Long Road Ahead: Lessons Learned from the (Soon to Be) Longest Running Cognitive Model

Siyu Wu (sfw5621@psu.edu)[1], Amir Bagherzadeh (abb6024@psu.edu)[2],
Frank E. Ritter (frank.ritter@psu.edu)[1], and Farnaz Tehranchi (farnaz.tehranchi@psu.edu)[3]

1 College of Information Sciences and Technology
2 Department of Industrial and Manufacturing Engineering
3 School of Engineering Design and Innovation
Penn State, University Park, PA, 16802, USA

Abstract

We present a cognitive model that plays a video game of driving a bus for a long time. The model was built using the ACT-R cognitive architecture and an extension to support perceptual-motor knowledge of how to interact with the environment (VisiTor and ACT-R. Our extension includes bitmap-level eyes and robot hands. We ran the model for a long time, over 4 hours on the way from Tucson to Las Vegas. We employed a design approach based on the ADDIE model to create different knowledge representations and actions; the model's predictions can be matched to some aspects of human behavior on the fine details regarding the number of course corrections and average speed and learning rate. However, it does not exhibit the same level of fatigue as human behavior. This contrasts with the way humans typically perform such long tasks. This model shows that (a) perception opens up new interfaces and provides a very accessible testbed for examining further aspects of behavior and (b) adding components of human behavior that remain missing from ACT-R can now be included.

Keywords: Cognition Computational Modeling; ACT-R; Driver Model

Introduction

Cognitive architectures can be used to develop cognitive models of various psychological phenomena and tasks (Newell, 1990). In addition, cognitive architectures afford procedures and structures that align with human behavior, such as reaction times, error rates, and fMRI results (Anderson, 2007; Laird, 2019).

ACT-R is one kind of cognitive architecture realized as software, through which we can construct models that can store, retrieve, and process knowledge, as well as explain and predict performance (Anderson, 1996; Bothell, 2017). Modelers and researchers have used ACT-R to create a variety of models, from models that only contain cognition-level activities (e.g., Tower of Hanoi) to models that contain comprehensive perception and motor behaviors (e.g., Fleetwood & Byrne, 2002; Tehranchi & Ritter, 2018).

In particular, ways to implement perceptual and motor behaviors can be classified into several categories based on how directly they interact with a task (Ritter et al., 2000). Perhaps the most commonly seen approach is models that interact with modified interfaces (e.g., Anderson & Douglass, 2001; Byrne et al., 2010).

Another important approach is to have models interact with unmodified tasks that users can see, using simulated eyes and hands (e.g., Bagherzade & Tehranchi, 2022; Ritter et al., 2006). Previous studies used ACT-R to build cognitive models and compare the model behavior with human behavior, finding that ACT-R's model behavior at the cognition level is more consistent with human behavior than at the perceptual motor level (e.g., Ritter et al., 2006; Schwartz, Tehranchi, & Ritter, 2020). However, extending ACT-R to model further aspects of vision and motor behavior on uninstrumented interfaces can be an important future direction (Ritter et al., 2019; Laird, 2019; Pew & Mavor, 2007).

For example, Schwartz et al., (2020) built a model using ACT-R in conjunction with an extended vision and motor management tool (JSegMan) to play Penn and Teller's *Desert Bus* video game (e.g., https://desertbus.org/; Parkin, 2013). However, the model's behavior shows discrepancies with human behavior, and one of them is that the model can only run the bus for less than 20 minutes, which is a much shorter time than a human can. Table 1 lists some of the discrepancies between their model and human behavior.

Table 1. Limitations of the Schwartz et al. (2020) model.

1.	Did not start the simulation
2.	Did not drive for more than 20 min.
3.	Did not make the first turn around
4.	Did not make the second turn around (which might be different, after 16 hours of driving)

Our research expands upon the work of Schwartz et al. (2020) by using the previous ACT-R model and revising its perceptual motor components to enable real-time control of a driving task. Our study presents a model that possesses two relatively novel capabilities for cognitive models: it can perform long-term tasks lasting up to 4 hours, and it can do so while interacting with an interface that was not specifically designed for models. Ultimately, the models will be able to play the video game *Desert Bus* for a much longer period of time, essentially, indefinitely.

