Integrated Cognition
AAAI Technical Report FS-13-03

The Macro Architecture Hypothesis: A Theoretical Framework for Integrated Cognition

Robert L. West
Institute of Cognitive Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada, K1S 5B6
robert_west@carleton.ca

Abstract

This paper presents a theoretical framework for researching how humans integrate their cognitive abilities to accomplish complex work in the real world. The work draws on the distinction between micro and macro cognition and proposes the existence of a macro architecture. The idea of a macro architecture is discussed in terms of facilitating research on integrated cognition. The SGOMS architecture is described as a working example of a macro architecture. Alternative frameworks are also discussed.

Introduction

Newell's (1973) idea of a cognitive architecture has guided much of the work on computational models of human cognition, but it has mainly been applied to the results of Cognitive Psychology experiments. Although successful, it can also be argued that this approach has been limited in terms of explaining the human ability to integrate different cognitive abilities to perform complex tasks, which was part of the original goal of creating cognitive architectures (Newell 1990).

Micro and Macro Cognition

Recently, the study of cognition has been sub-divided into micro and macro cognition (Cacciabue and Hollnagel 1995; Klein et al., 2003), where micro cognition refers to cognition as it is studied in Cognitive Psychology experiments and macro refers to cognition in complex, real world tasks. The question is then whether or not cognitive architectures, built to model micro cognitive tasks, can scale up to usefully model macro cognitive behavior. Some have argued that they cannot (Klein et al. 2002), due to the limitations and artificiality of cognitive psychology experiments. We have argued that micro cognitive architectures can, potentially, scale up to model macro cognition, but that it will require a deliberate and systematic attempt to theoretically link micro and macro theory (West and Nagy 2007; West and Pronovost 2009; West and Somers 2011). This paper lays out a framework for doing this, with examples from our own attempts to build the SGOMS architecture.

System Levels

The idea of a cognitive architecture is based on a systems level view of intelligence (Newell 1990). The assertion that the human brain is designed in a systems level way is, itself, a hypothesis. Difficult philosophical questions can be asked about the reality of systems levels and architectures but we will ignore them in this paper (although see West and Leibovitz 2012). However, the whole idea of treating macro cognition (or social cognition) as a coherent field of study suggests that a corresponding systems level exists in some way.

The Macro Architecture Hypothesis

The macro architecture hypothesis is meant as a way of framing the relationship between macro cognition, micro cognition, and the neural activity that produces it all. The hypothesis proposes there is a macro level architecture that describes the functions of the macro systems level. The macro cognitive architecture is built on the micro cognitive architecture (which is built on the neural architecture). The macro cognitive architecture is hypothesized to exist in the brains of individuals and to enable us to apply our information processing abilities (micro cognition) to complex, dynamic, multi-agent tasks (macro cognition).

A lot of very good work has been done scaling up micro cognitive architectures to model at the macro level (see Ritter et al. 2006 for a review). The macro architecture hypothesis is not a new way of thinking about this. Instead

---

it is a framework derived from ongoing work in this field. It is also a hypothesis and there are other ways of thinking about it (see the conclusion). However, one goal of this project is to generate a more explicit dialogue about how to model at the macro level. Another goal is to create a framework that clearly shows the relevance of cognitive architectures to researchers outside of the modeling community.

Unit tasks

Newell's (1990) original scheme for understanding systems levels included the neural level, the cognitive level, and the social level, which involved the cognition across multiple agents. The social level is now more commonly known as Distributed Cognition (Hutchins 1995). The macro cognitive level is hypothesized to exist between the (micro) cognitive level and the distributed cognition level.

Although Newell did not have a separate systems level to mediate between (micro) cognition and social actions, he did have a control mechanism. The *unit task* was hypothesized to mediate between the structure of the task and the abilities of the (micro) cognitive system (Newell 1990). Specifically, the unit task defined how the task was mentally broken up to avoid both overload and down time. For example, a task that involves remembering would be broken down into parts so that the capacity of short-term memory would not be overloaded. Likewise, parts of the task that will not necessarily follow each other are stored as separate unit tasks, so that the agent can be released in-between to do other things if there is time (i.e., avoid downtime).

