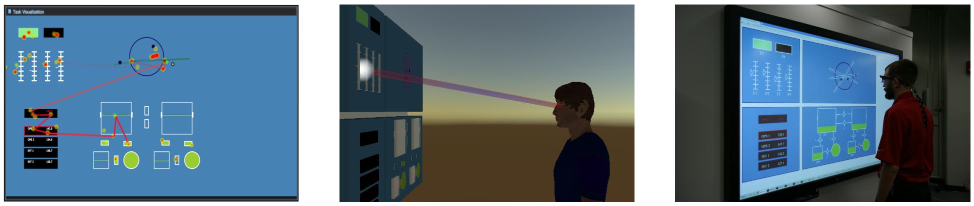
Visualizing Eye Movements in Formal Cognitive Models

J. Timothy Balint, Brad Reynolds, Leslie M. Blaha, Tim Halverson



(a) (b) (c)

Fig. 1: Visualizing eye tracking data generated by cognitive models of visual multitasking. (a) The model's internal representation of the experiment with its visual scanpath drawn in red. (b) Virtual simulation of a person, whose eye movements are powered by a model of visual attention. (a) and (b) simulate (c) a typical human user, like the one depicted here, performing the multitasking experiment wearing eye tracking equipment).

Abstract—We present two visualization approaches illustrating the value of formal cognitive models for predicting, capturing, and understanding eye tracking as a manifestation of underlying cognitive processes and strategies. Computational cognitive models are formal theories of cognition which can provide predictions for human eye movements in visual decision making tasks. Visualizing the internal dynamics of a model provides insights into how the interplay of cognitive mechanisms influences the observable eye movements. Animation of those model behaviors in virtual human agents gives explicit, high fidelity visualizations of model behavior, providing the analyst with an understanding of the simulated human's behavior. Both can be compared to human data for insight about cognitive mechanisms engaged in visual tasks and how eye movements are affected by changes in internal cognitive strategies, external interface properties, and task demands.

Index Terms—ACT-R, Cognitive architectures, Cognitive model visualization, Eye tracking, Virtual agents.

1 Introduction

Eye tracking technology provides a critical data source for the design and evaluation of visual analytics tools. The efficacy of information visualizations for human discovery, insight, and decision making is driven by a visualization's ability to successfully leverage perceptual and cognitive mechanisms [12]. Eye movements provide invaluable non-invasive measures of attention and visual information processing for assessing visualization interface efficacy. However, teasing apart the mechanisms supporting the observed behavior can be difficult based on eye tracking alone. Computational cognitive models provide powerful tools for understanding visual processes in complex tasks. Cognitive models are formal instantiations of theories about how the mind functions and operates in the physical world [3]. They capture perceptual, cognitive, and motor behaviors to produce predictions about human behavior, such as response choice and accuracy, response speed, manual motor activity, and eye movements. Critically, because of their computational nature, models can be extensively explored at a low cost to researchers in ways that provide specific predictions of human behavior.

Computational cognitive models can perform tasks with the same visual interface environments as human users and can be designed to produce eye movements similar to humans. Thus, we can use basic units of analysis like fixation locations, dwell times, and scanpaths, to study both human and model eye movements. In addition, if we can visualize the internal dynamics of the model, we can begin to gain insights into the underlying processes producing the observable eye behaviors. For example, the order of operations within the model highlight whether mental processes are causing eye movements or external events are diverting eye movements and triggering mental processes. For example, if an item in the memory process is activated before an eye movement process is activated, then we can infer that memory was causing the agent to shift their focus of attention in the display. In this way, we gain some understanding about the cognitive processes underlying visual task performance and how they are affected by interface attributes.