Using the ADDIE (Morrison et al., 2010) framework as our design approach, we improved the knowledge representation and actions of the model. We also added a new function for the extended hand. Although the model successfully completed the task longer, its behavior on this task revealed limitations in the ACT-R model, which we identify and attempt to address.

Components and Theoretical Foundations

We now explain our architecture and the perceptual interface to interact with the interface. We then describe the simulation that the model interacts with.

The Architecture of Cognition

ACT-R is a cognitive architecture and a theory of simulating and understanding human cognition (Anderson, 2007; Ritter, Tehranchi, & Oury, 2019). Its theory is embodied in the ACT-R software, through which we can construct models that can store, retrieve, and process knowledge, as well as explain and predict performance (Bothell, 2017).

There are currently two kinds of knowledge representations in ACT-R, declarative and procedural knowledge. Declarative knowledge consists of chunks of memory (e.g., apple is a kind of fruit), while procedural knowledge performs basic operations, moves data among buffers, and identifies the next instructions to be executed (e.g., to submit your answer, you have to click the submit bottom). When the model is driving a bus in a first-person perspective, these pieces of information will contain information such as what visual items presented to look at and what tasks to do next.

ACT-R is not complete, like all models. In this work we extend it to include new types of interaction knowledge and the capability to interact with all tasks that have a computer interface that is represented with a screen and that can be interacted with a keyboard and a mouse.

The Architecture of Interaction

Models interact with the world through their visual and motor systems. The interaction includes processing visual items presented (visual systems), pressing keys, and moving and clicking the mouse (motor systems).

Specifically, the visual system holds chunks of information about an object's location in the "where" buffer and chunks of information about objects in the visual scene in the "what" buffer. A central production system can reason about and lead to behavior based on these chunks. For example, the driving model may move forward or steer based on the position data retrieved from the visual buffer (Ritter et al., 2019).

Models can interact with the simulation, but the approach we will use is to use the screen's bitmap directly to find objects. Motor output can be put on the USB bus and appear as if a user at the keyboard typed characters or moved the mouse. In Table 2, we list previous models' history of interaction using this approach.

Table 2: Previous models history of interaction.

Name of model	Interaction tool	Reference
Eyes and Hands	ESegman	(Tehranchi & Ritter, 2017)
Biased coin	JSegman	(Tehranchi & Ritter, 2020)
Spreadsheet	JSegman	(Tehranchi, 2020)
Desert Bus 1	JSegman	(Schwartz et al., 2020)
Heads and Tails	VisiTor	(Bagherzadeh & Tehranchi, 2022)
Desert Bus 2	VisiTor	(this paper)

VisiTor (Bagherzadeh & Tehranchi, 2022) is a Python software package stored on a public GitHub that has been developed to provide simulated hands and eyes. It is comprised of two types of functions—motor and visual. The visual functions include “whatIsOnScreen”, which checks if certain visual patterns are present in the environment, "whereIs", which locates a pattern within a defined module, and "getMouseLocation", which retrieves the mouse's location. The motor functions consist of "click", which imitates a single mouse click, "Keypress," which replicates the pressing a key, "moveCursorTo", which emulates mouse movement to a specific screen location, and "moveCursorToPattern", which replicates mouse movement to a specific visual pattern.

The Simulation

Penn and Teller created the video game *Desert Bus* with the intention of making a statement about video games. The game is deliberately monotonous and lengthy, with the player driving a bus in real-time at a maximum speed of 45 mph from Tucson, AZ to Las Vegas, NV. Each leg takes at least eight hours, and the bus continuously drifts to the right. If the player swerves off the road, the engine will stall, and they will need to start over from Tucson. The game has no virtual passengers or other cars on the road. Once the player completes the eight-hour journey, the screen fades to black, and they return to the starting point to play again indefinitely. At night, the road is dark. Figure 1 provides a screenshot of the game available through Steam (there are other versions available now).

Figure 1: A Screenshot of the driver's view at approximately 10 min. into the game, oversteered.

