action substrate for cognitive modeling in HCI," *Int. J. Human-Comput. Stud.*, vol. 55, no. 1, pp. 15–39, 2001.

[21] E. Norling and F. E. Ritter, "Embodying the JACK agent architecture," in *AI Advances Artificial Intelligence Proc. 14th Australian Joint Conf. Artificial Intelligence*, Adelaide, Australia. Berlin, Germany: Springer-Verlag, 2001, pp. 368–377.

[22] F. E. Ritter, G. D. Baxter, G. Jones, and R. M. Young, "Supporting cognitive models as users," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, no. 2, pp. 141–173, 2000.

[23] J. R. Anderson, M. Matessa, and C. Lebiere, "ACT-R: A theory of higher level cognition and its relation to visual attention," *Hum.-Comput. Interact.*, vol. 12, no. 4, pp. 439–460, 1997.

[24] M. C. Lovett, L. M. Reder, and C. Lebiere, "Modeling working memory in a unified architecture," in *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, A. Miyake and P. Shah, Eds. New York: Cambridge Univ. Press, 1999, pp. 135–182.

[25] J. R. Anderson, J. M. Fincham, and S. Douglass, "The role of examples and rules in the acquisition of a cognitive skill," *J. Exper. Psychol., Learn., Mem., Cogn.*, vol. 23, no. 4, pp. 932–945, 1997.

[26] M. J. Schoelles and W. D. Gray, "Argus Prime: Modeling emergent microstrategies in a complex simulated task environment," in *Proc. 3rd Int. Conf. Cognitive Modelling*, Groningen, The Netherlands. Veenendaal, The Netherlands: Universal Press, 2000, pp. 260–270.

[27] P. M. Fitts, "The information capacity of the human motor system in controlling amplitude of movement," *J. Exp. Psychol.*, vol. 47, no. 6, pp. 381–391, 1954.

[28] R. St. Amant, "Interface agents as surrogate users," *Intelligence*, vol. 11, no. 2, pp. 28–38, Summer 2000.

[29] J. Aasman, *Modelling Driver Behaviour in Soar*. Leidschendam, The Netherlands: KPN Res., 1995.

[30] M. Antoniotti and A. Göllü, "SHIFT and SmartAHS: A language for hybrid systems engineering, modeling, and simulation," in *Proc. USENIX Conf. Domain Specific Languages*, Santa Barbara, CA, 1997, pp. 171–182.

[31] T. Bellet and H. Tattegrain-Veste, "A framework for representing driving knowledge," *Int. J. Cogn. Ergon.*, vol. 3, no. 1, pp. 3–49, 1999.

[32] J. Casper and R. Murphy, "Workflow study on human–robot interaction in USAR," in *Int. Conf. Robotics and Automation (ICRA)*, Washington, DC, 2002, pp. 1997–2003.

[33] M. F. Land and J. Horwood, "Which parts of the road guide steering?", *Nature*, vol. 377, no. 6547, pp. 339–340, 1995.

[34] M. F. Land and D. N. Lee, "Where we look when we steer," *Nature*, vol. 369, no. 6483, pp. 742–744, 1994.

[35] H. W. J. Robbe, "Marijuana use and driving," *J. Int. Hemp Association*, vol. 1, no. 1, pp. 44–48, 1994.

[36] P. Milgram, S. Zhai, D. Drascic, and J. J. Grodski, "Applications of augmented reality for human–robot communication," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*. Yokohama, Japan: IEEE, 1993, pp. 1467–1472.

[37] R. Murphy, J. Casper, M. Micire, and J. Hyams, "Mixed-Initiative Control of Multiple Heterogeneous Robots for USAR," Center for Robot-Assisted Search and Rescue, Univ. South Florida, Tampa, FL, Tech. Rep. CRASAR-TR2000-11, 2000.

[38] C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2000, ch. 10.

[39] F. E. Ritter, A. Reifers, L. C. Klein, K. Quigley, and M. Schoelles, "Using cognitive modeling to study behavior moderators: Pre-task appraisal and anxiety," in *Proc. Human Factors and Ergonom. Society*. Santa Monica, CA: Human Factors and Ergonomics Soc., 2004, pp. 2121–2125.