Visualization plays a critical role in elucidating the eye movement behavior from formal models of human cognition as seen across Figure 1. The goal of the present work is to leverage two very different types of visualization for an initial exploration of the complex interplay of cognitive activity and observable eye movements. We hypothesize that a combined approach will enable three types of insight. Insights about the cognitive mechanisms are gained by visualizing internal model dynamics during task performance together with model eye movement behavior. Insights about model predictions for the physical human eye movements themselves are gained by embodying model behavior in virtual human agents. Finally, insights about cognitive strategies, model validity, and realism are gained by simultaneously visualizing and embodying model eye movements, comparing those to human eye movement behaviors, or directly comparing the dynamics and predicted movements of multiple candidate models. To enable this exploration, we present two visualization tools: a Dashboard system for capturing internal model dynamics supporting eye movement behavior, and an approach to virtually embodying model eye movements in human agents.

Both visualization techniques leverage the client-server software Simplified Interfacing for Modeling Cognition - JavaScript (SIMCog-JS) [7]. SIMCog currently connects JAVA ACT-R (see next section) with a JavaScript-based multitasking environment. This environment, shown on the touchscreen in Figure 1c, contains four tasks requiring

- *J. Timothy Balint is with George Mason University. E-mail: jbalint2@gmu.edu.*
- *Brad Reynolds is with Wright State University. E-mail: reynolds.157@wright.edu.*
- *Leslie M. Blaha is with Air Force Research Laboratory. E-mail: leslie.blaha@us.af.mil.*
- *Tim Halverson is with Oregon Research in Cognitive Applications, LLC. E-mail: thalverson@gmail.com.*

continuous attention by the user to different alerts. Clockwise from upper left, these are a Monitoring task (response to out-of-state visual cues), Tracking task (continuously track an object with the mouse), Resource Management task (keep fuel levels within indicated range by controlling flow), and Communications task (change channel values when cued). All four quadrants entail visual alerts that can grab visual attention, and correct responses to the tasks require an observer (human or model) to periodically scan between tasks for those alerts. In the present work, to demonstrate the visual analytics and virtual embodiment of model behaviors, we utilize this task environment to simulate eye movements during multitasking from two models and illustrate the visualizations to compare model behaviors. In order to do this, we captured the SIMCog message streams. The internal activity of ACT-R is parsed in SIMCog and saved to a CSV for Dashboard analytics. The JSON task environment change messages are sent into the virtual embodiment task display. Model events such as eye locations and key presses are pushed into JSON messages and sent to the virtual embodiment. The result of this is that while ACT-R is multitasking, we can simultaneously visualize its internal activity and visualize the predictions for human eye movement behaviors. Before we introduce our visualizations, we review the characteristics of ACT-R.

2 Adaptive Control of Thought-Rational (ACT-R)

One way to model human activity is with cognitive architectures. The cognitive architecture used in the current research is ACT-R [1]. Other cognitive architectures, like EPIC [9] and Soar [10], or other modeling formalisms, like Guided Search [20], could also be utilized for such research. The methods for visualizing model behavior generalize across choice of models, and can serve as a method to compare candidate eye movement models.

ACT-R is a general theory of human cognition, including cognitive, perceptual, and motor processes. The ACT-R cognitive architecture is a computational instantiation of that theory. Figure 2 illustrates a box-and-arrow representation of ACT-R. The cognitive architecture is used to build models that simulate how people perform tasks given the cognitive, perceptual, and motor constraints provided by ACT-R and the dynamics of the task with which ACT-R interacts, much as a human would. For the current discussion, it is critical that ACT-R produce eye movements. ACT-R includes an implementation of Eye Movements and Movement of Attention (EMMA) [16], one theory that links eye movements to the covert shifts of visual attention produced by ACT-R. Other theories of eye movements, like PAAV [11], could be utilized within ACT-R.

In ACT-R, covert shifts of attention are required for visual object encoding. In EMMA, object eccentricity, with respect to the fixation location of the eyes, and how frequently the object has been attended affect the time to encode the object. Eye movements tend to follow covert attention. However, an eye movement to an attended object might not occur if the covert attention to that object is brief, and extra eye movements (i.e., corrective saccades) can occur when the eyes overshoot or undershoot the center of covert attention.

