Symposium on Human Performance Modeling

Wayne D. Gray (Organizer)
Rensselaer Polytechnic Institute

David Kaber and Guk-Ho Gil and Sang-Hwan Kim
North Carolina State University

Naom Ben-Asher and Joachim Meyer
Ben-Gurion University

Shi Cao and Yili Liu
University of Michigan

Cleotilde Gonzalez (Panel Member)
Carnegie Mellon University

Glenn Gunzelmann and Kevin Gluck (Panel Members)
U. S. Air Force Research Laboratory

This symposium is co-sponsored by the Human Performance Modeling Technical Group (HPM-TG) of the Human Factors & Ergonomics Society. Three Research Talks and a Panel Discussion were presented. Each talk used a different style of cognitive modeling and addressed a different problem of interest to the human factors community. For the Panel Discussion, three additional members of the HPM-TG joined with our three speakers in a round table discussion of the similarities and differences between cognitive modeling in applied versus basic science.

Keywords: human performance modeling, human factors

Introduction

The Human Performance Modeling Technical Group (HPM-TG) of the Human Factors and Ergonomics Society (HFES) is proud to sponsor its first Symposium at ICCM. Although much of the work presented at ICCM focuses on basic research, it is clear that ICCM recognizes the value of applied cognitive modeling. Indeed, a long tradition at ICCM is the *Siegel-Wolf Award for Best Applied Paper*. Not only does this reward reflect the value of applied modeling, but it is named after two men who were HFES members in the early days of human performance modeling.

The spirit of Siegel and Wolf lives on in the HFES and ICCM communities even though both largely go their separate ways. This symposium is intended as the first in a long term exchange that we hope will enrich ICCM and the HPM-TG. Those desiring a snapshot of the recent history and current status of cognitive modeling in human factors should see Gray (2008a, 2008b). Those interested in details from the early days of human performance modeling should see two excellent papers by Pew (2008, 2007).

Technical Talks

An Accessible Cognitive Modeling Tool for Evaluation of Pilot-Automation Interaction - Kaber, Gil, & Kim

One of the main limitations of existing approaches to complex human-in-the-loop system design is the requirement for empirical data as a basis for alternative design selection. Experimental studies can be time consuming and costly. In addition, design decisions are often based on collections of design guidelines with limited theoretical explanations for why such guidelines may be effective from a human information processing (HIP) perspective. The lack of a cognitive explanation limits understanding of when and how guidelines can be applied. In order to better support conceptual design, various cognitive modeling techniques and tools have been developed based on HIP architectures. However, these techniques and tools also have several limitations from a design perspective. Existing tools are not easy to use and designers or developers may need extensive training and practice in use. Furthermore, there is currently no fundamental set of tool capabilities, such as providing a task workload analysis or identifying patterns of HIP (e.g., memory use), simulating visual object use (e.g., eye movements), providing interface design support, etc. This research integrated various capabilities of existing modeling tools into a new enhanced cognitive modeling language based on GOMS (Goals, Operators, Methods, and Selection Rules).

While GOMS modeling methods and the GOMS language are considered easy to learn and use, the modeling approach has several limitations. The language is limited to representing expert behavior in tasks. In addition, GOMS models do not support modeling of lower-level behaviors, such as specific forms of visual processing (e.g., foveal vs. peripheral) as well as parallel processing of visual and motor operations. Another major limitation of GOMS modeling is that the operator time estimates are deterministic. Therefore, model output may not accurately represent individual differences in performance or the stochastic nature of human behavior in complex tasks. On the basis of these limitations, this research developed a new computational cognitive modeling tool using an enhanced-GOMS language to aid complex system designers in assessing human performance and errors in using complex automated systems.

