Behavior Research Methods
2005, 37 (2), 335-339

NAV: A tool for producing presentation quality animations of graphical cognitive model dynamics

TRENT KRIETE, MATTHEW HOUSE, BOBBY BODENHEIMER, and DAVID C. NOELLE
Vanderbilt University, Nashville, Tennessee

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

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 such as ACT–R (Anderson & Lebiere, 1998) to computational neuroscience packages such as GENESIS (Bower & Beeman, 1998) and connectionist simulators such as 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 cognitive model being examined. Rarely are these displays of direct utility in communicating model dynamics to the nonexpert, as is often important in the context of scientific presentations and in educational settings. Instead, researchers often resort to handcrafted 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, including the fact that they can be tedious to prepare. More important, 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.

The authors thank the members of the Vanderbilt University Computational Cognitive Neuroscience Laboratory for their helpful comments. Please send correspondence to D. C. Noelle, Department of Electrical Engineering and Computer Science, Vanderbilt University, Box 1679, Station B, Nashville, TN 37235 (e-mail: david.noelle@vanderbilt.edu).

Features of NAV

NAV was originally intended for use with connectionist cognitive models, and this class of models continues to be its focus; 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, such as those used in the memory mechanisms of ACT–R (Anderson & Lebiere, 1998), can be easily accommodated, as is shown in Figure 1. Computational neuroscience models also fit within this framework, with the node activation levels corresponding to relevant biological variables, such as neural firing rates (Dayan & Abbot, 2001). Although the focus of NAV is on graphical cognitive models, it is not, itself, a cognitive model simulator. Thus, it introduces no limitations whatsoever on the dynamics of node activity levels that may be displayed.

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. These simulation record files must be in a plain text format and must form a matrix of activation level values, with one row per time step of the simulation and one column for each node in the cognitive model. Data files of this kind are typically very easy to generate from within most cognitive modeling simulation systems.[1] 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

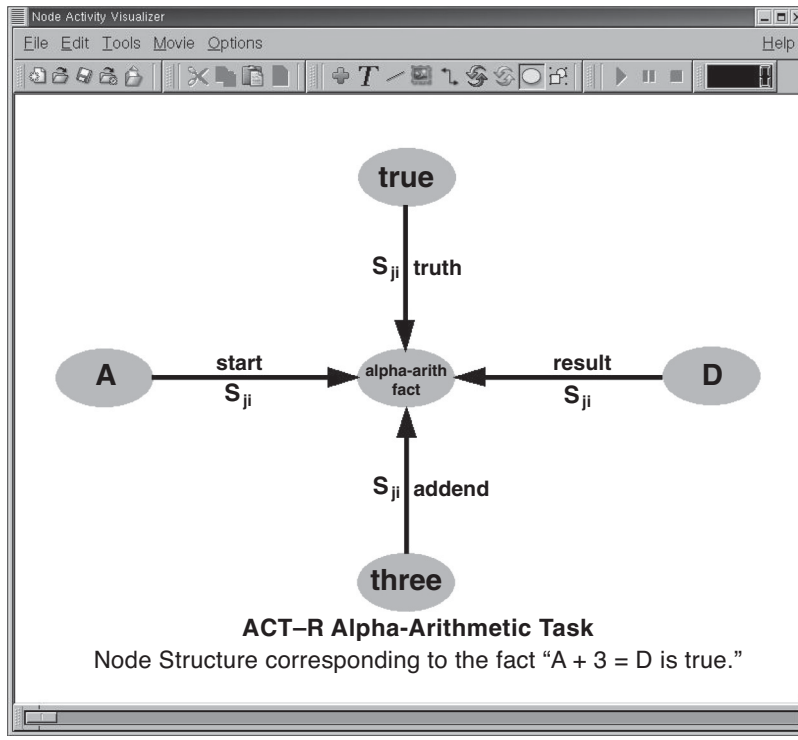

Figure 1. The main NAV window displaying an animation of a component of the spreading-activation–based memory network of an ACT–R model.

of nodes called *layers*, and arrows, or *links* 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, using a sliding-bar control at the bottom of the main NAV window. Most important, the animations may be recorded in standard movie file formats (e.g., MPEG), 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 & 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. Another example is shown in Figure 2, which displays an animation of a connectionist model of cognitive control in the Stroop task (Cohen & Servan-Schreiber, 1992), implemented using the Leabra modeling framework (O'Reilly & Munakata, 2000). Note the display of the current stimulus item, shown near the bottom of the frame, as well as a textual label for the current response, shown in the upper-left corner. In this animation, both of these textual labels change dynamically with the activation levels of the nodes. 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, since they quickly and intuitively associate model dynamics with the behavioral phenomena being modeled.

