General Methods for Communicating the Structure and Content of a Cognitive Model

Session Chair: Walter Warwick
Alion Science and Technology

Panelists:

Christian Lebiere, Carnegie Mellon University
Randolph M. Jones, Soar Technology
David Reitter, The Pennsylvania State University
Stuart Rodgers, TiER1 Performance Solutions
Scott A. Douglas, AFRL

The modeling and simulation of human performance forces the analyst to confront a range of well-known but difficult challenges. One challenge the analyst does not seem to face is a shortage of human performance modeling tools. But because there is no uniform framework for expressing the content and structure of a human performance model, it is difficult to understand what is at stake in the implementation of a given model and all but impossible to compare and contrast different models despite the proliferation of quantitative modeling tools. The inability to communicate model structure and content is not just a practical shortcoming: it is a major impediment to assessing the validity, plausibility, and extensibility of human performance models. The latter aspect is particularly important as it prevents the incremental construction of large human performance models along standard software engineering practices. The goal of this panel discussion is to review past and ongoing efforts to develop general languages that specify cognitive models at a functional level of description. We do not expect a standard to emerge from this discussion, but rather we hope to canvass both the theoretical and practical issues that confront any attempt to develop a uniform language that describes different modeling frameworks.

Copyright 2013 by Human Factors and Ergonomics Society, Inc. All rights reserved. DOI 10.1177/1541931213571132

INTRODUCTION

The modeling and simulation of human performance forces the analyst to confront a range of well-known but difficult challenges. One challenge the analyst does not seem to face is a shortage of human performance modeling tools. Indeed, as Pew and Mavor (1998) discuss, there is a wide variety of tools that can be used to represent human performance. While it might seem that such variety would ensure that the analyst could always find the right tool for the job, this plurality of approaches actually reveals largely unacknowledged challenges: because there is no uniform framework for expressing the content and structure of a human performance model, it is difficult to understand what is at stake in the implementation of a given model and all but impossible to compare and contrast across different models despite (or, perhaps, because of) the proliferation of quantitative modeling tools.

The inability to communicate model structure and content is not just a practical shortcoming: it is a major impediment to assessing the validity, plausibility, and extensibility of human performance models. Consider that before we can even begin to validate the predictions of a model we need to identify which aspects of performance the model is intended to predict. Discriminating between consequential model predictions and the results of incidental implementation details can be far from trivial, and even then different measures of performance might require different validation methods. Not every model predicts accuracy and latency nor does a measure of a correlation coefficient or root mean square error always establish a fit between model and data. Similarly, before we assess the construct correspondence between models and the behavior they purport to represent, we must be able to characterize the process-level features of the model in terms that are neutral with respect to a particular modeling approach. For example, the comparison of two models of human multitasking might require that we distinguish between the rapid interleaving of two otherwise serial processes from true parallelism of simultaneous activity; in such a case there could be real representational differences hidden in a common vocabulary that would become obvious only with recourse to a more general set of terms. Finally, a disappointing fact of human performance modeling is that the models themselves are often regarded as black boxes. Without insight into the internal workings of a model it is impossible to determine whether the approach implemented is likely to be scalable or extensible to a new domain. As a result, many modeling wheels are continually reinvented, and a hard ceiling on model complexity is all but established in practice. These shortcomings add considerable expense to individual modeling efforts and slow progress across the field, as the kind of scalability that has become common in software engineering through reductionist specification and combination is out of reach of modeling frameworks.

It turns out that while there may be many specific models of human performance, they are often employed in similar ways. For example, human performance tools almost always exploit some decomposition of the simulated work domain, whether in terms of the dimensions of a problem-space, a hierarchy of goals to be achieved, an enumeration of discrete tasks and courses of action that could be performed, or a set of different states that define the dynamic control of a system. Such decompositions are necessary in order to map the continuous activity of the human to the specific computational mechanisms available in the human performance modeling tools, and more fundamentally reflect the dual constraints of the complexity and interactivity of the world and the capacity limitations (e.g., attentional, working memory) of human cognition (Simon, 1969). These mechanisms, in turn, group along familiar lines: there are mechanisms for sensing and affecting (i.e., mapping system variables to model inputs and outputs); mechanisms for searching and planning; and mechanisms for recognizing, reasoning, and deciding. Associated with those mechanisms, there are often various parameters for affecting the continuous and variable aspects of human performance (e.g., remembering, forgetting, fatiguing, etc.). The important point is that while human performance modeling tools might differ in terms of specific implementation details, there is considerable overlap among these tools at the functional level of description.