A human information processing model described by Wickens (1992) was used in this research as a cognitive architecture to support and constrain E-GOMSL model coding. New operators as part of the E-GOMS language were defined with four properties based on Wickens HIP model, including: (1) the cognitive processing channel used, (2) control objects, (3) operator syntax and (4) operator times. Each channel has control objects (e.g., flow of control, parallel processing, etc.). E-GOMSL operator syntax is similar to GOSML and NGOMSL operator syntax. The new operator set was primarily based on NGOMSL operators, as originally defined by Kieras (1997); however, the control structure of EGOMSL models follows GOMSL models, in order to support compilation and model execution with a simulation engine. As the second step in E-GOMSL development, stochastic variables were defined to represent operator times in behavior models. Because computational cognitive modeling is conceptually similar to discrete event simulation of human task performance, methods used in systems simulation for representing or quantifying event processing times have been extended to cognitive modeling. An overall stochastic time estimate can be calculated as the summation of all operator time estimates in an E-GOMSL model. The time estimates can be considered to represent the range of human performance, including normal (average), super skill and slacker behavior (Niebel & Freivalds, 2003).

The cognitive modeling tool development included: a prototyping module; a user activity flow diagram (AFD) development module; an AFD to E-GOMS language translator; an E-GOMSL editor; a model parser and compiler; and a model simulation tool and report generator. A designer is able to use images to define a prototype including visual and non-visual objects (e.g., auditory interfaces). The designer can also develop an AFD based on the results of a cognitive task analysis (CTA) involving expert operators. The AFD is directly translated to E-GOMS by the translator module. After coding the model, the parser and compiler can be used to obtain a quantitative analysis including task execution times based on stochastic estimates of individual operation times and a workload analysis. With these results and the GOMSL models, the simulator can be used to visualize the flow of HIP, represent patterns in HIP, and present a graphical workload analysis. Last, the report generator can be used to produce a summary of the quantitative analysis and simulation.

In order to validate the results of the modeling tool, a flight simulator experiment was conducted with a futuristic form of cockpit automation (a Continuous Descent Approach (CDA) tool for flight route replanning). A CTA was conducted to identify pilot behaviors and to generate a data set for validation of the cognitive model output. An E-GOMSL model of pilot behavior with the CDA tool was compared against the experiment data. There was a marginal positive correlation between the model and pilot experiment task times ($p = 0.3489$, $n = 27$, $p = 0.0745$). Comparison of E-GOMSL model outputs at various points in task performance with actual pilot heart rate responses (correlation analysis: $p = 0.2055$, $p = 0.0181$) indicated working memory (WM) item counts from a model could serve as a basis for predicting automation and task-induced cognitive load. In general, when the model predicted WM count was at a minimum, the HR response for pilots revealed low arousal. When the model-predicted count was at a maximum, the HR response for pilots revealed high arousal. These findings indicate that the E-GOMSL model may explain differences in automation or task-induced cognitive load in terms of WM use.

In line with expectations, results demonstrated the modeling approach to support accurate explanation and prediction of human behaviors and performance in using complex systems. The findings of this research support the new EGOMSL tool use during the conceptual design of complex human-in-the-loop systems and/or interfaces.

Modeling Users' Risk-related Behaviors when Interacting with Computer Systems - Ben-Asher & Meyer

Computer security is gaining importance because of the ubiquitous introduction of computers into all domains of life, the use of computers to store and access sensitive information of various kinds (e.g., bank accounts, medical records), and the increasing use of mobile devices to access these systems. In recent years, it has become clear that the human user is often the weakest link in computer security. Even if the system requires long and complex passwords, it becomes unsecured if users paste them on their computer monitors. Also, even if one has sophisticated algorithms for detecting malicious software, for instance on websites, the user may override the system recommendation and become exposed to these threats. The design of adequate computer security requires us therefore to predict the user's risk-related behaviors with computer systems. An adequate understanding of user behavior and the prediction of user actions will allow us to design systems and security measures, so that users will tend to act securely.

Two main issues need to be considered when modeling users' risk-related behavior with computer systems. First, very little behavioral data are available on how users cope with security risks. The main reason is that publishing information on how users, for instance, respond to indications of security threats and what affects their responses to these threats can possibly be exploited by those who generate threats and increase the severity of threats. Second, the user's risk-related behavior may actually be a combination of several different, interrelated behaviors. We suggest the notion of a ”triad of risk-related behavior” (Ben-Asher & Meyer, submitted for publication), where the user’s coping with security issues in computer systems is affected by the user’s exposure to risk, the installation and setting of security features, and the response to risk-related communications.