The link arrows displayed in a NAV animation often reflect weighted connections in an underlying graphical cognitive model. In some models, such as connectionist learning models, the weights associated with these connections are dynamic quantities as well, worthy of display over time. NAV supports the animation of link changes over time in much the same way that it supports the display of node activation dynamics. A textual data file containing connection weight values over time may be pro-

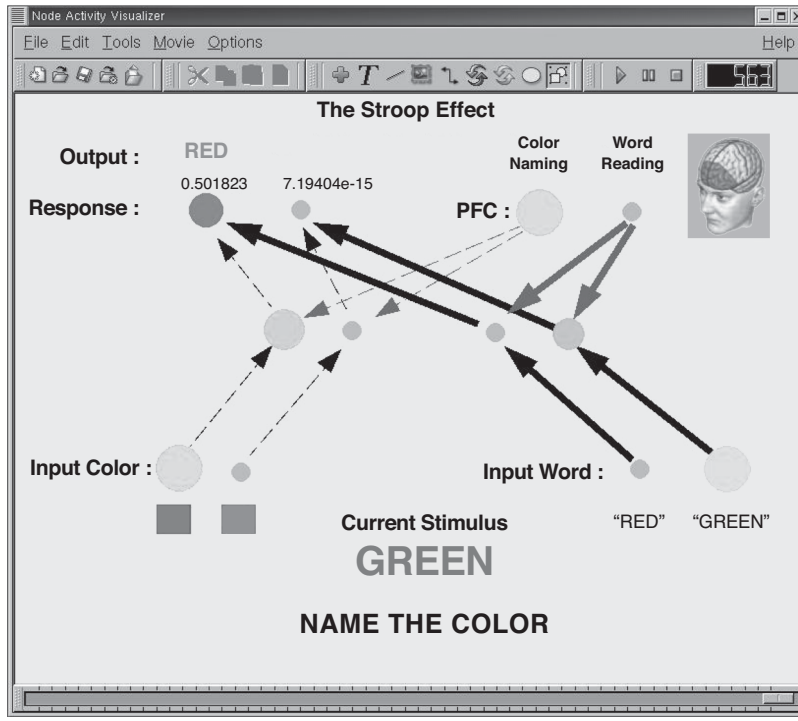

Figure 2. The main NAV window displaying an animation of a model of the Stroop task. Node activity is reflected in both the size and the color of the nodes, and textual labels are dynamic.

duced from within the user's simulation software of choice, and the weight values in this data file may be associated with the links in the NAV display. Currently, arrow thickness and color may track dynamic weight values. An additional feature allows a link property to change only when the corresponding weight value passes a given threshold, allowing, for example, a link to be green as long as the corresponding weight is excitatory (i.e., positive) and changing to red when the weight becomes inhibitory (i.e., negative). Although the use of dynamic link weights is strictly optional in NAV, the inclusion of these tools allows NAV to be used to illustrate both model activation dynamics and the dynamics associated with the learning of connection weights.

NAV has been designed to be easy to learn and use. It supports the flexible display of model dynamics, and it is not dependent on any particular modeling framework or simulation software. To our knowledge, this separation of the visualization task into an isolated software tool is unique to NAV, with other visualization tools being tightly integrated with specific modeling frameworks (e.g., Bower & Beeman, 1998; Finnie, 2004; Mathworks, 2004; O'Reilly, 2004; StatSoft, 2004; Streeter, Ward, & Alvarez, 2001; Zell et al., 1993). NAV also has strong cross-platform support. Written in C++, using the Qt user interface tools (Blanchette & Summerfield, 2004), the open source NAV software runs under Windows, Mac OS X, Linux, and Unix.

