Title: NAV: A Tool for Producing Presentation-Quality Animations of Graphical Cognitive Model Dynamics

Authors: Trent Kriete, Matthew House, Bobby Bodenheimer, and David C. Noelle
Department of Electrical Engineering & Computer Science Vanderbilt University

Abstract: Computational models of cognition often exhibit complex dynamics that are difficult to discern without the use of visualization tools. Current visualization tools often provide insight only to the modeling expert, however, and they provide limited functionality for communicating model dynamics to the non-expert, as is needed during scientific presentations and in educational settings. We present NAV: an easy-to- use software tool which interactively transforms the output of cogni- tive modeling simulators into presentation-quality animations of model performance.

Summary:

Computational models of human cognition often exhibit rich and complex dynamics. This richness can be crucial for capturing the nuances of human performance, and computational simulation of cognitive models is often relied upon exactly because the dynamics of such models resist more analytic approaches. Complex dynamic behavior can hinder the development of a deep understanding of a cognitive model, however, masking essential mechanisms behind a flurry of activity. To aid researchers in understanding their models, simulation software packages typically provide tools for the visualization of model dynamics. Simulators, ranging from psychological modeling environments like ACT-R 5.0 (Anderson and Lebiere, 1998) to computational neuroscience packages like GENESIS (Bower and Beeman, 1998) and connectionist simulators like PDP++ (O'Reilly, 2004), provide a wide range of methods for monitoring model performance as it unfolds over time. These visualization tools are almost universally designed for the modeling expert, however, providing large arrays of data to help enrich the understanding of those who are already intimately familiar with the cognitve model being examined. Rarely are these displays of direct utility in communicating model dynamics to the non-expert, as is often important in the context of scientific presentations and in educational settings. Instead, researchers often resort to hand-crafted cartoons of their models, which may be easily embedded in presentation slides or web pages. Such cartoons of model performance have a number of drawbacks, not the least of which is that they can be tedious to prepare. More importantly, they hide actual model behavior from the critical audience, providing the presenter's interpretation of the model in place of actual simulation results. To address the need for presentation-quality illustrations of actual cognitive model dynamics, we have produced an easy-to-use software tool for interactively transforming the output of cognitive modeling simulators into clear and informative animations of model performance. This tool is called the Node Activity Visualizer, or NAV.

NAV was originally intended for use with connectionist cognitive models, and it continues to focus on this class of models, but it may also be used to illustrate any model that both possesses a graphical structure and relies on numerical values associated with graph nodes (e.g., node activation levels) in order to function. For example, spreading activation networks, like those used in the memory mechanisms of ACT-R, can be easily accommodated. Computational neuroscience models also fit within this framework. While NAV focuses on graphical cognitive models, it is not, itself, a model simulator. Thus, it introduces no limitations whatsoever on the dynamics of node activity levels.

To make use of NAV, a model simulation must first be executed using simulation software of the user's choice, recording "snapshots" of important node activity values into a file as the simulation runs. Once such a record file is in hand, the NAV program may be used to craft how the simulation results are to be displayed. NAV presents the user with an intuitive "drawing program"-like interface, allowing for the creation and placement of nodes, groups of nodes called "layers", and arrows that are associated with connections between nodes or layers. Once the graphical model has been constructed, NAV provides the user with an easy way to associate nodes in the display with entries in the simulation record file. Many options exist for the graphical display of node activation levels, ranging from node intensity or color to node size or orientation. For example, NAV provides tools for generating custom color scales that may be used to reflect activity levels. With the mapping between the simulation record file and the graphical model display

complete, NAV may then be used to generate animations of simulation results. The speed of these animations may be varied, and they may be inspected and edited on a frame-by-frame basis. Most importantly, the animations may be recorded in standard movie file formats (e.g., MPEG, animated GIF), allowing them to be readily used in presentation slides or on web pages.