We developed an experimental system, based on the Tetris game, to allow us to collect empirical data on all three behaviors. In our version of the game users try to accumulate as many points as possible but, different from the usual Tetris, completed rows remain on the screen until the user decides to ”save them” (an action that stops the game and is therefore costly users are paid according to their performance, and the game is limited in time). Occasionally ”attacks” occur in which a malicious virus deletes part of the cells the user has accumulated and hasn’t saved, yet. The user sees alerts from a security system (with imperfect validity) about the possibility of an attack.

To apply the insights gained from the experiments for the generation of design recommendations, it is important to model the different behaviors and their interactions. One model, based on the Memo-workbench, focuses on the analysis of the user-system interaction (Möller, Ben-Asher, Englert, & Meyer, 2011) We have begun to develop models, adopting three additional modeling approaches:

1. A cost-benefit model to predict the optimal user actions, given the properties of the system.

2. A reinforcement-based learning model in which we attempt to predict the changes in the security system settings and in the users’ tendency to expose themselves to risk and to respond to alerts.

3. A system dynamics model that analyzes the feedback loops in the process.

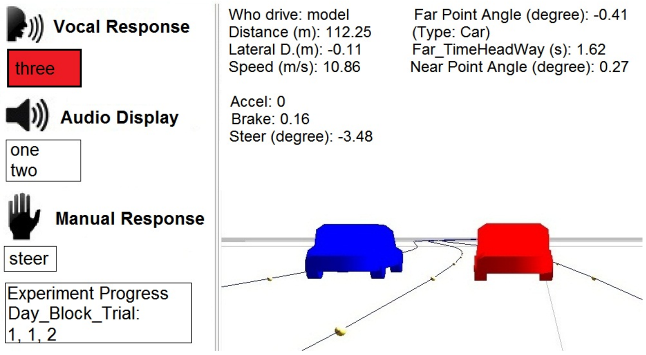
All models start with the parameters of the experiment for a given condition and then generate predictions of user behavior, which we compare to empirical results. We discuss the challenges that exist when trying to model a complex behavior in an experimental microworld in which users’ actions result from the combination of different, interrelated behaviors. We also discuss the advantages and problems with each of the different modeling methodologies we employed and point towards the requirements for a comprehensive modeling of users’ risk-related behaviors with computer systems.

ACTR-QN: Integrating Queueing Network and ACT-R Cognitive Architectures - Cao & Liu

ACTR-QN is a cognitive architecture that integrates Adaptive Control of Thought-Rational (ACT-R) and Queueing Network (QN) architectures. ACT-R (Anderson et al., 2004) has sophisticated declarative memory mechanism based on chunk activation and procedural memory mechanism based on production rule utility. It is particularly powerful in modeling cognitive tasks such as learning and problem solving. QN (Liu, Feyen, & Tsimhoni, 2006), on the other hand, has its mathematical basis of queueing theory, which supports the modeling of complex mental structures and scheduling mechanisms. As a result, the QN architecture has its strength in modeling multitask performance and mental workload. The integrated ACTR-QN represents ACT-R as a QN, whose servers are ACT-R modules and buffers with information paths in between and entities correspond to ACT-Rs information units such as chunks and production rules. Theoretically, ACTR-QN allows modelers to combine the power of ACT-R and QN and examine a wider range of fundamental cognitive issues from new perspectives, for example, modeling multitask performance involving complex cognitive tasks.

For cognitive engineering applications, a software program implementing ACTR-QN has been developed using Micro Saint Sharp (www.maad.com), which was chosen because it provides natural supports for QN modeling and visualization. Further, it is the same platform on which IMPRINT is implemented. Full integration of ACTR-QN was achieved by porting ACT-R (v 6.0) from Lisp into Micro Saint Sharp (C#). In addition, ACTR-QN also integrated the PG-C version of utility computation and the recent work on threaded cognition. Workload modeling capability was inherited from QN using server utilization.