This game offers the player a first-person view as they carry out tasks, and the surroundings change dynamically based on their actions. The specific edition that we use was created by Dinosaur Games and released by Gearbox Software, based on the unreleased "Smoke and Mirrors" Sega CD game. The game's driving environment, Desert Bus, was obtained from Steam (https://store.steampowered.com/app/638110/Desert_Bus_VR/) and can be downloaded for free on Windows machines. There were no alterations made to the game to support the model.

Desert Bus Model

To extend the amount of time the models could drive the bus, we created a more sophisticated model than Schwartz et al. (2020). We also explain the extensions to ACT-R's perceptual-motor system (VisiTor) to support this model, and then explain the details of the model.

Extending ACT-R 7 With VisiTor

This model is built in the latest version of ACT-R, ACT-R 7. It includes a Perceptual-Motor module (Bothell, 2011) that provides models with direct access to interfaces built in Macintosh Common Lisp (MCL). Therefore, by modifying modules, researchers can refine ACT-R 7 models to produce more complex behavior with neurologically compatible mechanisms. However, the current ACT-R PM module enabled models to interact only with MCL interfaces built with a window type provided with ACT-R, which limits their ability to interact with interfaces not created in that tool, such as *Desert Bus*.

To allow ACT-R 7 to access uninstrumented interfaces, a potential solution is to use VisiTor (Bagherzadeh & Tehranchi, 2022). It can simulate the user's visual attention (vision) as well as their use of a mouse and keyboard (motor). VisiTor functions as a vision manager tool that receives motor commands from the ACT-R PM module and sends them to the environment through an Emacs/slime link. By using this tool, ACT-R can engage with any environment while maintaining operations that are as similar as possible to those of the user. Additionally, VisiTor's capabilities can be expanded by incorporating modules into the tool.

ACT-R instructs VisiTor to scan the screen for particular pixel patterns that activate a production rule to initiate the program. Once VisiTor detects the start pattern, it sends a signal to ACT-R to begin running. Subsequently, ACT-R activates an "if-then" production rule that directs VisiTor to hold down the "W" key which starts the bus and accelerates when the start pattern is located in the visual environment. ACT-R then requests VisiTor to use the simulated hands to maintain pressure on the "W" key, effectively holding the bus's accelerator down. When VisiTor observes that the "right border of the road" objects deviate more than 200 pixels from the center of the road (approx. 5 degrees for someone 1.5 feet from the display), it signals to execute the steering production rule that specifies to hold down the "W" and "A" key until the car returns to the center of the road when the right edge of the road appears in the designated environment.

To undertake this task, VisiTor required a few minor extensions. It needed to simplify the process of describing visual objects and incorporate a range of visual objects. Furthermore, it had to transfer motor commands with a variable duration to maintain a keystroke. To support the novel task of driving a desert bus indefinitely, we implemented the "longpresskey" feature to VisiTor. This functionality enables the simulation of key-pressing actions, with the option of defining a duration of time to hold the key. (There are numerous other ways to implement this motor output, and we are also exploring those.)

The Driving Task

The tasks in Drive the Bus can be seen as occurring over three sections. (a) The player starts the game and starts driving the bus. (b) The player drives the bus from Tucson to Las Vegas. (c) After arriving in Las Vegas, the bus appears again at the end of an eight-hour stretch of road and starts again in an endless way.

This study reports the work of having the model do task (b), drive the bus from Tucson to Las Vegas. Tasks (a), and (c) will be reported later.

Driver model

Our objective was to redesign the model to make it do the long hours of driving. We employed the ADDIE (Morrison et al., 2010) framework for developing tutors (which we are familiar with) to create the model. ADDIE is a popular instructional design framework that can be adapted to create cognitive computational models. The ADDIE model consists of five stages: analysis, design, development, implementation, and evaluation.

In our analysis phase, the modeler gathers information about the target simulation environment, task objectives, and constraints. Based on the analysis, the modeler creates a plan for the model in the design phase, which includes the overall structure, content, and development strategies.

The development phase involves creating and refining the knowledge components and extended eyes and hands functions, such as declarative memories, production rules, visual patterns, and functions that will be utilized in VisiTor. Once the components analyzed are complete, the implementation phase involves delivering the model to the intended simulation environment.