The strategies that ACT-R models bring to bear on a task are encoded in production rules. Production rules are fine-grained representations of procedural knowledge that “fire” when the rule’s conditions are met. This firing can change the model’s internal state and initiate external actions. For example, a rule might specify that whenever a red object appears and visual attention is not busy encoding something else, then shift attention (and perhaps the eyes) to that red object. Only one rule may fire at a time, and which rule fires is primarily determined by the contents of the buffers (internal storage units). ACT-R includes a number of modules and associated buffers that instantiate theory related to specific cognitive processes (e.g., declarative memory). The modules make predictions about the timing and accuracy associated with the specific cognitive process. It is the interaction among the modules, buffers, production rules, and external environment that generate predictions of human behavior.

The output of model simulations includes behavioral data similar to that produced in human observations, such as response time, error rate, and eye movements. In addition, the cognitive architecture provides a detailed trace of the perceptual, motoric, and cognitive processes recruited by the simulated human to produce that behavior.

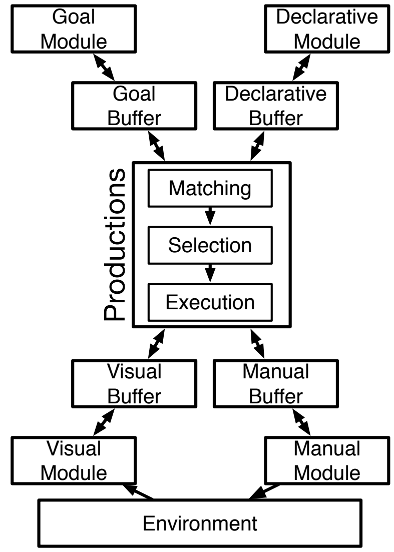


Fig. 2: Adaptive Control of Thought-Rational (ACT-R) [1] diagram

The use of formal cognitive models, such as ACT-R, allows one to explore how visual behavior is determined through the interaction of perceptual processes, manual motor interaction, and other cognitive processes like associative memory. An analyst encodes their hypotheses of how people accomplish a task in computational models, runs numerous simulations with those models, and then compares the results of the simulation with human data from the same task to help confirm or deny those hypotheses. While the use of such models does not fully confirm the underlying hypotheses, previous research has shown the utility of this methodology for understanding, among other things, how people visually scan graphs [14], visually interleave complex dual-tasks [21], and visually search menus [6]. Visualizing the model’s simulated eye movements, along with the detailed trace of the processes that produced those scanpaths, can provide additional insight into how people accomplish visual tasks.

3 Visual Analytics Dashboard

We present a Model Visualization Dashboard to visualize the internal and eye movement activity of cognitive models. To populate the Dashboard, model states are stored from SIMCog into a time series data set, stored in a CSV file. These data contain information about the internal states of the model, requests for interaction with the interface, and changes in interface elements “perceived” by ACT-R. By reading this CSV, the Dashboard enables playback of ACT-R states and behaviors over the course of the task, and provides a human user the ability to examine the data using different performance metrics. The Dashboard visualizes the raw data file in a web-browser application written in Node.js. The Dashboard interface uses common web languages, such as HTML and CSS, and JavaScript libraries (e.g., D3 [2]) to capture different data types within customizable dynamic gauges.

The Model Visualization Dashboard is illustrated in Figure 3; two models are visualizations are shown side-by-side, each containing a set of gauges capturing model behaviors. The Dashboard can replay a model performing tasks, and playback controls are given at the top of the screen. Thus, we get real-time and summary visualizations of the model behaviors. Total task execution time is given in the radial Task Time gauge.

