Cognitive Computing and Internet of Things, Vol. 165, 2025, 169–178
https://doi.org/10.54941/ahfe1006100

CogDriver: The Longest-Running Autonomous Driving Cognitive Model Exhibits Human Factors

Christian Wasta, Siyu Wu, and Frank E. Ritter

College of IST, Penn State, University Park, PA 16802, USA

ABSTRACT

We are exploring how models can use models of human perception and motor control to interact directly with interfaces. We present CogDriver, a cognitive driving model capable of performing a long-duration autonomous driving task in a virtual simulation environment. This model, built using the ACT-R cognitive architecture and enhanced with robotic hands and eyes, supports the cognitive-perceptual-motor knowledge essential for simple human driving. It has two main strengths compared to other autonomous driving models: (a) it is built upon human-observed driving behavior, incorporating error-making and learning, and (b) it leverages a cognitive architecture to provide insights into psychological driving behavior. Compared to our previous version, this model shows improved endurance, maintaining its driving state for over 18 h from Tucson to Las Vegas, even under nighttime conditions. The enhancements were realized through incorporating human-like driving knowledge representations, and actions. It now includes a model of error handling and several logical visual cue strategies. The model's predictions can match certain aspects of human behavior in fine detail, such as the number of course corrections, average speed, learning rate, and adaptation to low visibility conditions. This model demonstrates that (a) perception and action loops with fallback handling provide a very accessible testbed for examining further aspects of behavior and (b) the model-task combination supports exploring aspects of human behavior that remain missing from ACT-R. Model, simulation, and data can be accessed at https://github.com/christianwasta/DriveBus/tree/drivebus-wasta.

Keywords: Cognitive autonomous driving, ACT-R cognitive architecture

INTRODUCTION

Current cognitive architectures provide many practical use cases across fields including obstacle avoidance and navigation (Kotseruba & Tsotsos, 2020; in press). Our approach through cognitive modeling provides the opportunity to add human factors until a human-like autonomous simulation is made. Because cognitive architectures (CAs) can develop cognitive models of various psychological phenomena and tasks (Newell, 1990), they also provide procedures and structures that align with human behavior, such as reaction times, error rates and types, and fMRI results.

This paper develops a human-like cognitive model that can perform a long-term autonomous driving task in a virtual simulation environment.

The model has two strengths compared to other autonomous driving models on similar tasks: (a) it built upon human-observed driving behavior, incorporating error-making and learning, and (b) it leverages a cognitive architecture to provide insights into psychological driving behavior.

The CogDriver architecture is shown in Figure 1, which shows a closed-loop system where cognitive reasoning (through production and declarative memory) interacts with perceptual and motor processes. It begins with the (a) ACT-R Cognitive Architecture (Anderson, 2007; Ritter et al., 2019), which serves as the foundation for building the model. This architecture is composed of a (a1) cognition layer with production memory, which encodes human subjects' procedural knowledge for decision-making, actions and directing attention to specific targets within the environment. And declarative memory, which stores subjects' factual knowledge, retrieved visual information, and provides motor intuition to guide task execution.

The perceptual/motor layer (a2) includes a vision manager, which manages visual attention and perception by instructing the eyes to focus on specific locations in the environment. The visual system processes chunks of information about an object's location in the "where" buffer and information about objects in the visual scene in the "what" buffer. The motor manager coordinates motor actions, such as steering and keypresses, based on instructions from the cognition layer. The cognition and perceptual/motor layers are tightly integrated. The central production system can reason about chunks of information stored in the visual buffers to guide behavior. In a driving context, this enables the model to move forward, or steer based on position data retrieved from the visual buffer. However, the ACT-R model is not complete with the restriction that interaction knowledge cannot work on unaltered tasks. In this work, we extend it to include new types of interaction knowledge and human behavior aspects that were previously missing.

To extend the interaction knowledge of the cognitive model, we use the (b) interaction management layer to facilitate the synergy between visual and motor functions. This layer allows the cognitive model to process inputs and execute outputs through visual functions (e.g., whatIsOnScreen rule identifies visual patterns, whereIs locates patterns, and getMouseLocation tracks the mouse's position) and motor functions (e.g., click simulates mouse clicks, Keypress replicates key presses, moveCursorTo moves the mouse to a location). These functions enable the system to interact directly with an (c) unaltered simulation environment, using the screen's bitmap to detect objects and respond accordingly through passing mouse moves, clicks, and keypresses onto the simulation vehicle. For example, the system analyzes pixels or symbols to identify objects and locations on the screen, guiding motor responses like steering or clicking. The integration of these components ensures the model can dynamically adapt to and interact with its environment in a human-like manner guided by the cognitive model. Compared to the previous cognitive driving model for the same simulation environment (Wu et al., 2023), the model has demonstrated improved endurance, maintaining its driving state for over 18 h, even under dynamic changes in the simulation environment (from day to night), achieved through incorporating human factors in the ACT-R model and interaction layer.