Finally, in the evaluation phase, the modeler collects feedback and data to assess the effectiveness of the model and make improvements as needed. Using the ADDIE framework can help ensure that the computational cognition model is designed with the task in mind and are effective in achieving the desired simulation outcomes. It also encourages more intermediate products and buy-in from stakeholders and reflection, similar to the risk-driven spiral model (Pew & Mavor, 2007)

We consider two important pieces in the models' design. The first is how to represent the necessary knowledge for the model to be able to perform the task, and the other is the steps the model will perform to complete the task.

To start the model creation process, the task and simulation environment is analyzed by examining the game interface, 2 human subjects' keystrokes, and interkey intervals, as well as the visual cues and triggered actions. A list of declarative knowledge chunks representing visual cues and keypresses is formulated, such as "push the key to move forward," and a set of production rules representing the sequential actions that are triggered, such as "if the deviation of the bus exceeds 200 pixels, steer left". For example, we had two research assistants one Saturday afternoon literally drive the bus from Tucson to Las Vegas and attempt to record their behavior.

Additionally, supportive functions in VisiTor are developed by including a long-key-press operator. Then, the model is tested in the simulation, and necessary implementations are made, such as redefining visual cues.

Finally, the ACT-R output data are analyzed, and the model's performance is evaluated. The information collected from this evaluation is used to guide the iterative development of the model.

Below we describe the model we have written for this task. The following is a detailed explanation of how the model control loop is built, as well as the model's knowledge representations, actions to perform, and capabilities and functions used via the interaction architecture and VisiTor.

Control Loop

Figure 2 shows a flowchart of the mechanism underlying the model's control loop. It uses the visual buffer and simulated eyes to attend to and harvest the two visual objects, and then use the "whereis" function of VisiTor to encode the screen-x locations of the two objects. The model will then subtract the value of screen-x, and decide to steer left if the deviation is over 200 pixels.

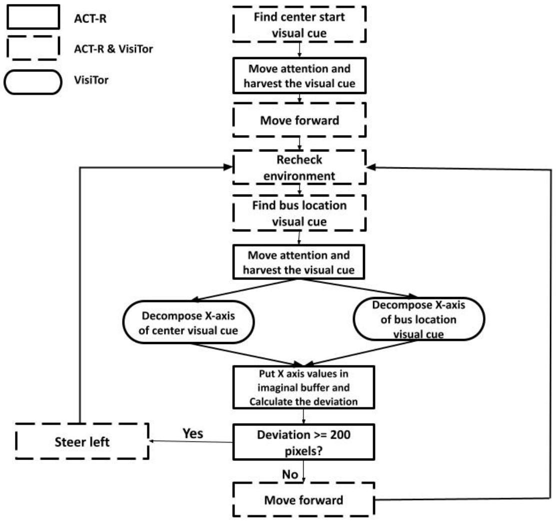

Figure 2: The control loop of the model.

Knowledge Representation

The model has two types of chunks, and a total of 12 declarative memories, which are working memories that tells the model to make the action based on the visual cues it saw. The first chunk is named "drive" and has two slots, "strategy" and "state", with state having parameters as object items. Another chunk type is "encoding", which has slots for the screen-x locations of the two visual cues and a deviation slot.

Actions to Perform

This model uses an explicit goal state to control the model. It contains 13 production rules. Table 3 list the high-level descriptions of the steps the model performs and the corresponding production rules.

Table 3 indicates that the model begins by investigating the simulation environment to locate and collect the visual cue necessary to initiate gameplay. It then utilizes the manual buffer to maintain forward motion by holding down the key. While doing so, the model continuously evaluates the environment to identify and gather visual cues related to the bus's location. It then breaks down the x-axis of the center line and bus location and computes the difference between the two values. If this difference is above 200 pixels, the model will hold down the key to turn the bus left; otherwise, it will just continue moving forward.

Table 3: High level description of the steps and the production rules that have been used in the model X.