When building a model of a task in a micro architecture, it is common practice to first construct a unit task model (i.e., a high level model of the components of the task) and then to work out how the micro cognitive architecture accomplishes each unit task. Therefore, models built in this way can be viewed as implicitly embodying two hierarchically arranged systems - the processes contained within the unit tasks and the system for selecting and coordinating the unit tasks. However, the tasks used in most psychology experiments arguably fit in a single unit task, so the higher level system is often not needed.

In theory, a macro cognitive architecture can be built based on any appropriate theoretical structure. However, since numerous different architectures and models embrace the concept of the unit task, we have focused on that. So, working within this theoretical structure, the macro cognitive architecture can be thought of as the system for selecting unit tasks. In fact, it can be argued that the idea of the macro cognitive architecture is implicit in Newell's original scheme.

Normally, when building a model, the system for managing unit tasks (when there is more than one unit task) is constructed on an ad hoc basis for the task. However, this defeats the purpose of using an architecture (i.e., to create a unified system). In contrast, using the concept of a macro cognitive architecture means that a single system for managing unit tasks should apply across all tasks.

Empirical Evaluation

Evaluating macro models is both similar and different from evaluating micro models. The big difference is in terms of the types of measurements and analyses that can be used. Macro cognition is generally at a time scale that is too long and too noisy to use reaction time measures. Also, macro tasks are not randomized so it is often the case that the order of events will not be the same across trials, and this means that averaging is of limited use. Finally, macro cognition is often concerned with higher-level constructs that can only be evaluated using qualitative methods.

Here, it is important to note the distinction between system levels and levels of analysis. Level of analysis refers to analyzing something at a particular level of measurement (e.g., fMRI, RT, questionnaire, discourse analysis, etc.). Systems level means that all behavior at a particular level can be explained by one unified architecture. Level of analysis and systems level tend to be correlated in that higher systems levels are usually studied at higher levels of analysis, but this is not a rule.

In terms of similarities, macro models and micro models are both models and certain truths about modeling apply equally to both. In particular, we argue that Newell's (1973) critique of modeling in cognitive psychology applies equally to the study of macro cognition. In his well-known paper, *You can't play 20 questions with nature and win*, Newell pointed out that creating different models for each cognitive phenomenon is of limited use. The models provide insight into individual phenomena but the practice, unchecked, results in a bewildering plethora of unrelated models. This criticism can be applied to macro cognition as well. Without the use of a macro cognitive architecture, the result is a vast array of ad hoc models. Note that this is true even if a micro cognitive architecture is used to make the models, since the system for choosing unit tasks is still an ad hoc solution designed for each specific task.

Also, using a macro cognitive architecture creates a systematic way to relate results from higher levels of analysis (i.e., macro cognition, social sciences, and humanities) to results at a lower level of analysis (i.e., cognitive psychology, neuroscience), and could lead to the meaningful use of lower level analyses for analyzing higher-level functions.

Unified Lakatosian Framework

In terms of evaluation, like micro cognitive architectures, macro cognitive architectures cannot be evaluated within a strict Popperian scientific framework. This is because, although a specific model built within an architecture can be disproven, this does not mean that the architecture is false. It is also possible that the model was built in the wrong way or that the architecture needs an additional component or a minor adjustment, or that the task was misunderstood. Because of this, Newell (1990) noted that cognitive architectures should be evaluated within a Lakatosian scientific framework (Lakatos 1970; also see Cooper 2007 for discussion of this framework applied to cognitive architectures). In the Lakatosian framework theories are evaluated across time in terms of progress and usefulness. Therefore, in this framework, an architecture is considered scientific as long as it continues to further unify different phenomena and produce parsimonious explanations. For example, under this framework the theory that planets orbit in circles was initially a valid scientific theory as it produced significant progress in understanding our solar system, but it became less valid as more and more epicycles were required to describe the orbits, which were actually elliptical and only approximated by circles.

An important concept for applying the Lakatosian approach to cognitive architectures is Lakatos's idea that a theory (or architecture) can be understood in terms of core and peripheral commitments. When a model fails, the first line of defense is the peripheral commitments, whereas the core concepts are only challenged if it is not possible to make progress by altering the peripheral concepts (ACT-R is a good example of an architecture that is developed in this way, see Cooper, 2007, for discussion).