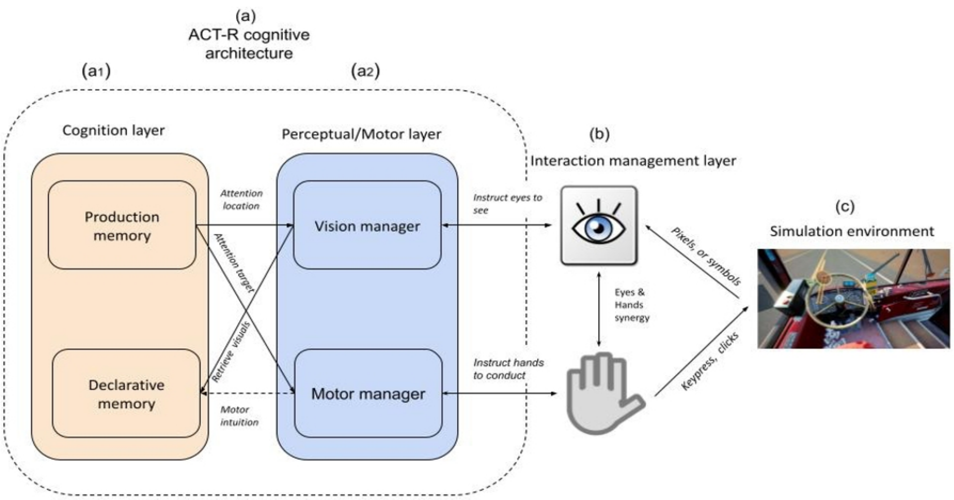

Figure 1: The CogDriver architecture, composed of (a) a CA that has cognitive layer encoding procedural and declarative knowledge and a vision and motor layer encoding perceptual and motor knowledge; (b) an interaction management layer that bridges the CA and the (c) simulation environment.

The following section introduces the task, the model design and development, the model evaluation, and the implications in the field of autonomous driving.

THE TASK

Penn and Teller created the video game, Desert Bus, with the intention of making a statement about video games. The game has the player drive a bus in real-time at a maximum speed of 45 mph from Tucson, AZ to Las Vegas, NV. Each leg takes 360 miles to complete, or at least eight hours at maximum speed, and the bus continuously drifts to the right. If the player swerves off the road, the engine stalls, and the bus starts over from Tucson. If the driver pauses, the bus is towed back as well. The game has no virtual passengers or other cars on the road. Once the player completes the journey, the screen fades to black, and they return to the starting point to play again indefinitely. At night, the road is dark. Figure 1c provides a screenshot of the game (daytime). Figure 2b shows a night time view.

This game offers the player a first-person view as they carry out tasks, and the surroundings change dynamically based on their actions. We used the latest version created by Dinosaur Games and released by Gearbox Software, based on the unreleased "Smoke and Mirrors" Sega CD game. The game, Desert Bus VR, can be downloaded for free on Steam for Windows machines only (store.steampowered.com/app/638110/Desert_Bus_VR/). Desert Bus VR supports both virtual reality (VR) and 2D (non-VR) environments. All testing was done in the non-VR environment although future work could expand support for virtual reality headsets. There were no alterations made to the game to support the model.

RELATED WORK

There is related work in cognitive architectures and models of interactions.