High level descriptions of steps	Corresponding production rules
1. When it detects a start visual cue, attend it, and press the "W" key using the manual buffer	Go PerceiveEnvironment Move-attention Ahead
2. Clear the visual buffer and attend to the bus location	Recheck-environment Danger Finding-danger Move-attention-danger
3. Calculate the bus deviation from the center lane	Where-is-danger Where-is-center Calculate-deviation
4. Use the manual buffer by pressing "w" if the deviation is less than 200 pixels	Consider-ahead
5. Clear the manual buffer if the deviation > 200 pixels. Using the manual buffer, align the bus by pressing the key for 6 seconds.	Consider-steer
6. After that, clear the visual buffer, Repeat steps 1,2,3,4,5	Loop back to perceive-environment

Demonstration Observations

The experiment involved running a model to assess its performance and collect ACT-R output data. The model was found to be capable of running for hours in the long term. The declarative memories and production rules that were developed proved to be successful in meeting the needs of the simulation. Additionally, a new feature was incorporated into VisiTor, enabling the bus to accelerate with "w" held down.

However, if the driving speed exceeded the game setting, the ACT-R perceptual motor module in conjunction with simulated eyes and hands may not be able to identify, harvest, and process the location deviation as quickly as required.

As seen in Table 4, The ACT-R output data revealed that the model had an average time of 0.235 s to find a visual cue using the Visicon in conjunction with simulated eyes, an average time of 0.05 s to move attention to the visual cue, and an average time of 0.2 s to decompose the visual cue location and place it into the imaginal buffer. The total decision-making time of the model in gauging the deviation and making the next action decision of punching the keys was 0.9 s. This reflection time would be efficient for the model to identify danger cues and steer back to the road when the bus was driving at a maximum speed of 45 mph. One the order of minutes, this behavior can be compared to human behavior. On the order of hours we will see the model will outperform humans. This allows the model to accomplish a task in driving the bus that surpasses human capability, as it does not experience fatigue or mistakes (Gunzelmann, Moore, Salvucci, & Gluck, 2011).

Table 4: The output script of the ACT-R model that shows buffers, fired productions, and VisiTor commands with time stamps.

```
CL-USER>  (run 10)
          0.000  GOAL           SET-BUFFER-CHUNK GOAL GOER NIL
          0.000  VISION         SET-BUFFER-CHUNK VISUAL-LOCATION CHUNKO NIL
          0.050  PROCEDURAL     PRODUCTION-FIRED GO
          Ready to go
          0.100  PROCEDURAL     PRODUCTION-FIRED PERCEIVE-ENVIRONMENT
          0.150  PROCEDURAL     PRODUCTION-FIRED MOVE-ATTENTION
          0.150  VISION         SET-BUFFER-CHUNK VISUAL-LOCATION CHUNKO
          0.200  PROCEDURAL     PRODUCTION-FIRED AHEAD
          hiii
          continuouspress a key!
          0.200  MOTOR          PUNCH HAND RIGHT FINGER INDEX
          0.235  VISION         SET-BUFFER-CHUNK VISUAL CHUNK2
          0.250  PROCEDURAL     PRODUCTION-FIRED RECHECK-ENVIRONMENT
          0.285  VISION         SET-BUFFER-CHUNK VISUAL CHUNK3
          0.300  PROCEDURAL     PRODUCTION-FIRED DANGER
          0.350  PROCEDURAL     PRODUCTION-FIRED FINDING-DANGER
          0.350  VISION         SET-BUFFER-CHUNK VISUAL-LOCATION CHUNK!
          0.485  VISION         SET-BUFFER-CHUNK VISUAL CHUNK4
          0.535  PROCEDURAL     PRODUCTION-FIRED MOVE-ATTENTION-DANGER
          0.585  PROCEDURAL     PRODUCTION-FIRED WHEREISDANGER
          0.620  VISION         SET-BUFFER-CHUNK VISUAL CHUNKS
          0.185  IMAGINAL       SET-BUFFER-CHUNK-FROM-SPEC IMAGINAL
          0.835  PROCEDURAL     PRODUCTION-FIRED WHEREISCENTER
          1.035  IMAGINAL       SET-BUFFER-CHUNK-FROM-SPEC IMAGINAL
          1.085  PROCEDURAL     PRODUCTION-FIRED CALCULATE-DEVIATION
          1.135  PROCEDURAL     PRODUCTION-FIRED CONSIDER-STEER
```

In comparison to the driving the bus model created by Schwartz et al. (2020), there was a significant improvement in the accuracy of identifying visual cues in our model, as well as the ability to drive for a longer period of time. This model's better performance can be attributed to the following reasons.