One important feature of NAV is its ability to display, in a time-dependent fashion, objects other than a set of connected nodes, allowing for multiple views on the cognitive model (Wejchert and Tesauro, 1990). The user may also include textual or graphical "sprites" to label or explain features of the model over time. For example, a model of a picture naming task might contain nodes whose activities correspond to visual features of a picture being viewed and nodes whose activities correspond to phonetic features of names produced by the model. To make the modeled experimental situation clear to the audience, a graphical sprite may be used to display an actual picture corresponding to the visual input provided to the model, and a textual sprite may be used to translate the model's output into a printed word. We believe that supportive annotations of this kind, which are not commonly provided by simulator-native visualization tools, are critical for the production of presentation-quality model animations.

In addition to being simulator independent, NAV has strong cross-platform support. Built using the Qt user interface tools (Blanchette and Summerfield, 2004), the open source NAV software runs under Windows, Mac OS X, Linux, and Unix.

While other commercial software tools exist for the general production of animations (e.g., Macromedia Flash (deHaan, 2004)), NAV relieves the modeler of the burden of learning such general tools and provides specific support for cognitive model illustrations. Ease-of-use and ease-of-learning have been paramount design concerns. Indeed, a preliminary user evaluation study, involving an early prototype of the NAV system, was conducted with the help of 10 novices in connectionist modeling and 3 experts. These evaluators were asked to construct a specified animation, given a simulation record file. They were then surveyed concerning their experience with NAV, and design feedback was solicited. In general, ease-of-use ratings were high (averaging 4.59 on a 5 point Likert scale), though some problems with graphical object selection were identified. Members of the expert group called for a larger range of display options, such as the dynamic graphical display of numerical values associated with connections rather than nodes (e.g., connection weights) and the ability to specify non-linear relationships between activation levels and node color scales (in order to highlight salient activation ranges). By making use of evaluative feedback, NAV continues to be improved and refined.

References

Anderson, J. R. and Lebiere, C. (1998). *The Atomic Components of Thought*. Lawrence Erlbaum, Mahway, New Jersey.

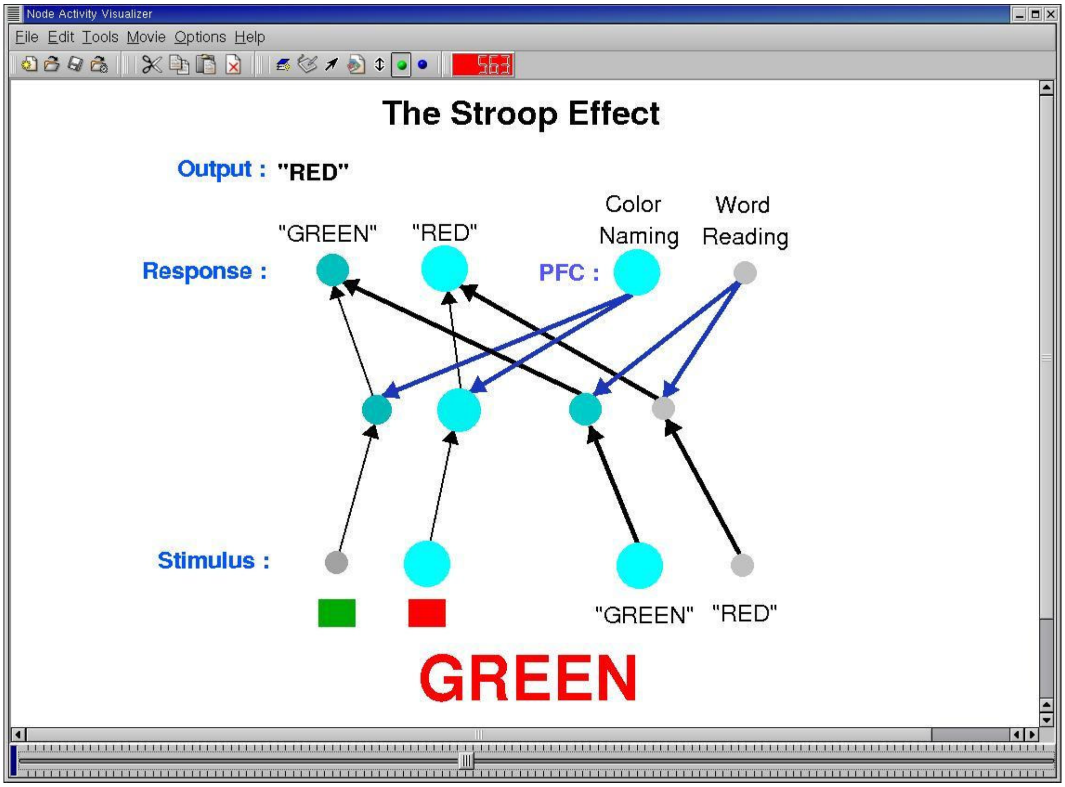
Blanchette, J. and Summerfield, M. (2004). *C++ GUI Programming with Qt 3*. Prentice Hall, Englewood Cliffs, New Jersey.