Cognition Architecture and Cognitive Models

To create a cognitive driving model, we bring a suite of tools rooted in cognitive architecture (CAs). CAs are computational frameworks designed to capture the invariant mechanisms of human cognition. These mechanisms include functions related to attention, control, learning, memory, adaptivity, perception, and action. Cognitive architectures propose a set of fixed mechanisms to model human behavior, functioning akin to agents and aiming for a unified representation of the mind. By using task-specific knowledge, these architectures not only simulate but also explain behavior through direct examination and real-time reasoning tracing. One representative cognitive architecture is ACT-R, a theory of human cognitive mechanisms embodied in the ACT-R program, through which we can construct models that can store, retrieve, and process knowledge, as well as explain and predict performance (Anderson, 2007). There are currently two primary kinds of knowledge representations in ACT-R, which are declarative and procedural knowledge. Declarative knowledge consists of chunks of memory (e.g., apple is a fruit), while procedural knowledge performs basic operations by moving data among buffers, like identifying the next instructions to be executed (e.g., to submit your answer, you must click the submit button). When the model is driving a bus in a first-person perspective, these pieces of information will contain information such as what visual items 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. We will also note new limitations with it that are made clear from this work.

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 (Tehranchi & Ritter, 2018).

Models can interact directly with the simulation (e.g., Jones et al., 2000), but our approach uses the screen's bitmap to find objects. Motor output is put on the USB bus and appears as if a user typed characters or moved the mouse. Table 1 lists a history of previous cognitive models that interact using this approach.

Table 1: Previous cognitive models using this approach to interaction.

Name of Model	Interaction Tool	Reference
Eyes and Hands	ESegman	(Tehranchi & Ritter, 2017)
Biased coin	JSegman	(Tehranchi & Ritter, 2020)
Spreadsheet	JSegman	(Tehranchi & Ritter, 2020)
Desertbus 1	JSegman	(Schwartz et al., 2020)
Heads and Tails	VisiTor	(Bagherzadeh & Tehranchi, 2022)
DesertBus 2	VisiTor	(Wu et al., 2023)
CogDriver	VisiTor	(This paper)

This study uses VisiTor (Bagherzadeh & Tehranchi, 2022) as a vision implementation that receives motor commands from the ACT-R PM module and sends them to the environment through an Emacs/SLIME link. VisiTor is a Python software package stored on a public GitHub repository. It was developed to provide simulated hands and eyes. It provides motor and visual interaction.

By using VisiTor, ACT-R can engage with any environment while maintaining operations similar to those of the user. For example, in the driving simulation, when visual patterns are detected, ACT-R executes production rules that control the bus through a combination of continuous forward movements (via the "W" key) and steering controls (via the "A" key). For example, ACT-R instructs VisiTor to scan the screen for pixel patterns that activate a production rule to initiate the program.

COGDRIVER

This section starts with capturing intuition and domain knowledge from the human subjects, followed by the model structure and learning mechanism, and concludes by examining a model's driving performance.

Incorporating Human Factors into Model Design

The model, built upon human factors distilled from the behavior of human subjects in driving simulations, incorporates principles of cognitive model design for human-like driving simulations. Data collection was conducted (a) using RUI—a simple keystroke and mouse movement logger (Kukreja et al., 2006)—to capture keystroke patterns, and (b) through post-game interviews to investigate how human subjects steered during gameplay. We analyzed the subjects' reported eye movements and mapped the reported visual cues into the ACT-R model's visual module. Additionally, we analyzed the key press time intervals and incorporated this data into the design of the hands' keypress behavior.

First, VisiTor now supports short and long key presses, a key motor factor. This allows the model to execute motor actions based on detected deviations. A short key press corrects minor drifts, while a long key press adjusts for larger deviations, mimicking human motor control in driving.

Second, our extension to ACT-R integrates error logging and a continuously looping event procedure. Error logging records failures and unexpected events, improving future iterations by addressing issues like missed visual cues or excessive deviations. The looping procedure enables continuous environmental monitoring and dynamic adaptation, mirroring human self-monitoring, where drivers constantly scan their surroundings and adjust accordingly. These features enhance resilience and adaptability, key traits of human-like decision-making in driving.

Declarative Chunks

The model has two types of chunks, and a total of 12 declarative memories, which are working memories that tell 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.

Procedural Memories

Figure 2 shows how the control loop of the cognitive model simulate a dynamic driving scenario by processing visual cues and making corrective actions. The loop begins with detecting left and right road line visual cues, followed by shifting attention to harvest detailed information. The model then moves forward and rechecks the environment to ensure updated visual data. It identifies specific elements, such as the bus location, processes the positions of the left and right road lines, and calculates the deviation of the vehicle’s position from the center of the lane. If the deviation exceeds 300 pixels, a long press is applied to steer left for significant correction; otherwise, a short press is used for minor adjustments. After these corrections, the model executes a long forward press to maintain its course and loops back to reassess the environment, creating a continuous cycle of attention, evaluation, and action to simulate human-like driving behavior.