The central Task Visualization gauge shows the model’s visicon, its internal visual representation. Both the recent eye movements (red line) and mouse movements (green line) are overlaid on the Task. These paths are transient and show the recent few seconds of activity. A heat map is also overlaid to show all fixations from the start of

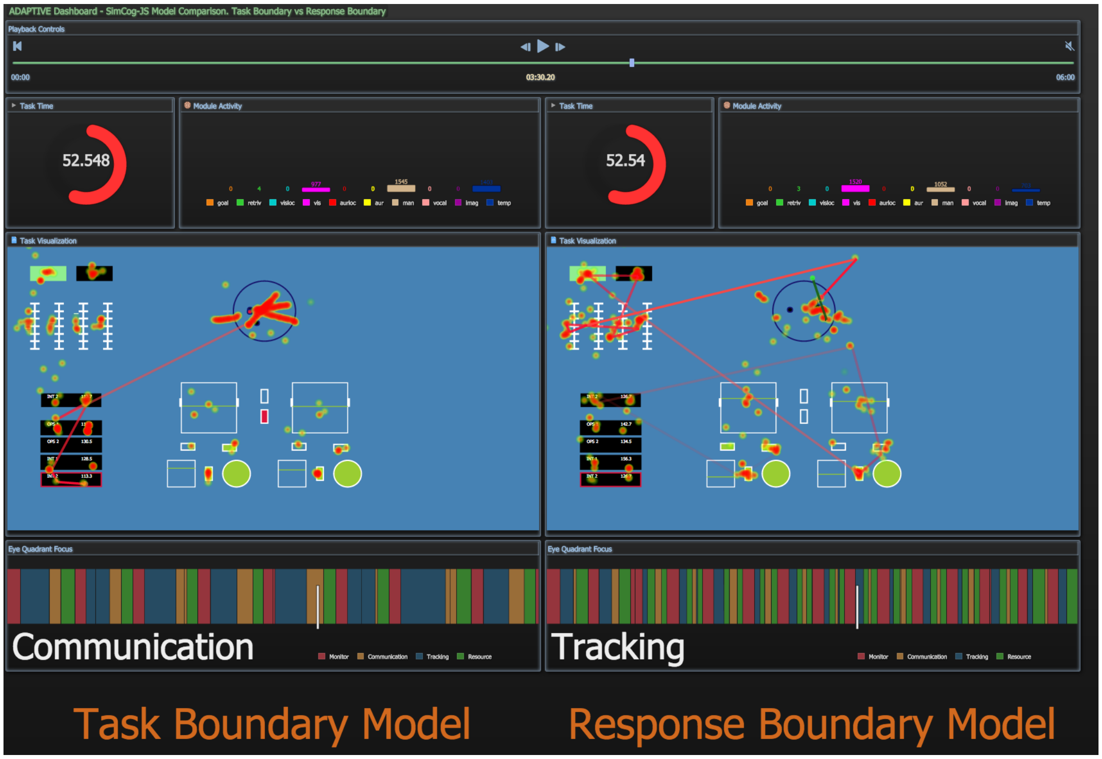


Fig. 3: Dashboard Visualization of simulated eye movement data from ACT-R models implementing different multitasking strategies.

the task. This allows the researcher to see the task from the model's perspective with the additional insight given by the mouse and eye locations.

Below the Task Visualizer is the Eye Quadrant Focus gauge, utilizing a scarf plot visualization of the duration spent in each area of interest (AOI) in the task. The AOIs correspond to the four quadrants. If a model switches tasks frequently, the scarf colors will change more often. During playback, the current quadrant in which the model is looking is named below the plot, and the white slider corresponds to the current time point in the model replay.

The Module Activity Gauge above the Task Visualizer gives a bar graph showing the level of activity in each of ACT-R's modules from the start of the task. The Module Activity Gauge can reveal when specific resources are a bottleneck across all goals. For example, if all task performance declines over an interval in which visual resource usage is maximized, then we can infer that the visual demands of the tasks are too high for successful completion of all goals. As shown in Figure 3, different strategies may utilize resources differently, resulting in different heights of the Module Activity Bars.

The Visualization Dashboard provides multiple ways to view model strategies in action, going well beyond a simple video playback of the model performing the task. Additional gauges might be added to illustrate the sequence of production rules firing as the task is performed, as well as illustrations of the model's motor activity, to further capture the underlying cognitive processes. Our Model Visualization Dashboard allows an analyst to see which tasks or actions are slowing performance for a given model. This empowers a researcher to draw conclusions about human performance or make modifications to a model or the task interface. Insights about internal processes lead to hypotheses about observable human behaviors which can be tested through both animation and human experiments.