Cooper (2007) notes that, in addition to core and peripheral commitments, models built in (micro) cognitive architectures have to be evaluated in terms of the accuracy of the task model (i.e., the knowledge added into the architecture to allow it to do the task). He also points out the scientific desirability of validating the task model separately so it does not become an added source of variance for evaluating the architecture. Tasks used in cognitive psychology experiments are kept very simple so the task model is reasonably obvious. However, this is not the case for macro cognitive tasks. A lot of assumptions about the task knowledge are needed to get a micro cognitive architecture to model most macro cognitive tasks.

A macro cognitive architecture would ameliorate this problem. A macro cognitive architecture would have core and peripheral components that constrain how task knowledge is organized. Figure 1 illustrates how a macro cognitive architecture could be combined with micro cognitive and neural architectures within a Lakatosian framework adapted to accommodate systems levels. Note that in this scheme the core mechanisms of the architecture above are used to challenge the peripheral mechanisms of the architecture below. That is, the system above gives guidance in further developing the system below. If the core mechanisms of the level above required changes to the core mechanisms of the level below then the system levels would be considered incommensurate. In terms of upward constraints, it must be possible to build the architecture above on the architecture below (possibly with some modifications to peripheral components).

Using this framework, a macro cognitive architecture can be evaluated in two different ways. The first is whether it can reasonably and efficiently model human behavior across a diverse set of macro cognitive tasks. The way to evaluate this is to build the architecture and test it across a diverse set of macro cognitive tasks. The second way is whether the macro cognitive architecture can be reasonably and efficiently produced by a micro cognitive architecture. The way to evaluate this is to build the macro cognitive architecture on top of a micro cognitive architecture.

Note that this also changes how the micro cognitive architecture is evaluated. Instead of being evaluated directly on its ability to perform the task it is evaluated based on its ability to provide the core and peripheral functionality of the macro cognitive architecture. If it can, and the macro architecture can model the task, then it should all work (although the whole thing should be simulated to check it).

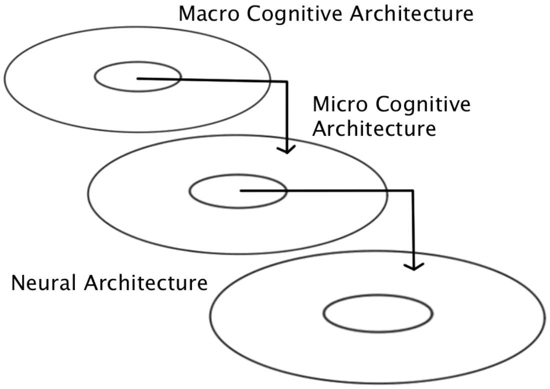

Figure 1. Lakatosian framework adapted for use with systems levels

Neuroscience

Some people are uncomfortable with a systems level approach and find it hard to think about functions that are divorced from neurons. Currently the dominant way of thinking about (micro) cognitive functions is that that they are produced by dedicated groups of neurons, or neural modules. However, this is not the only possibility. The alternative is neural reuse (Anderson 2010) whereby cognitive functions are created through the interaction of different neural groups. We do not take a stand on this but neural reuse is probably how the macro cognitive architecture is produced. That is, it is produced through the interaction of basic cognitive functions, which are produced through neural activity. Therefore, neural evidence for the macro cognitive architecture would come from the pattern of interactions across the brain and not from localization. Neural imaging techniques are now able to identify the network of brain areas involved across tasks with reasonable accuracy (e.g., Varela et al. 2001) so it is theoretically possible to link the macro architecture directly to neural activity.

In terms of the neural origins of the macro cognitive architecture, there are two possibilities. One is that the macro cognitive architecture is genetically hardwired. That is, brains come into the world designed to get tasks done in the world. The other possibility is that it is learned. According to the Rationality Principle (Card, Moran, and Newell 1983), if people are exposed to similar problems, over time they will converge on learning similar ways to deal with the problem. Therefore, the macro cognitive architecture could be created through the developmental process of learning how to do macro tasks in the real world from a young age. However, in either case the result is the same - a system for processing macro cognitive tasks that is similar across individuals. It is also possible that the two processes work together as in language development.