The goal of this panel discussion is to review past and ongoing efforts to develop general languages that describe cognitive models at a functional level. Unlike many previous efforts (cf. Ritter, Haynes, Cohen et al., 2006), our interest is not so much in languages that can be compiled into executable code for a given cognitive architecture, but rather identifying and then codifying features that are common across cognitive architectures. We hope to canvass both the theoretical and practical issues encountered in *any* attempt to develop a uniform language that describes different modeling frameworks. Each panelist has had direct involvement in the development of a high-level modeling language or represents the user community's requirements for such a language. This experience will allow the panelists to discuss the challenges they faced and how they overcame them (or what they would do differently next time). Our emphasis will be on lessons learned more than the details of any particular high-level language. That said, for those interested in advancing a particular language, a "birds-of-a-feather" meeting will follow the panel discussion so that we might solicit guidance from the modeling community for a "human performance modeling mark-up language" currently under development.

PANELIST ABSTRACTS

Human Performance Modeling Mark-up Language

Christian Lebiere
Carnegie Mellon University

The goal in developing the Human Performance Modeling Mark-up Language (HPM-ML) is not to impose top-down methodological standards across human performance modelers, but rather to provide a common vocabulary with which to express what is already contained in their models. HPM-ML will be illustrated using a set of Human-In-Control (HIC) models for a Ballistic Missile Defense System (BMDS) task. There are any number of approaches one could take to modeling a human interacting with and controlling a BMDS. For example, these could range from a very simple and very abstract representation of the "goodness" of a decision making process as determined by a probabilistic draw, to a more concrete but still highly stylized finite-state representation of the procedural steps followed by an ideal operator, to a more cognitively-based representation of the knowledge, strategies, techniques, and procedures that define decision making at the tactical level, to an even more sophisticated representation of the continuous learning and adaptation that supports the abductive inference needed for the simulated decision maker to diagnose and act in a situation. Exploring the full range of possible representations for a BMDS operator ensures that the HPM-ML is adequately expressive. These models are described using the Human Behavior Architecture

(HBA) (Warwick et al., 2008), a hybrid of the C3TRACE task network modeling tool (Swoboda & Plott, 2012), and the ACT-R cognitive architecture (Anderson & Lebiere, 1998; Anderson et al., 2004). As such, the HBA supports a span of human performance modeling from the coarse-level representation of high-level operator functions and tasks to the fine level of detail needed to represent an operator's cognitive mechanisms and limitations. HPM-ML includes modeling primitives that cluster around three basic functions of search, comparisons, and selection. Within these functions it distinguishes between parallel versus serial search, exact versus fuzzy match, and probabilistic versus rule-based versus adaptive selection. Each of these mechanisms can be further refined and combined with others. For instance, rather than a random draw among equally likely alternatives, probabilistic choice can be conditioned on contextual features or even given as the result of another inferential process like the Bayesian accumulation of evidence. Similarly, rule-based approaches might be distinguished by how rules are selected—by exact matching criteria versus fuzzy-matching criteria, for example. Adaptive approaches can be distinguished by their representation of features or states that learning mechanisms are grounded in. An essential aspect of HPM-ML is its extensibility to reflect new or changing modeling paradigms.