Each model in ACTR-QN has two parts: the mind and the task. To build the mind part, including chunks, production rules, and parameters, ACTR-QN reads and uses the same ACT-R codes. For the task part (displays and controls), easy-to-use templates have been developed to model both static tasks, in which display stimuli are predetermined and not affected by responses, and dynamic tasks, in which responses affect display stimuli dynamically, such as driving. Modelers simply need to follow instructions and specify the parameters of a task, such as the frequency of a tone or the geometry of a road.



three

one
two

steer

Experiment Progress
Day_Block_Trial:
1, 1, 2

Who drive: model
Distance (m): 112.25
Lateral D.(m): -0.11
Speed (m/s): 10.86

Far Point Angle (degree): -0.41
(Type: Car)
Far_TimeHeadWay (s): 1.62
Near Point Angle (degree): 0.27

Accel: 0
Brake: 0.16
Steer (degree): -3.48

Figure 1. Task visualization in ACTR-QN

ACTR-QN provides visualization of the mind, the task, and mental workload. Figure 1 illustrates the visualization of model performing a dual task of auditory-vocal arithmetic addition (left) and driving (right). Model testing shows that ACTR-QN produces the same results as ACT-R for typical cognitive tasks. Future research will examine the benefits of further integration between ACT-R and QN cognitive architectures, especially in modeling performance and workload in multitask scenarios involving complex cognition.

Panel

During the Conference, part of the Symposium presentation included a discussion among the three panelists, three presenters, and the audience regarding the differences and similarities of cognitive modeling for human factors applications versus other types of cognitive modeling. It is unfortunate that a transcript of this exchange cannot be provided here.

Acknowledgements

Kaber, Gil, and Kim's work was supported by NASA Ames Research Center under Grant No. NNH06ZNH001. Mike Feary was the technical monitor. The opinions and conclusions expressed in the associated paper are those of the authors and do not necessarily reflect the views of NASA. This research was completed while the third author, Sang-Hwan Kim, worked as research assistant at North Carolina State University.

Gray's work was supported, in part, by grant N000141010019 from the Office of Naval Research, Dr. Ray Perez, Project Officer.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglas, S., Lebiere, C., & Quin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060.

Ben-Asher, N., & Meyer, J. (submitted for publication). The triad of risk-related behavior in computer security: effects of skill, costs and attack likelihood.

Gray, W. D. (2008a). Cognitive architectures: choreographing the dance of mental operations with the task environments. *Human Factors*, *50*(3), 497–505.

Gray, W. D. (2008b). Cognitive modeling for cognitive engineering. In R. Sun (Ed.), *The cambridge handbook of computational psychology* (pp. 565–588). New York: Cambridge University Press.

Kieras, D. E. (1997). A guide to GOMS model usability evaluation using NGOMSL. In M. Helander, T. K. Landauer & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd Edition, pp. 733–786). Elsevier.

Liu, Y., Feyen, R., & Tsimhoni, O. (2006). Queueing network-model human processor (qn-mnp): a computational architecture for multitask performance in human-machine systems. *ACM Transactions on Computer-Human Interaction*, *13*(1), 37–70.

Möller, S., Ben-Asher, N., Englert, K.-P., & Meyer, J. (2011). Modeling the behavior of users who are confronted with security mechanisms. *Computers & Security*, *30*(4), 242–256.

Niebel, B. W., & Freivalds, A. (2003). *Methods, standards, and work design* (12th Edition). Boston: WCB/McGraw-Hill.

Pew, R. W. (2008). More than 50 years of history and accomplishments in human performance model development. *Human Factors*, *50*(3), 489–496.

Pew, R. W. (2007). Some history of human performance modeling. In W. D. Gray (Ed.), *Integrated models of cognitive systems* (pp. 29–44). New York: Oxford University Press.

Wickens, C. D. (1992). *Engineering psychology and human performance* (Second). New York: HarperCollins.