Wayne Gray

Wayne Gray is a researcher in the fields of computational cognitive modeling, cognitive neuroscience, interactive behavior, cognitive task analysis, cognitive workload, and human error. Since earning his Ph.D. from UC Berkeley he has worked for government and industry research laboratories, as well as universities. He is currently a Professor of Cognitive Science at Rensselaer Polytechnic Institute. Wayne is a Fellow of the Human Factors & Ergonomics Society (HFES), the Cognitive Science Society, and the American Psychological Association (APA). In 2008, APA awarded him the Franklin V. Taylor Award for Outstanding Contributions in the Field of Applied Experimental & Engineering Psychology. He is a past Chair of the Cognitive Science Society and the founding Chair of the Human Performance Modeling technical group of HFES. At present he is the Executive Editor for the Cognitive Science Society's first new journal in 30 years, *Topics in Cognitive Science (topiCS).*

~~~~~~

The Shape of Things to Come: An Emerging Constellation of Interconnected Tools for Developing the Right Cognitive Model at the Right Scale

There are at least three major problems with the current state of cognitive modeling. First, modeling is *too hard and takes too long*. There is a paucity of tools that allow you to set up a cognitive model at the same high level of abstraction that tools such as SPSS™ or SAS™ allow you to set up a complex statistical model for data analysis. Rather, most modeling formalisms require some computer science or mathematics training and typically each new model takes just as long to build as the last model. Second, cognitive modeling seems to engender the "*to a man with a hammer, everything looks like a nail"* syndrome. Once a modeling technique is mastered, too many people try to apply it to every situation whether or not it is the best tool for the current task. Third is *scale inflexibility* and a concomitant *lack of interconnectedness*. Modeling with any given technique locks you into a certain level of analysis. Popping up or down a level of analysis, say from a model of *reading with understanding* to a model of the *perception, eye movements, and memory involved in reading* requires abandoning one model and building another.

I will describe the *shape of things* to come by introducing two modeling tools and the *emerging constellation* that has resulted from their interconnectedness with each other and with the ACT-R (Anderson, 2007) architecture of cognition. The two tools, CogTool (John, Prevas, Salvucci, & Koedinger, 2004) and the *Stochastic Analysis Network Laboratory for Cognitive Modeling* (SANLab-CM, Patton & Gray, 2009) do not require the average user to have a background in computer science or mathematics. In contrast, modeling in ACT-R requires learning a specialized programming language. Although prior computer science or mathematics background is not strictly necessary, few modelers get very far without some training in these disciplines.

CogTool allows the modeler to create *Keystroke Level Models* (KLM, Card, Moran, & Newell, 1983) by demonstrating a sequence of moves in a storyboarded version of the task environment. The KLMs predict the performance times of expert users. It makes these predictions by creating and running a simple *ACT-R* model that uses default ACT-R parameters and the constraints imposed by the task environment.

SANLab-CM is the first tool designed to facilitate the development, manipulation, and comparison of activity network models for cognitive modeling. Examples of this type of modeling include CPM-GOMS (Gray, John, & Atwood, 1993; John, 1990) and the critical-path scheduling of mental processes (Schweickert, 1980; Schweickert, Fisher, & Proctor, 2003). Additionally, SANLab-CM is the first modeling tool that we know of specifically designed to explore the influence of stochasticity on cognitive outcomes. Whereas past CPM-GOMS models enabled the modeler to assign a fixed time to each operation, SANLab-CM enables the modeler to assign means and distributions of times. (Different types of operations may be assigned different default mean times and/or different default distributions. This is a
~~~~~~

feature, not a limit, as it is possible to assign times and distributions to individual operations.) When the resulting model is run, multiple *critical paths* are produced along with predictions of expected minimum and maximum response times. The utility of SANLab-CM will be demonstrated by comparing SANLab-CM models of Telephone Operator-Customer-Workstation interactions to the nonstochastic models of the same task built by Gray and John (Gray, et al., 1993).

CogTool, SANLab-CM, and ACT-R are interconnected. Whereas SANLab-CM can be used alone, it is possible to build a SANLab-CM model by importing the trace produced by running an ACT-R model. Once imported, SANLab-CM can be used to quickly explore the influence of different distributions (e.g., Gaussian versus gamma), different parameters of the distribution, or (to a limited degree) different designs of the task environment.

Likewise, SANLab-CM can be used in conjunction with CogTool. Running CogTool's simplified ACT-R model produces the KLM's predicted expert performance times. The trace produced by that model can be imported into SANLab-CM. Once in SANLab-CM it can be inspected, edited, manipulated, assigned various distributions, and run to inspect the various critical paths that would be produced by the stochastic activity network.

This is the shape of things to come. CogTool and SANLab-CM require no mathematical or computer science expertise to produce a model. Indeed, whereas SANLab-CM requires cognitive science expertise, CogTool does not. Each of these three tools, CogTool, SANLab-CM, and ACT-R can be used to develop models at different temporal scales so that a modeler who starts with one type of model can quickly develop another. The interconnectedness of SANLab-CM enables an emerging constellation of tools for developing the right model at the right scale.

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

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world performance. *Human-Computer Interaction, 8*(3), 237–309.

John, B. E. (1990). *Extensions of GOMS Analyses to Expert Performance Requiring Perception of Dynamic Visual and Auditory Information.* Paper presented at the ACM CHI'90 Conference on Human Factors in Computing Systems, New York.

John, B. E., Prevas, K., Salvucci, D. D., & Koedinger, K. (2004). Predictive human performance modeling made easy *ACM CHI'2004 Conference on Human Factors in Computing Systems* (pp. 455-462). New York: ACM Press.

Patton, E. W., & Gray, W. D. (2009). SANLab-CM: A tool for incorporating stochastic operatons into activity network modeling. *submitted for publication*.

Schweickert, R. (1980). Critical-Path Scheduling Of Mental Processes In A Dual Task. *Science, 209*(4457), 704-706.

Schweickert, R., Fisher, D. L., & Proctor, R. W. (2003). Steps toward building mathematical and computer models from cognitive task analyses. *Human Factors, 45*(1), 77–103.