[40] B. Cheng and T. Fujioka, "A hierarchical driver model," in *Proc. IEEE Conf. Intelligent Transportation Systems (ITSC)*, Boston, MA [CD-ROM]. Piscataway, NJ: IEEE, 1997, 960–965.

[41] T. R. Johnson, H. Wang, and J. Zhang, "An ACT-R model of human object-location memory," in *Proc. 25th Annu. Meeting Cognitive Science Society*, Austin, TX. Mahwah, NJ: Lawrence Erlbaum, 2003, p. 1361.

[42] R. D'Andrea and M. Babish, "The RoboFlag testbed," in *Proc. American Controls Conf.*, Denver, CO, 2003, pp. 656–660.

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.

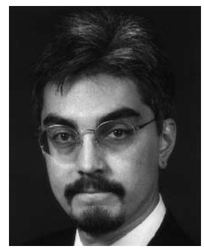
[43] P. Lima, T. Balch, M. Fujita, R. Rojas, M. Veloso, and H. Yanco, "RoboCup 2001," *IEEE Robot. Autom. Mag.*, vol. 9, no. 2, pp. 20–30, 2002.
[44] P. Prasithsangaree, J. Manojlovich, J. Chen, and M. Lewis, "UTSAF: A simulation bridge between OneSAF and the unreal engine," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics,* Washington, DC. Piscataway, NJ: IEEE Press, 2003, vol. 2, pp. 1333–1338.
[45] M. Lewis, K. Sycara, and I. Nourbakhsh, "Developing a testbed for studying human-robot interaction in urban search and rescue," in *Proc. 10th Int. Conf. Human Computer Interaction (HCII)*, Heraklion, Greece. Mahwah, NJ: Lawrence Erlbaum, 2003, pp. 270–274.
[46] H. Schempf, "Ultra-rugged soldier–robot for urban conflict missions," presented at the Unmanned Systems 2003 Conf.—Association for Unmanned Vehicle Systems International (AUVSI) 30th Annu. Symp. and Exhibition, Baltimore, MD. CD Proceedings.
[47] R. M. Young, "Rational analysis of exploratory choice," in *Rational Models of Cognition*, M. Oaksford and N. Chater, Eds. Oxford, U.K.: Oxford Univ. Press, 1998, pp. 469–500.

Frank E. Ritter (S'82–M'84) received the B.S.E.E. degree from the University of Illinois at Urbana-Champaign, and the M.S. degree in psychology and the Ph.D. degree in artificial intelligence and psychology from Carnegie-Mellon University, Pittsburgh, PA.

He is an Associate Professor at the School of Information Sciences and Technology at the Pennsylvania State University, and is affiliated with the Psychology and the Computer Science and Engineering Departments. He is on the editorial board of *Human Factors, AISB Quarterly*, and *AISB Journal*. One of his recent projects, a report on applying cognitive models in synthetic environments, was recently published as a State of the Art Report on *Techniques for modeling human performance in synthetic environments: A supplementary review* (Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center. 2003, iac.dtic.mil/hsiac/S-docs/SOAR-Jun03.pdf).

Dirk Van Rooy received the B.S., M.S., and Ph.D. degrees from the Free University of Brussels, Belgium.

He is currently a Lecturer at Birmingham University, Birmingham, Staffordshire, U.K.



Robert St. Amant received the B.S. degree in electrical engineering and computer science from the Johns Hopkins University, Baltimore, MD, in 1985 and the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 1996.

He is an Associate Professor in the Computer Science Department at North Carolina State University, Raleigh. He codirects the IMG Lab, which performs research in the areas of intelligent user interfaces, multimedia, and graphics.

Prof. St. Amant won, along with Frank E. Ritter, the "Best Applied Paper" award at the 2004 International Conference on Cognitive Modeling.

Kate Simpson received the B.S. degree in psychology and in computer science and engineering in 2001 from The Pennsylvania State University, University Park.

She is a Research Assistant at the School of Information Sciences and Technology at the Pennsylvania State University.

Authorized licensed use limited to: Australian National University. Downloaded on January 3, 2010 at 20:04 from IEEE Xplore. Restrictions apply.