To improve the model’s performance at night, the revised cognitive model marked improvements through a few architectural and behavioral areas. First, it transitioned from a single-reference visual system (monitoring the center yellow line) to a dual-reference system that tracks both the left and right white road lines. This improvement was realized by adding productions that failed to track road lines frequently due to a lack of redundancy. ACT-R now detects both left and right road lines, and their coordinates are sent to the Emacs/SLIME link to initiate the driving productions.

Second, additional modifications to VisiTor were made to enable the model to see and act more human. The “continue-cycle” and “handle-missing-cue” procedures imitate how humans self-monitor driving and steering by adding continuous operation mechanisms. Demonstrated in the “consider-ahead” production, deviations from the left road line of less than 300 pixels now result in a “short-keypress,” which keeps the model in the right road lane by adding error-handling for eyes and hands synergy.

The model uses an explicit goal state to control the model. It contained 13 production rules. Table 2 lists the high-level descriptions of the steps the model performs and the corresponding production rules.

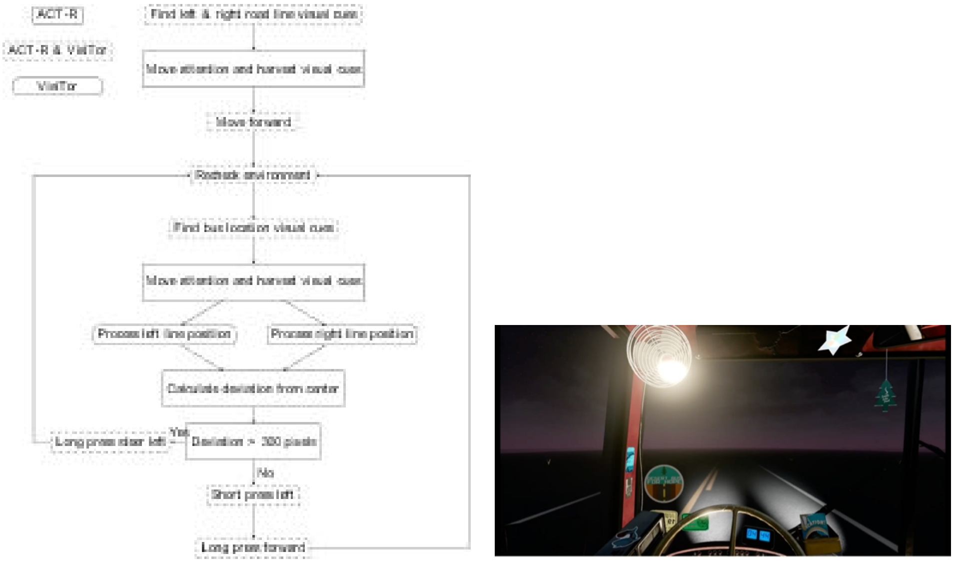

Figure 2: Control loop of the model (a, left) and the driver's view at night (b, right).

Model Evaluation

The model was tested multiple times in nighttime conditions. After testing the model by running the bus in the middle of the night, a second test was run from the game's start to finish. At approximately 8 h of driving, the environment is visibly dark, but the automatic headlights have yet to turn on (whereas they are on in Figure 3) . Long and short duration key presses are essential at 8 h because the bus's speed is reduced, and the bus reliably stays in the same lane. The introduction of the 'my-short-keypress' function represents a more human-like driving behavior that gives VisiTor more processing time to differentiate yellow from white road lines. Instead of relying solely on continuous steering inputs, the model now makes brief, corrective adjustments when steering left.

As seen in Table 3, The ACT-R output data revealed that the total decision-making time the model took to detect the first visual cue to action execution was 0.450 s. The previous model's action execution time of 0.90 s makes this model twice as fast, slowly making the model's reaction time equivalent to the average of a human, which is around 250 ms.

Testing the model revealed that the bus could now be driven for the entire 360-mile distance. Including the night time environment in Figure 2b using the improved visual detection.

Table 2: High level description of the steps and the production rules.