Preliminary User Evaluations

Although 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. In order to provide an initial assessment of the facility with which users could learn and manipulate the NAV interface, two evaluation studies were conducted. The first of these studies focused on users who had little experience with computational cognitive modeling, and the second sought deeper insights from modeling experts. In both cases, the participants were introduced to NAV through the use of a printed tutorial document and were then asked to construct an animation that met a given set of specifications. Following this experience, the participants were surveyed concerning their views on the strengths and weaknesses of the NAV software.

Novice user evaluation. In order to assess the ease with which the NAV interface could be learned, the first study involved 10 participants who were regular computer users but lacked any background in computational models of cognitive processes. This group consisted of 6 females and 4 males with a mean age of 25.7 years ($SD = 3.0$), all graduate students at Vanderbilt University. When asked to self-rate their own computer proficiency and cognitive modeling experience on a 5-point

Table 1
Novice User Evaluation Results

Topic Area	No. of Ratings	Average Score	*SE*
Overall reaction to the system	3	4.80	0.13
Creation of objects	2	4.80	0.13
Movement of objects	2	5.00	0.00
Modifying the properties of objects	2	4.80	0.13
Associating activation data with nodes	2	4.60	0.22
File management	2	4.60	0.70
Building a movie	2	4.30	0.47
Scrolling through movie frames	2	5.00	0.00
Learning to use the application	2	4.50	0.27

Note—Each participant provided 19 ratings, broken down by topic area, as shown. Each rating was on a 5-point Likert scale, ranging from *difficult* (1) to *easy* (5).

Likert scale, ranging from *novice* (1) to *expert* (5), these participants considered themselves moderately strong computer users, with a mean rating of 3.40 ($SD = 0.70$), and very weak modelers, with a mean rating of 1.20 ($SD = 0.42$). After completing a tutorial on NAV and an exercise in which a new animation was constructed, these participants were asked to produce 19 ratings of the ease of learning, ease of use, and general experience of using NAV. Each rating was on a 5-point Likert scale. These questions were segregated into nine topic areas (Table 1).

For each of these topic areas (except the first), the first rating involved difficulty of use, whereas the second rating involved the level of frustration or satisfaction experienced by the user. The mean results for question pairs tracked each other closely, so we will report only the more direct ratings involving difficulty/ease-of-use. These results are reported in Table 1. Note that NAV was rated very highly in all categories for ease of use and ease of learning.

Expert user evaluation. In order to obtain guidance for the further development of NAV, 5 users with substantial cognitive modeling experience were surveyed in a more open-ended manner than that used with the novice group. These expert participants were all graduate students who had completed at least one graduate level course on computational cognitive modeling or computational neuroscience. After completing the NAV tutorial and animation construction exercise, a questionnaire consisting of eight fairly general questions was administered to each participant. Responses to these questions indicated that NAV met its ease-of-use goals, but the experts also suggested some opportunities for improvement.

Of the 5 experts, all found NAV to be a useful tool for visualizing and presenting the activation dynamics of computational cognitive models. All the participants rated the software as extremely easy to learn and very easy to use. When asked to list some of the benefits of NAV, 3 of the 5 experts identified the ability to add dynamic images and textual labels as a valuable feature for providing deep insights into model dynamics in a short period of time. In comparison with other model visualization tools, including cognitive model simulation software, the experts volunteered a number of unique beneficial features of NAV. Two of the 5 experts listed NAV's support for dynamic images and textual labels as an important discriminating feature. Two experts asserted that other tools tend to be more cumbersome than NAV, and 3 called attention to the fact that other tools tend to impose undesirable limits on the ways in which model architectures and dynamics can be displayed.

The surveyed experts also made a number of suggestions for the improvement of NAV. Two of the 5 requested support for embedded dynamic plots and graphs—a feature that is available in some model simulation software. Two experts also requested greater support for "undoing" interface actions and for the construction of default value templates for particular kinds of model animations. Other suggestions included the incorporation of on-line help messages, improvements to the mechanisms for moving through movie frames and selecting individual nodes, and a means to automatically save when an application is closed.