Evolutionary Constraints

Evolutionary Psychologists like to map real world functions, theorized to be evolutionarily significant, to modules in the brain. From a computational point of view this is not very satisfying, as it does not explain how those functions are computed or how they are integrated into the overall system. From the perspective of the macro architecture hypothesis, this approach is misguided as it goes directly from the macro level to the neural level. The neural areas involved should compute micro cognitive functions, not macro cognitive functions. If this is the case, then understanding the relationship between micro and macro cognitive functions is essential to make sense of any neural localization related to macro level behaviors.

The macro architecture hypothesis is also relevant for linking micro cognitive theory to evolutionary theory. Other than rational analysis (Anderson 1990), which is limited to saying that we do things in optimal ways, micro cognition does connect up to evolution. To connect to evolution we need to say which things we have been optimized for. A macro cognitive architecture, in theory, provides the set of real world functions that are brains were evolved to cope with. This also provides important constraints for defining what the macro functions are. For example, it makes no evolutionary sense that we should have a macro function for dealing with computers or the Internet.

SGOMS

SGOMS (West and Nagy 2007) is a macro architecture designed to model expertise in dynamic, multi-agent tasks. From an evolutionary point of view (see section above) this amounts to a claim that the human macro cognitive architecture was evolved, in whole or in part, to allow humans to acquire expertise and to use it in real world, collaborative situations (which seems reasonable).

Most studies of macro cognition can be considered studies of experts operating in the real world. This is probably due to the practical need to produce applied research for different areas, but whatever the reason, the dominant approach in the study of expertise (regardless if the focus is micro or macro) is to treat each domain of expertise separately (see Ericsson et al. 2006). This is also the dominant approach for Cognitive Engineering within Systems Engineering (Kirlik 2012). Most work on expert learning does not consider the possibility of a task independent way of organizing expert knowledge.

In contrast, the SGOMS architecture is based on the claim that all expert behavior is mediated by a fixed set of interacting cognitive structures. SGOMS was created by extending the GOMS modeling system (which is good for modeling uninterrupted, solitary expert tasks) to model expert tasks in dynamic, multi-agent environments. We found that the problem lay in GOMS being unable to handle the frequent interruptions, task switching, and re-planning that occur in real world tasks (West and Nagy 2007; Kieras and Santoro 2004). To fix this we modified the definition of the unit task by adding the criterion that a unit task should be small enough so that it will most likely not be interrupted. That is, we defined the unit task as a control structure that functions to avoid overload, downtime, *and* interruptions. This modification allows the unit task to continue to serve its original function, that is, to define islands of work that can be executed in a well-defined way.

We also added a second control structure, called the planning unit. In SGOMS, the unit task mediates between the micro cognitive level and the macro cognitive level, while the planning unit mediates between the macro cognitive level and the real world. In contrast to unit tasks, planning units are designed for interruptions and task switching. Planning units also allow efficient communication and coordination between agents by functioning as the building blocks for creating plans and modifying them. For example, planning units are theorized to have names that are used in communication to establish common ground (Klein 2004) between agents. Therefore, data related to planning units can be found in the behavior of individuals as well as the interactions between individuals.

The simplest, and we believe, the most common form of planning unit, is an ordered list of unit tasks. If a planning unit is interrupted, the current unit task is either finished or abandoned and the situation is assessed. The task can be resumed or a new planning unit can be chosen based on the current constraints. When a planning unit is interrupted the progress on the planning unit is stored in memory so that it can be resumed. In the SGOMS theory, the highest level of decision-making is the constraint satisfaction process used to choose planning units based on the current context, which is constantly updated. If there is a plan, then that is also part of the context. In addition, each planning unit is associated with a set of constraints. The goal of this system is to allow agents to react locally and independently as much as possible without disrupting coordination between agents (i.e., the plan).