Lessons Learned About Formal High-Level Abstractions for Modeling the Mind

Randolph M. Jones
SoarTech

Computer science is the science of understanding phenomena at multiple levels of abstraction, including the development of formal representations to describe those levels. Cognitive Science, as well as *being* one level of abstraction for understanding brains and minds, also *applies* this computational approach to understanding the mind. Cognitive architectures and cognitive models represent formal descriptions of mind processes, as well as the content (sometimes called "knowledge") of those processes. Computer science teaches us that there is no right level of abstraction; rather, each level of abstraction is useful for different forms of understanding. Thus, there are neural-level, cognitive-level, and social-level descriptions of mind processes, to name a few. Symbolic cognitive architectures, such as Soar and ACT-R, have proven to be at the right level of abstraction to support scientific development of symbol-level theories of mind, as well as numerous individual models of various cognitive tasks and capabilities. However, this level of abstraction has proven somewhat tedious for attempts to build models of human reasoning of increasing complexity. By studying a number of Soar and ACT-R models, we identified several programming patterns or idioms used by cognitive modelers, which had the potential to serve as the basis for a higher-level language for symbolic cognition. We pursued methods for formalizing these patterns into HLSR (High Level Symbolic Representation), a formal language that compiles to ACT-R and Soar and provides modelers with a tool for building more complex models at a more appropriate level of abstraction. These efforts produced a useful formal language for several patterns, as well as experimental results demonstrating modeling improvements using HLSR's high-level abstractions. Our efforts also produced lessons for future development. For example, it is useful to separate procedural and declarative abstractions, in part to allow high-level models to reason over their own representations. Additionally, future high-level abstractions should be paired with visual representations to make interactions between components clearer and easier to manage for model developers. It is also useful to consider formal systems that mix different levels of abstractions for different types of cognitive processing, such as numerical reasoning, spatial reasoning, empirical and analytical learning, and knowledge-intensive decision making.

Computational and Underspecified Approaches to Standardization

David Reitter
The Pennsylvania State University

The design of a cognitive modeling program that makes incremental improvement and standardized evaluation possible should adopt the following three ideas:

(1) ACT-UP (Reitter & Lebiere, 2010) re-casts the cognitive architecture ACT-R as a framework that models cognitive processes as computer programs. It implements modeling as programming because computational models such as production rule systems eventually translate to algorithms. Modelers can and should benefit from encapsulation, reusability, and compile-time verification provided by all modern programming languages. Behavioral and neurophysiological predictions are made by the ACT-UP library that implements invariant cognitive mechanisms.

(2) Under-specification allows a modeler to define subprocesses in detail while leaving out portions of a model that cannot be evaluated empirically at the time. Clearly defined interfaces between these different kinds

of processes are important, as are mechanisms to specify variables that may be estimated from data, e.g., behavioral measures such as time delays. I argue that this is a scientifically more honest approach than to develop overly specific models manually to fit available data.

(3) Existing frameworks provide some much-used components: a) declarative and procedural memory: learning through repetition; un-learning through decay; and blending of, abstraction from, and chunking of sets of previous experiences; b) perception (selective attention, encoding); and c) explanations of results in dual-task paradigms via limited parallelism using modular resources.

Understanding these three ideas—modeling as programming, the utility of under-specification, and the existence of common components across architectures—will provide insight into the challenges of developing a general mark-up language for cognitive models.

Who Uses Human Performance Models and What Do They Need?

Stuart Rodgers
TiER1 Performance Solutions, LLC

Outside of research, various camps in the U.S. Department of Defense (DoD) use modeling and simulation for many analytical purposes and for training and rehearsal purposes. In the analytical camp are activities such as systems and requirements analysis. These are used to ensure the military adequately understands how a system will be used and to estimate how a system will perform in a given scenario. Also, in the analytic camp are “system of systems” analyses that help answer the question “Is the military buying and developing the right mix of systems for the expected global scenarios?” In the training and rehearsal camp, the military uses models and simulations for both replacement and augmentation of live systems due to the time, cost, and risks involved in training with live systems. As government, academia, and industry have matured models of human performance, the analytical camps and training camps have included human performance models into the suite of models and simulations to improve the systems analysis and the military training capabilities. This talk will outline some use cases of human performance models by these camps and discuss the needs and priorities in these pursuits.