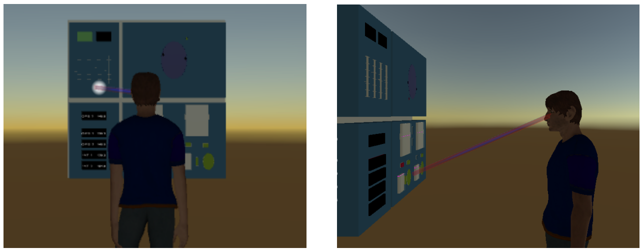
4 Virtual Embodiment of Model Eye Movements

Beyond examining the internal model dynamics, we visualize eye movement by embodying model activity in a virtual character operating in a 3-D environment. This gives us a concrete way to examine model predictions as they would play out in real people, and may allow an analyst to quickly observe a cognitive model's head and eye movements, determining unrealistic behavior at a glance. The 3-D environment allows the analyst to move the camera for different views of the agent in action. The virtual characters further allow us to examine data such as head movements from mobile eye trackers, complimenting the 2-D view of our analytics dashboard. Figures 1b and 4 illustrate this concept with a character multitasking. Typically, eye tracking software captures the foveal region, measured in pixels relative to the captured image. This is often interpreted as the location of visual attention and is similar to the information produced from ACT-R's vision module. Using a virtual character to display eye movement patterns can provide an analyst with a high fidelity visualization method to examine the realism of a model's saccade, fixation, or smooth pursuit behaviors. Again, in the present work, the data driving this virtual agent is derived from SIMCog-JS JSON messages.

The utility of virtually modeling attention in 3-D space has been shown in work eliciting joint attention between agents and people [4]. However, these methods only model the position of the eyes and do not model the scanpath. Furthermore, unlike the models used by Itti et al. [8], which implemented eye movements based on low-level salience maps to control virtual agents, we use cognitive architecture-based models. These architecture-based models capture higher-level cognitive processes supporting eye movements with parameters that can be tailored to emulate eye tracking data in specific tasks.

To calculate eye movements from either a model-generated or human-measured center of attention, we first must determine the world position of the fixation region in 3-D space. This is accomplished by assuming all of our data is modeled from a task displayed on a screen in front of the virtual character. By using this assumption, the area of attention for a given task can be converted into world coordinates by a simple transformation, treating the center of the screen in the virtual environment as the origin. We attach a halo or orb to the world position, allowing an analyst the ability to track the movement of a character's eyes, and essentially view in a virtual simulation the same information that is seen in Figure 1a.

Once the fixation location is determined in world coordinates, the rotation of the eye and head is then determined through cyclic coor-



(a) (b)

Fig. 4: Front view (a) and side view (b) showing the explicit fixation beams illustrating the gaze angle and focus of attention within the task environment. The beams are the red lines and the focus of attention is the white halo on the virtual task screen.

dinate descent inverse kinematics [19]. This provides an easy method to switch from one eye position to another. Inverse kinematics only provide the end rotations of a series of selected joints. This will cause the center gaze of the eye to snap to a given rotation. This appears as a jump, which is acceptable for ACT-R as it currently only produces saccadic eye movements. Yet, for human data or other models, an agent needs to perform other forms of eye movement, such as a smooth pursuit. For smooth pursuit eye movements, using cyclic coordinate descent can cause undesirable and unrealistic behavior. Therefore, we linearly interpolate between gaze points, which provides the movement trajectory between known points. We examined two other common interpolation methods, SLERP and SQUAD [5], but have found that these methods create strange eye movement patterns between points due to their inherent spherical nature. More complex interpolation techniques such as Gaussian Interpolation have not been examined using virtual human animations and are left for future work.