Bower, J. M. and Beeman, D. (1998). *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*. TELOS, New York, 2nd edition.

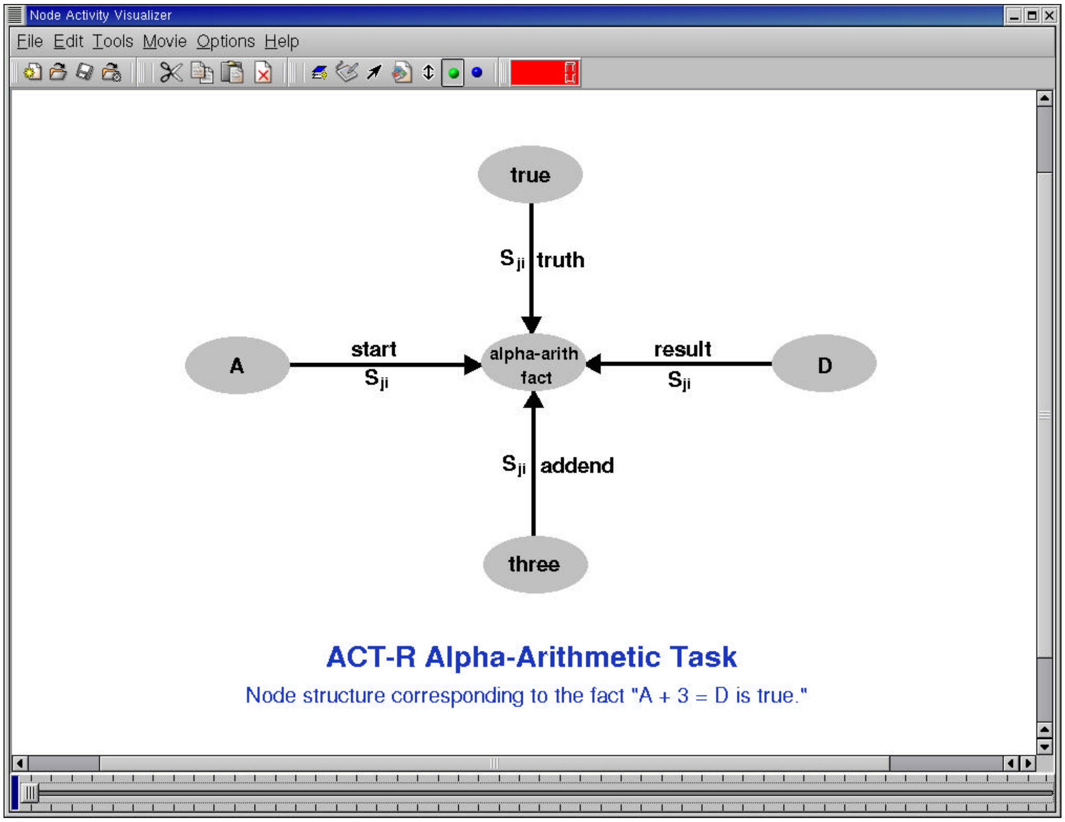
deHaan, J. (2004). *Macromedia Flash MX 2004: Training from the Source*. Macromedia Press, San Francisco.

O'Reilly, R. C. (2004). The PDP++ software home page. http://psych.colorado.edu/˜oreilly/PDP++/PDP++.html.

Wejchert, J. and Tesauro, G. (1990). Neural network visualization. In Touretzky, D. S., editor, *Advances In Neural Information Processing Systems 2*, pages 465–472, Denver. Morgan Kaufmann.



The Stroop Effect



ACT-R Alpha-Arithmetic Task

Node structure corresponding to the fact "A + 3 = D is true."