Event Processing and Levels of Abstraction in a Modeling and Simulation Framework

Scott A. Douglas
Air Force Research Laboratory

For the last three years, an AFRL Large-Scale Cognitive Modeling (LSCM) research effort has worked to close capability gaps slowing the development and deployment of training, automated sensemaking, and decision support capabilities built using cognitive models and agents. To do this, the LSCM initiative has researched and developed the following:

- Domain-specific languages (DSLs) that allow cognitive scientists to specify models and agents using domain and behavioral abstraction hierarchies they themselves define.
- Authoring environments supporting model development using these DSLs.
- Code-generation technologies that transform models specified in these authoring environments into executable artifacts.
- Service Oriented Architectures in which models and agents are executed.

The vanguard DSL/execution architecture combination of the LSCM initiative encourages users to conceive of, specify, and execute their models as complex event processing agents (Luckham, 2002).

The DSL, currently referred to as RML, is based on a single fundamental representation called an event. The abstract syntax of RML allows users to formally describe relationships between events using (1) event patterns, (2) event pattern rules, (3) behavior models, and (4) representations of domain knowledge referred to as cognitive domain ontologies. Event relations expressed in event patterns can be used to capture part-whole, correlational, temporal, and causal relationships between events. Event relations expressed in event pattern rules can be used to create an event abstraction hierarchy reflecting the unique interleaving of conceptual and behavioral abstractions proposed by the modeler and exploited by the model. Event relations captured in behavior models can be used to formally describe behavior using constructs at any level of the event abstraction hierarchy. Lastly, event relations specified in cognitive domain ontologies can be used to capture a form of non-deterministic domain knowledge that can be used by a model to “soft-assemble” effective action *in situ*.

I will describe RML and the cognitively enhanced complex event processing (CECEP) architecture in which RML models execute. The central argument of my presentation will be that cognitive modelers are using languages and simulators that are complex event processing frameworks. The challenge to developing a *lingua franca* for the specification, comparison, and communication of cognitive models is coming to an understanding of how modelers using different architectures or modeling behavior at different levels of abstraction can unify their efforts. I will argue that such a unification can be achieved through the adoption of standardized (1) event abstraction hierarchies and (2) complex event processing frameworks into which they can semantically anchor their specific modeling languages and event sources (architecture modules, knowledge sources, methods of inference, etc.).

ACKNOWLEDGEMENT

This material is based upon work supported by the Missile Defense Agency (MDA) under Contract No. HQ0147-13-7414. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of MDA.

REFERENCES

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought.* Mahwah, NJ: Lawrence Erlbaum Associates.

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

Pew, R. W., & Mavor, A. S. (Eds.). (1998). *Modeling Human and Organizational Behavior*. Washington, DC: National Academy Press.

Reitter, D., & Lebiere, C. (2010). Accountable modeling in ACT-UP, a scalable, rapid-prototyping ACT-R implementation. In *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 199-204). Philadelphia, PA: ICCM.

Ritter, F. E., Haynes, S. R., Cohen, M., Howes, A., John, B., Best, B., Lebiere, C., Jones, R. M., Crossman, J., Lewis, R. L., St. Amant, R., McBride, S. P., Urbas, L., Leuchter, S., & Vera, A. (2006). High-level behavior representation languages revisited. In *Proceedings of ICCM -2006—Seventh International Conference on Cognitive Modeling* (pp. 404-407). Trieste, Italy: Edizioni Goliardiche.

Simon, H. A. (1969). *The Sciences of the Artificial*. Cambridge, MA: MIT Press.

Swoboda, J., & Plott, B. (2012). C3TRACE: Modeling information flow and operator performance. In P. Savage-Knepshield, J. Martin, J. Locket III, & L. Allender (Eds.), *Designing Soldier Systems* (pp. 431-446). England: Ashgate Publishing.

Warwick, W., Archer, R., Hamilton, A., Santamaria, A., Chong, R., Allender, L., & Kelley, T. (2008). Integrating architectures: Dovetailing task network and cognitive models. In *Proceedings for the Seventeenth Conference on Behavior Representation and Simulation*. Providence, RI: SISO.