After examining the character operating on virtual tasks using our generated eye movement models, we noticed that it is difficult to watch their eye and head movement while simultaneously examining the areas of attention that eye tracking data and cognitive models produced. The size of the eyes, specifically of the pupil and iris, is quite small relative to the rest of the body and to the distance between the character and screen. Therefore, we also provide a method to exaggerate the eye movement and track the connection between the area of attention and center point of the eye socket. The connection is modeled as fixation beams, one for each eye, seen in Figure 4. This exaggeration can also be combined with the halo over the area of attention, which can be seen in Figure 4a.

To model the fixation beams connecting a character's eye to the area of attention, we construct a rod from the area of attention (transformed into world coordinates) to the center-point of the agent's eye. Pitch and yaw rotations are calculated between the two points, providing a new transformation matrix to the rod. Using a rod between eyes allows for more exaggerated movements of the eyes and head, creating more noticeable differences that might be lost using other visualization tools.

5 Visualizing Eye Movements Strategies

We illustrate the utility of this multi-pronged approach to visualizing formal model eye movements in the comparison of two candidate multitasking models. Using ACT-R [1] with EMMA [16], we developed multiple, hypothetical task-interleaving strategies. These models are *a priori* predictions of human behavior based on task analyses and constraints provided by the cognitive architecture.

One theory about how people multitask is that they interleave multiple "threads of thought", switching between these threads as cognitive processes and task demands allow [17]. Two simplified multitasking strategies along these lines are Task-Boundary Interleaving (TBI) and Response-Boundary Interleaving (RBI). The TBI strategy only shifts visual attention to another task when either no response is needed to the currently attended task or after all key presses to resolve that task have been initiated. That is, only when one task is completed will the system switch to a different task. The RBI strategy attempts to maximize the overlap of visual attention shifts and keyboard responses by shifting visual attention to another task if consecutive key presses are needed for a response to the currently attended task *and* shifts of visual attention are not required to determine the correct key response. This is based on the assumption that if people only need to finish a determined sequence of motor actions to complete the current task, then visual attention can be reallocated to the next task while the response is completed. The communication task in the lower left corner is the only task in our environment that meets this criterion of repeated key presses (multiple presses to change the channel value). So by definition of the RBI, the model (and people using this strategy) should switch attention to the next goal while completing a response to the communication task.

Each strategy elicits a different set of scanpaths and visual fixation patterns. Based on our strategy definitions, we predict that TBI will result in the model spending more time fixated on the lower left quadrant than RBI, and that RBI may scan between the different quadrants more frequently than TBI, reflecting more shifts in covert attention. The scarf plots in Figure 3 illustrate exactly this predicted pattern, with RBI switching AOIs more frequently than the TBI model.

Animation provides an overview and high-level understanding of the differences between the embodied eye movements produced by the two models. Very soon into the playback, it becomes clear that the TBI model fixates on the continuous object tracking task (upper right quadrant, reflected in the darker fixation heat map in that task for TBI in Figure 3) but fixates on the communication task less frequently, which is contrary to our original predictions. This might be a result of the continuously moving target shifting the tracking goal. Consequently, the model acts like the task is not complete, and it continues to execute the tracking while ignoring activity in the other tasks. On the other hand, the RBI model shows more saccades between quadrants, but often appears to overshoot the target locations across quadrants and has to make extra corrective movements. The Module Activity interface reveals additional differences in the underlying processes brought to bear on the task by the two models. It shows the TBI model recruits more temporal planning (blue bar) and motor (light orange bar) resources, while the RBI model utilizes more visual resources (bright pink bar). This information is only observable in the Module Activity gauge.

6 Conclusion

Visual analytics presents a complex task environment for human decision making, and cognitive modelers have begun developing models for many aspects of visual analytics, including network visualization interpretation [18] and graph reading and comprehension [13, 15]. We show that multiple visualization approaches can both capture the model predictions for eye movements and elucidate key underlying cognitive mechanisms supporting those eye movements. Extending the Dashboard to include human eye movement data would complete the analytics process. First, by comparing metrics on movements (total saccades, fixation durations, etc.) between the model and human behavior, we can test a model's ability to capture human eye tracking performance. Then the Model Visualization Dashboard provides the tools for inferring the mechanisms underlying the observed human performance. The animation can further show the model eye movements next to the actual eye movements, validating or refuting the predictions made by the model. Thus, visualizing formal cognitive models provides the capability to make the complete set of desired inferences from eye tracking data about the efficacy of visual information processing.