To begin with, the ADDIE framework is a suitable choice for building the model for the current task because it helps in creating declarative memories, production rules, and control loop mechanism that closely resemble human driving behavior. It is important to distinguish between human driving behavior and human behavior. Human driving behavior pertains to how drivers use visual cues, such as the center line and bus location, to gauge deviation and make steering decisions. Human behavior is also determined by other psychophysiological factors, such as fatigue and decreasing correction rate, which are not included in this model yet and will be discussed separately. In this model, a superior control mechanism was implemented that replicates human driving behavior and allows for better long-term bus driving performance.

Furthermore, the integration of VisiTor into ACT-R 7 leads to enhanced coordination between perceptual and motor behavior. The entire process of ACT-R sending a request to prompt VisiTor to search the screen for the bus location visual cue, extract the location of the bus and center line, calculate the deviation, and decide on the next action can be completed in just 0.9 s. This time represents a significant improvement compared to the previous model, where JSegman was used in combination with ACT-R 6. According to Schwartz et al. (2020), the average time required to just match the visual template was already 6.01 s. Additionally, VisiTor's extensibility allows for the creation of new functions that support the specific requirements of the task, thereby considerably enhancing the model's performance. The long key press function, which is incorporated in this model, effectively enables the model to complete long-term tasks successfully by preventing ACT-R key presses from being interpreted as immediate press and release actions.

Nevertheless, a key factor that affects the model's performance is its limited ability to simulate activities in a dynamically changing environment. During gameplay, the environment undergoes dynamic changes, and at the near four-hour mark, the game environment shifts from a daytime mode to a nighttime mode, accompanied by a complete alteration in the visual pattern of the visual display. Although the PM module with simulated eyes can adapt to minor environmental changes, such as changes in road position or decorations along the roadside, the Visicon and VisiTor are not yet equipped to recognize an entirely different environment.

Discussion and Conclusion

The aim of this study was to employ ACT-R 7.X and its architecture of interaction to successfully complete a demanding cognitive modeling task of driving. The average run time for the model was one hour, with the longest run time lasting four hours until the gaming environment transitioned into night mode.

Instead of altering the game environment to accommodate ACT-R's MCL interfaces, we utilized the perceptual motor module of ACT-R 7 along with the vision and motor management software VisiTor to enable the model to play on the uninstrumented game interface.

We captured the model's gameplay and examined the ACT-R output, which demonstrated that the coordination between motor and vision using ACT-R 7 and VisiTor was highly effective, taking less than one second to steer the bus back into the safe range of the road. This was only feasible if the car's speed was not more than 45 mph (The speed limit in this game is 45 mph). We anticipate that if the bus speed is higher, then a shorter transport signal will be necessary for communication between ACT-R and VisiTor.

We have found that the superior human behavior model has implied limitations in the ACT-R perceptual-motor modules.

There are also limitations in the central modules (production rules). These limitations include the lack of consideration for physiological factors such as fatigue or decreasing correction rates over time. In a study by Schwartz et al. (2020), it was suggested that incorporating physiology with ACT-R could make the model more realistic. We agree with this point and plan to add that in our future work. This new approach can help with testing the compound effects of fatigue and learning rate on our model.

ACT-R + VisiTor playing *Drive the Bus* provides an excellent platform for studying the interaction of vision, attention, errors, and fatigue. It is a more naturalistic task than the PsychoMotor Vigilance task (PVT, Dinges & Powell, 1985). We can now explore an existing fatigue model (Gunzelmann, Gross, Gluck, & Dinges, 2009), and examine fatigued driving (e.g., Gunzelmann, Moore, Salvucci, & Gluck, 2011), visual attention, the need for micuration, and modeling the details of interaction.

We could gain understanding about how long-term and repetitive physical activities, like driving a bus for an extended period, affect human performance. It remains to be seen if this task is more like the PVT or like motor control (Bolkhovsky, Ritter, Chon, Qin, 2018). This task would also allow us to determine whether psychological factors could potentially harm or the increasing of learning rate due to the practice would enhance driving skills. We could also introduce additional variables, such as caffeine, to examine their combined impact.