Although all 5 experts indicated that they would consider using the current version of NAV to produce animations for their presentations, 2 of the 5 experts indicated that the visualization options provided by existing cognitive model simulation software was still to be preferred when a deeper analysis of model dynamics was required. This assessment fits well with NAV's design goals. As was discussed in the introduction, the visualization capabilities of cognitive model simulators are typically designed to assist in expert model analysis. In comparison, NAV is intended to assist in the construction of animations that are useful for communicating model dynamics to those who are unfamiliar with the specific model in question, leaving more complex analysis, as well as actual model simulations, to full-fledged cognitive model simulation software.

Evaluation summary. All of the NAV users surveyed were able to learn the tool and construct an animation meeting precise specifications within a period of about 45 min. Those surveyed universally found NAV to be easy to learn and easy to use. Although the modeling experts pointed out some ways in which NAV could be improved, they also identified a number of strengths of NAV, in comparison with the visualization tools that are embedded in current cognitive modeling systems. The experts unanimously endorsed NAV as a useful tool for the generation of presentation quality animations illustrating cognitive model dynamics.

Conclusion

We have presented NAV, the Node Activity Visualizer. NAV is a multiplatform tool for the generation of presentation quality animations illustrating the dynamics of graphical cognitive models. User studies have shown NAV to be easily learned and easy to use and to contain capabilities not found in other common products for model visualization. NAV has been successfully used to

embed cognitive model animations in professional presentations.

NAV is an open source software project. The current release, including executable binaries, source code, and documentation, may be downloaded from the NAV Web site: www.vuse.vanderbilt.edu/~noelledc/resources/NAV/. NAV is an ongoing project. Current development efforts include the implementation of the features suggested by the modeling experts we surveyed, as well as the inclusion of three-dimensional display objects. Feedback concerning the use of this software is welcome and should be sent to nav-devel@ccnl.vuse.vanderbilt.edu.

REFERENCES

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Blanchette, J., & Summerfield, M. (2004). *C++ GUI programming with Qt 3*. Englewood Cliffs, NJ: Prentice-Hall.

Bower, J. M., & Beeman, D. (1998). *The book of GENESIS: Exploring realistic neural models with the GEneral NEural SImulation System* (2nd ed.). New York: TELOS.

Cohen, J. D., & Servan-Schreiber, D. (1992). Context, cortex, and dopamine: A connectionist approach to behavior and biology in schizophrenia. *Psychological Review*, **99**, 45-77.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: Computational and mathematical modeling of neural systems*. Cambridge, MA: MIT Press.

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

Finnie, C. (2004). *The neural viewer home page*. Available at http://www.btinternet.com/~cfinnie/.

Mathworks (2004). *Neural network toolbox home page*. Available at http://www.mathworks.com/products/neuralnet/.

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

O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. Cambridge, MA: MIT Press.

StatSoft (2004). *Statistica neural networks home page*. Available at http://www.statsoft.com/products/stat_nn.html.

Streeter, M. J., Ward, M. O., & Alvarez, S. A. (2001). NVIS: An interactive visualization tool for neural networks. In R. F. Erbacher, P. C. Chen, J. C. Roberts, C. M. Wittenbrink, & M. Groehn (Eds.), *Visual data exploration and analysis VIII: Proceedings of SPIE* (pp. 234-241). San Jose: Society for Optical Engineering.

Wejchert, J., & Tesauro, G. (1990). Neural network visualization. In D. S. Touretzky (Ed.), *Advances in neural information processing systems* 2 (pp. 465-472). Denver: Morgan Kaufmann.

Zell, A., Hübner, R., Korb, T., Mache, N., Mamier G., Schmalzl, M., Sommer, T., & Voget, M. (1993). SNNS: An efficient simulator for neural nets. In H. D. Schwetman, J. C. Walrand, K. K. Bagchi, & D. DeGroot (Eds.), *MASCOTS '93: Proceedings of the international workshop on modeling, analysis and simulation of computer and telecommunication systems*. San Diego: Society for Computer Simulation.

NOTE

1. For example, an appropriately configured TextLog object in PDP++ (O'Reilly, 2004) will automatically generate an activity record file of this form.

(Manuscript received November 27, 2004;
revision accepted for publication May 10, 2005.)