Acknowledgments

This work was supported by AFOSR LRIR to L.M.B. Distribution A: Approved for public release; distribution unlimited. 88ABW Cleared 08/26/2015; 88ABW-2015-4021.

References

[1] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, Oct. 2004.

[2] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data driven documents. *IEEE Transactions on Visualization and Computer Graphics*, pages 2301–2309, 2011.

[3] J. R. Busemeyer and A. Diederich. *Cognitive modeling*. Sage, 2010.

[4] M. Courgeon, G. Rautureau, J.-C. Martin, and O. Grynszpan. Joint attention simulation using eye-tracking and virtual humans. *Affective Computing, IEEE Transactions on*, 5(3):238–250, July 2014.

[5] E. B. Dam, M. Koch, and M. Lillholm. Quaternions, interpolation and animation. Technical Report DIKU-TR-98/5, University of Copenhagen, Universitetsparken 1, DK-2100 Kbh, Denmark, 1998.

[6] T. Halverson and A. J. Hornof. A Computational Model of "Active Vision" for Visual Search in Human-Computer Interaction. *Human-Computer Interaction*, 26(4):285–314, 2011.

[7] T. Halverson, B. Reynolds, and L. M. Blaha. SIMCog-JS: Simplified interfacing for modeling cognition - javascript. In *Proceedings of the International Conference on Cognitive Modeling*, Groningen, The Netherlands, April 2015.

[8] L. Itti, N. Dhavale, and F. Pighin. Photorealistic attention-based gaze animation. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 521–524, July 2006.

[9] D. E. Kieras and D. E. Meyer. An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12(4):391–438, 1997.

[10] J. E. Laird. *The Soar Cognitive Architecture*. MIT Press, 2012.

[11] E. Nyamsuren and N. A. Taatgen. Pre-attentive and Attentive Vision Module. In *Proceedings of the 2015 International Conference on Cognitive Modeling*, pages 211–216, 2012.

[12] R. E. Patterson, L. M. Blaha, G. G. Grinstein, K. K. Liggett, D. E. Kaveney, K. C. Sheldon, P. R. Havig, and J. A. Moore. A human cognition framework for information visualization. *Computers & Graphics*, 42:42–58, 2014.

[13] D. Peebles. A cognitive architecture-based model of graph comprehension. *Nele Rußwinkel— Uwe Drewitz— Hedderik van Rijn (eds.)*, page 37, 2012.

[14] D. Peebles. Strategy and pattern recognition in expert comprehension of 2 x 2 interaction graphs. *Cognitive Systems Research*, 24:43–51, 2013.

[15] D. Peebles and P. C.-H. Cheng. Modeling the effect of task and graphical representation on response latency in a graph reading task. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 45(1):28–46, 2003.

[16] D. D. Salvucci. An Integrated Model of Eye Movements and Visual Encoding. *Cognitive Systems Research*, 1(4):201–220, 2001.

[17] D. D. Salvucci and N. A. Taatgen. Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review*, 115(1):101, 2008.

[18] M. Schoelles and W. D. Gray. Speculations on model tracing for visual analytics. In R. L. West and T. C. Stewart, editors, *Proceedings of the 12th International Conference on Cognitive Modeling*, 2013.

[19] L.-C. Wang and C. Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *Robotics and Automation, IEEE Transactions on*, 7(4):489–499, Aug 1991.

[20] J. M. Wolfe. Guided Search 4.0. In W. D. Gray, editor, *Integrated Models of Cognitive Systems*, pages 99–119. Oxford, New York, 2007.

[21] Y. Zhang and A. J. Hornof. Understanding multitasking through parallelized strategy exploration and individualized cognitive modeling. In *Conference on Human Factors in Computing Systems*, pages 3885–3894, New York, New York, USA, Apr. 2014. ACM.