SGOMS has the following hierarchical structure of representations. Each is associated with a different set of cognitive mechanisms:

- Context - constraint based decision making
- Planning units - list or plan, can be modified
- Unit tasks - expert systems, smart but brittle
- Methods - fixed set of actions
- Operators - basic units of perceptual and motor actions
- Bottom up monitoring - when not busy the system checks the environment and memory for relevant information

The level above controls the level below but the resources are shared. So, for example, different planning units can call on the same unit task. Figure 2 shows the cycle of operations. Interruptions can occur at any level and may be solved on any level. For example, unit tasks can solve expected or common interruptions related to that unit task because it is part of the routine process. Only if an interruption percolates to the top does it result in constraint based re-planning.

The SGOMS macro architecture can be used without implementing it on a micro cognitive architecture. Similar to GOMS, models are created by interviewing and observing experts, and filling in the structures above. To evaluate SGOMS models we use the model to track an expert, either live or on video. The model is evaluated in terms of the percent times it predicts the next move correctly. For constraint based decision-making a judgment is made as to whether the model could have reasonably predicted a task switching event given the interruption, the context, and the constraints associated with the planning unit that was chosen. SGOMS predicts that all experts will act in the same way except at the level of choosing planning units, where personal style can also enter (e.g., if an expert is risk averse or not). So far we have found that SGOMS provides good predictions (above 80%) for a variety of tasks (West and Nagy 2007, looked at network maintenance; other examples are more recent and not yet published, they include: team play in a first person shooter video game and professional mediation sessions). The requirement that the architecture must work across different types of expertise creates a much stronger scientific framework for evaluation compared to one-off macro cognitive models.

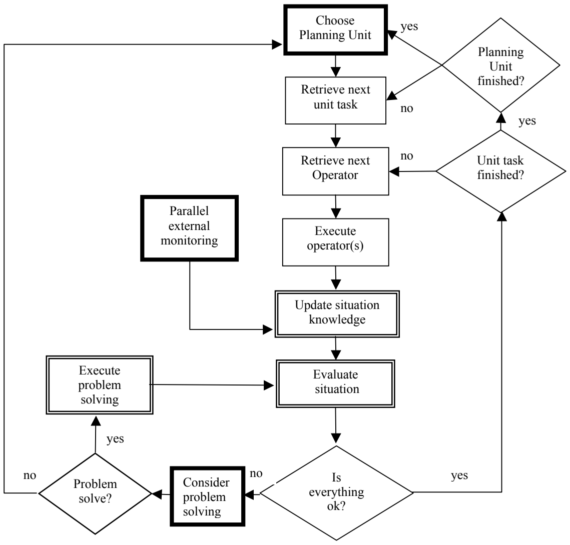

Figure 2. Flow of control for SGOMS

SGOMS:ACT-R

SGOMS:ACT-R is a version of SGOMS implemented in ACT-R (Anderson and Lebiere 1998) which is a well studied and well tested micro cognitive architecture. From an ACT-R perspective, SGOMS:ACT-R represents the hypothesis that the right way to build a model of an expert in ACT-R is by using the SGOMS macro cognitive architecture (see West and Pronovost 2009; West and Somers 2011, for a discussion). Doing this requires every ACT-R model to have a fixed set of dedicated productions that mediate task related productions and declarative memory content stored in a specific format. As outlined above, these dedicated productions could be hard wired or they could be arrived at through developmental learning and the Rationality Principle, or both.

Note, though, that this is not the first attempt to create systems for generating models of macro level tasks using micro level architectures (see Ritter et al. 2006 for a review). In our opinion, this type of endeavor implicitly presupposes some sort of macro architecture. At a minimum, the concept of a macro architecture is a candidate for framing this type of research and making the goals and commitments clearer.

Unit tasks are built in the way they are normally built in ACT-R. That is by using task specific productions that can store and retrieve information in declarative memory, issue motor commands, and retrieve perceptual information. Planning units are constructed in declarative memory and managed by a dedicated set of productions for retrieving the next unit task from the active planning unit. This is similar to the system for learning from instructions in ACT-R. Normally, ACT-R will compile instructions in declarative memory into productions. However, due to the hierarchical relationship between planning units and unit tasks, and also the difference in the time scale, ACT-R will compile within unit tasks but will not compile planning units. This is a really nice result because it's what we needed to happen. Planning units need to be in declarative memory so they can be altered to adapt to special circumstances.