High Level Step Descriptions	Corresponding Productions
1. When it detects a start visual cue, attend it, and press the "W" key using the manual buffer	Go PerceiveEnvironment Move-attention Ahead

Continued

Table 2: Continued

High Level Step Descriptions	Corresponding Productions
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. Checks if both cues are missing and continues to move forward if true	Handle-missing-cue – Ensures the model does not stop if visual information is not immediately found when started
5. Use the manual buffer by pressing "w" (default, about 100 ms) if the deviation is less than 300 pixels	Consider-ahead
6. Clear the manual buffer if the deviation > 300 pixels. Using the manual buffer, align the bus by pressing the "w" key for 6 s.	Consider-steer
7. Sets state to perceive and resets goals	Continue-cycle – Prevents dead-end states where no production can fire due to "consider-steer" not firing.

So far, the model has been able to achieve one leg after driving for 18.5 h at an average speed of 20 mph. While this is lower than the top speed of 45 mph, future adjustments and enhancements to the visual cue recognition will naturally make the bus drive faster, because the model presses the accelerator after every steering decision.

DISCUSSION AND CONCLUSION

We start by summarizing the contributions and then note the limitations and future work. CogDriver operates for an average of 18.5 h, with slower versions exceeding 24 h. This demonstrates how cognitive architectures can maintain reliable performance while adapting to environmental changes. After several hours, the model eventually outperforms humans, which raises interesting issues. CogDriver marks an advancement in autonomous driving cognitive model. First, the integration of human behaviors to the model through cognitive architecture is achieved by adding behavioral error handling and learning improvements to the ACT-R model. Second, the model's ability to sustain an 18-h driving record was achieved by human-like driving knowledge representations, error handling mechanisms, and new visual cues. Finally, this improvement showcases the capability to establish human behavioral models by examining human perception, action loops, and fallback procedures.

These contributions advance both autonomous driving research and cognitive modeling, showing how integrating human factors and psychological insights can improve the performance and reliability of autonomous systems, particularly in challenging conditions that have traditionally been difficult for cognitive models to handle.

Limitations and Future Work

One limitation is the need to improve the spatial memory system to assist when the bus drifts too far left. Another is the absence of physiological factors such as fatigue and declining correction rates over time. Schwartz et al. (2020) suggest integrating physiology into ACT-R for greater realism (e.g., use of ACT-R/Phi, Dancy et al., 2015). We support this and propose testing the combined effects of fatigue and learning rate on the model (Wu et al., 2023).

The ACT-R+VisiTor platform provides a naturalistic setting for studying vision, attention, errors, and fatigue—more so than the PsychoMotor Vigilance Task (PVT; Dinges & Powell, 1985). Future work could incorporate a fatigue model (Gunzelmann et al., 2009) to study fatigued driving (Gunzelmann et al., 2011), sleep restriction effects (Bolkhovsky et al., 2018), visual attention, and other human factors. This setup also offers insights into how prolonged, repetitive activities like bus driving affect performance.

ACKNOWLEDGMENT

The authors would like to thank Amir Bagherzadeh for help with model wrangling.

REFERENCES

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

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).

Dancy, C. L., Ritter, F. E., Berry, K., & Klein, L. C. (2015). Using a cognitive architecture with a physiological substrate to represent effects of psychological stress on cognition. *Computational and Mathematical Organization Theory, 21*(1), 90–114.

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.

Jones, G., Ritter, F. E., & Wood, D. J. (2000). Using a cognitive architecture to examine what develops. *Psychological Science, 11*(2), 93–100.

Kotseruba, I., & Tsotsos, J. K. (2020). A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications. *AI Review, 53*, 17–94.

Kotseruba, I., & Tsotsos, J. K. (in press). *The computational evolution of cognitive architectures*. Oxford, UK: Oxford University Press.

Kukreja, U., Stevenson, W. E., & Ritter, F. E. (2006). RUI—Recording User Input from interfaces under Windows and Mac OS X. *Behavior Research Methods, 38*(4), 656–659.

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

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

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., Bagherzadehkhorasani, A., & Ritter, F. E. (2023). A user model to directly compare two unmodified interfaces: A study of including errors and error corrections in a cognitive user model. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 37*(e27), 1–11.

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. (2020). Extending JSegMan to interact with a biased coin task and a spreadsheet task. In *17th International Conference on Cognitive Modeling, ICCM 2019*. 259–260. Applied Cognitive Science Lab, Penn State.

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*. 162–167.

Wu, S., Bagherzadeh, A., Ritter, F. E., & Tehranchi, F. (2023). Long road ahead: Lessons learned from the (soon to be) longest running cognitive model. In *Proceedings of 21st International Conference on Cognitive Modeling (ICCM)*, 281–287.