References

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, *51*(4), 355.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press.

Anderson, J. R., & Douglass, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *27*, 1331–1346. https://doi.org/10.1037/0278-7393.27.6.1331

Bagherzadeh, A., & Tehranchi, F. (2022). Comparing cognitive, cognitive instance-based, and reinforcement learning models in an interactive task. *Proceedings of ICCM, The 20th International Conference on Cognitive Modeling.* 1-7.

Bolkhovsky, J. B., Ritter, F. E., Chon, K. H., & Qin, M. (2018). Performance trends during sleep deprivation. *Aerospace Medicine and Human Performance, 89*(7), 626-633(8).

Bothell, D. (2017). ACT-R 7 reference manual. Available at actr.psy.cmu.edu/wordpress/wpcontent/themes/ACT-R/actr7/reference-manual pdf, Accessed February 2023.

Byrne, M. D., O'Malley, M. K., Gallagher, M. A., Purkayastha, S. N., Howie, N., & Huegel, J. C. (2010). A preliminary ACT-R model of a continuous motor task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *54*(13), 1037–1041. https://doi.org/10.1177/154193121005401308

Dinges, D. I., & Powell, J. W. (1985). Microcomputer analysis of performance on a portable, simple visual RT task sustained operations. *Behavior Research Methods, Instrumentation, and Computers*, *17*, 652–655.

Fleetwood, M. D., & Byrne, M. D. (2002). Modeling icon search in ACT-R/PM. *Cognitive Systems Research, 3*(1), 25–33. https://doi.org/10.1016/S1389-0417(01)00041-9

Gunzelmann, G., Gross, J. B., Gluck, K. A., & Dinges, D. F. (2009). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science, 33*(5), 880-910.

Gunzelmann, G., Moore, L. R., Salvucci, D. D., & Gluck, K. A. (2011). Sleep loss and driver performance: Quantitative predictions with zero free parameters. *Cognitive Systems Research, 12*, 154-163.

Laird, J. E. (2019). *The Soar cognitive architecture*. Cambridge, MA, MIT Press.

Morrison, G. R., Ross, S. M., Kahlman, H. K., & Kemp, J. E. (2010). *Designing effective instruction* (6th Edition ed.). Hoboken, NJ: John Wiley & Sons.

Pew, R. W., & Mavor, A. S. (Eds.). (2007). *Human-system integration in the system development process: A new look.* Washington, DC: National Academy Press. books.nap.edu/catalog/11893, checked May 2019.

Newell, A. (1990). *Unified Theories of Cognition.* Harvard University Press.

Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Supporting cognitive models as users. *ACM Transactions on Computer-Human Interaction, 7*(2), 141–173. https://doi.org/10.1145/353485.353486

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *WIREs Cognitive Science, 10*(3), e1488. https://doi.org/10.1002/wcs.1488

Ritter, F. E., Van Rooy, D., Amant, R. St., & Simpson, K. (2006). Providing user models direct access to interfaces: An exploratory study of a simple interface with implications for HRI and HCI. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 36*(3), 592–601. https://doi.org/10.1109/TSMCA.2005.853482

Schwartz, D. M., Tehranchi, F., & Ritter, F. E. (2020). Drive the bus: Extending JSegMan to drive a virtual long-range bus. In *Proceedings of the 18th International Conference on Cognitive Modeling (ICCM 2020)*. 241-246.

Tehranchi, F., & Ritter, F. E. (2017). An eyes and hands model for cognitive architectures to interact with user interfaces. In *MAICS* (pp. 15-20).

Tehranchi, F., & Ritter, F. E. (2018). Modeling visual search in interactive graphic interfaces: Adding visual pattern matching algorithms to ACT-R. In *Proceedings of ICCM-2018-16th International Conference on Cognitive Modeling* (pp. 162-167)

Tehranchi, F. (2020). An Eyes and Hands Model: Extending Visual and Motor Modules for Cognitive Architectures (Doctoral dissertation, The Pennsylvania State University).

Wilson, D., & Sicart, M. (2010). Now it's personal: On abusive game design. *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, 40–47. https://doi.org/10.1145/1920778.1920785