Threading, which is a way to get ACT-R to multi-task by allowing it to have multiple, parallel goals (Salvucci and Taatgen 2008), is used within unit tasks. However, the goals for threading are reset each time the system goes in or out of a unit task. In our opinion, this solves a problem with scaling up the threading mechanism, which we believe would create chaos in the context of a larger, more complex task. We have also extended the representation of the task by having separate buffers for context, planning units, unit tasks, methods, and operators. This is needed for re-use and for dealing with interruptions. Collectively this can be viewed as an expansion of the goal module.

One issue that is difficult to deal with is unexpected interruptions. In the real world, events are happening all the time and any one of them could be important enough to trigger an interruption. We know from research (e.g., the Cocktail Party Phenomena) that, although we are unaware of it, we constantly monitor our environment for events that might require our attention. In terms of neural anatomy, this bottom-up alarm system is generally associated with the amygdala. To model this we use a separate production system that evaluates environmental information gathered when the system is not engaged in top-down directed search. This system can post an alarm in the alarm buffer, where the top down production system normally used in ACT-R can react to it (note, the ACT-R production system can respond to bottom up interruptions, however, our system allows for more sophisticated responses).

Constraint-based decision-making is also an interesting problem. We have taken the position that this is a learned skill and usually task specific. Therefore, this system is constructed in the usual ACT-R way. We have implemented constraints as single productions (usually associated with emergencies), expert systems, and heuristic based memory search.

To test the SGOMS:ACT-R system we are implementing SGOMS models and running them (in Python ACT-R, Stewart and West 2006). So far we have successfully run models of airplane pilots and sandwich bar workers. We have also implemented random problems that occur in the environment and ACT-R already has the ability to model less than perfect perception. We have also created the ability to run multiple, communicating agents in a dynamically changing environment (i.e., the environment changes and agents can independently change the environment). We can already see that getting multiple agents to collaborate on a task under these conditions will throw up lots of challenges. However, following the Lakatosian framework outlined above, as long as SGOMS:ACT-R continues to clarify rather than complicate, we will continue on this path.

Conclusion

We have presented the idea of a macro architecture and described SGOMS as an example of a macro architecture. This approach to research has implications for both macro and micro cognitive modeling. For macro cognition it provides the same advantage that the architecture concept provides for micro cognition - to avoid a meaningless proliferation of descriptive models. For micro cognition it creates a unified and principled way to understand how the components of micro cognition are organized to produce macro cognition. Instead of scaling up micro architectures in ad hoc ways to model different macro level tasks, micro architectures are scaled up to have the functionality needed to model all macro level tasks across all knowledge domains.

However, the larger purpose of this paper is to point out the need for an explicit understanding of the theoretical landscape in which macro cognitive models are built. SGOMS is a particular example of a macro cognitive architecture, but other possibilities exist. Also, SGOMS could be implemented in a different micro cognitive architecture, which would create a different variant of SGOMS. Alternatively, the idea of a micro cognitive architecture can be rejected altogether by having macro cognitive functions computed directly by neural systems, with no intervening systems level (e.g., as in Evolutionary Psychology or Social Neuroscience). It is also possible to keep the idea of a micro cognitive architecture and reject the idea of a macro cognitive architecture (i.e., the status quo). Another possibility for ACT-R and similar architectures is to put macro cognitive functions (e.g., planning, meta cognition, social functions) in modules. Yet another possibility is that macro cognition does not exist outside of distributed cognition (e.g., Hutchins 1995). But, whatever the approach, being clear about the theoretical scaffolding and the larger issues at stake should lead to more systematic research and communication across the field. It should also make research on this topic more understandable to those outside of the field.

References

Anderson, J. R. 1990. *The adaptive character of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. and Lebiere, C. 1998. *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, M. L. 2010. Neural reuse: A fundamental organizational principle of the brain. *Behavioral and brain sciences* , 33 (4): 245-66.

Card, S.; Moran, T.; and Newell, A. 1983. *The Psychology of Human-Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Cooper, R. P. 2007. The Role of Falsification in the Development of Cognitive Architectures: Insights from a Lakatosian Analysis. *Cognitive Science* 31: 509–533

Ericsson, K. A.; Charness, N.; Hoffman, R. R.; and Feltovich, P. J. Eds. 2006. *The Cambridge handbook of expertise and expert performance*. New York: Cambridge University Press.

Hutchins, E. 1995. *Cognition in the Wild*. Cambridge, MA: MIT Press.

Kieras, D., and Santoro P. 2004. Computational GOMS Modeling of a Complex Team Task: Lessons Learned. In the Proceedings of CHI 2004, 97-104. Vienna, Austria: CHI

Klein G. 2002. Year One Final Report: High-Level Cognitive Processes In Field Research. Prepared through participation in the Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009.

Klein, G.; Woods, D. D.; Bradshaw, J. D.; Hoffman, R. R.; and Feltovich, P. J. 2004. Ten challenges for making automation a “team player” in joint human-agent activity. *IEEE: Intelligent Systems*, 91-95.

Kirlik, A. 2012. The Emerging Toolbox of Cognitive Engineering Models. Paper presented at the International Conference on Cognitive Modeling, Berlin, Germany, 13-15 April.

Lakatos, I. 1970. Falsification and the methodology of scientific research programs. In *Criticism and the Growth of Knowledge*, Eds. Lakatos I., and Musgrave A., 91–196. Cambridge, UK: Cambridge University Press.

Melchin, K., and Picard, C. 2008. *Transforming Conflict through Insight*, Toronto, ON: University of Toronto Press, Scholarly Publishing Division.

Newell, A. 1973. You Can’t Play 20 Questions with Nature and Win: Projective Comments on the Papers of this Symposium. In *Visual Information Processing: Proceedings of the 8th Symposium on Cognition*, Ed. W. G. Chase, 283-308. New York: Academic Press.

Newell, A. 1990. *Unified theories of cognition*. Cambridge: Harvard University Press.

Ritter, F. E.; Haynes, S. R.; Cohen, M. A.; Howes, A.; John, B. E., Best, B.; Lebiere, C.; Jones, R. M.; Lewis, R. L.; St. Amant, R.; McBride, S. P.; Urbas, L.; Leuchter, S.; and Vera, A. 2006. High-level behavior representation languages revisited. In Proceedings of the International Conference on Cognitive Modeling, 404-407. Trieste, Italy: International Conference on Cognitive Modeling.

Salvucci, D. 2006. Modeling Driver Behavior in a Cognitive Architecture. *Human Factors*, 48 (2): 362-380.

Salvucci, D. D., and Taatgen, N. A. 2008. Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review,* 115 (1): 101–130

Stewart, T.C., and West, R. L. 2006. Deconstructing and reconstructing ACT-R: Exploring the architectural space. *Cognitive Systems Research*, 8(3): 227-236.

West, R. L., and Leibovitz, D. P., 2012. Understanding each other: Defining a conceptual space for cognitive modeling. In the proceedings of Cognitive Science, 2535-2539. Sapporo, Japan: Cognitive Science.

West, R. L., and Nagy, G. 2007. Using GOMS for Modeling Routine Tasks Within Complex Sociotechnical Systems:Connecting Macrocognitive Models to Microcognition. *Journal of Cognitive Engineering and Decision Making,* 1: 186-211

West, R. L., and Pronovost, S. 2009. Modeling SGOMS in ACT-R: Linking Macro- and Microcognition. *Journal of Cognitive Engineering and Decision Making*, 3 (2): 194-207.

Varela F; Lachaux JP; Rodriguez E; and Martinerie J. 2001. The brainweb: phase synchronization and large-scale integration. *Nat Rev Neurosci*. Apr;2(4): 229-39.

West, R. L., and Somers, S. 2011. Scaling up from Micro Cognition to Macro Cognition: Using SGOMS to build Macro Cognitive Models of Sociotechnical Work in ACT-R. In the proceedings of Cognitive Science, 1788-1793. Boston, Mass: Cognitive Science.