ACT-R versus not-ACT-R: Demonstrating Cross-Domain Validity

Terrence C. Stewart (terry@ccmlab.ca)
Robert L. West (robert_west@carleton.ca)
Institute of Cognitive Science, Carleton University
Ottawa, Ontario, Canada, K1S 5B6

Introduction

The goal of creating a cognitive architecture is to develop a single system that can account for results across all domains (Newell, 1990). ACT-R is currently the most promising candidate in this direction, having been validated in a wide variety of situations. However, Many critics of ACT-R (and computational modeling in general) believe that, with enough tweaking, an ACT-R model could be produced for any experimental observation (see Roberts & Pashler, 2000). To some degree this is can be dealt with by having fixed parameter settings or theories about when the parameter settings vary (Anderson & Lebiere, 1998). However, as argued more completely in (Stewart, 2006), another way to address this problem is to not only demonstrate that ACT-R models fit various observations, but also *that other models do not*. To do this, we need to be able to apply completely different architectures to the same situations as our ACT-R models. Furthermore, we should follow a similar approach for variations on ACT-R itself.

For example, to show that the PG-C learning rule is correct, we need to not only show that it results in predictively accurate models in a variety of situations; we also need to show that an alternate learning rule (such as Q-Learning, or some other Reinforcement Learning strategy) does not. Alternatively, we may determine that a variety of learning rules (over a specified range of parameter settings) all produce equivalently accurate results over a set of tasks. In this case, we can potentially identify the unique aspect that separates accurate models from inaccurate ones. Similar considerations exist for those researchers developing variations on ACT-R modules, such as the spacing effect (Pavlik & Anderson, 2005) or various production weighting schemes (Gray, Schoelles, & Sims, 2005).

Modular Model Creation

To achieve this goal of examining a wide variety of models (both ACT-R-based and non-ACT-R-based), we need to be able to rapidly construct models, and to easily reorganize the basic structure of ACT-R. This can include construction of new modules and buffers to extend ACT-R, or adjusting various fundamental formulae. Python ACT-R (Stewart & West, 2005), which is a re-implementation of ACT-R within the Python programming language, was created to facilitate this. In creating Python ACT-R the goal was to make it as open as possible to modify the ACT-R architecture.

Also, to create experimental environments for the resulting models and to analyze the data, the Carleton Cognitive Modelling Suite was created (Stewart, 2006). This includes tools for the exploration of parameter spaces, the use of equivalence testing rather than correlation or mean-squared-error for model evaluation, and a variety of non-ACT-R systems, including neural networks, reinforcement learning, and genetic algorithms.

All software, including implementations of the spacing effect (Pavlik & Anderson, 2005), production weighting (Gray, Schoelles, & Sims, 2005), the SOS vision system (West, Emond, & Tacoma, 2005), and both Q-Learning and TD-learning for productions (Fu & Anderson, 2004) are freely available at <http://ccmlab.ca/ccmsuite.html>.

References

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

Fu, W-T. & Anderson, J. R. (2004). Extending the Computational Abilities of the Procedural Learning Mechanism in ACT-R. In proceedings of the 26th Annual Conference of the Cognitive Science Society.

Gray, W. D., Schoelles, M. J., & Sims, C. R. (2005). Adapting to the task environment: Explorations in expected value. Cognitive Systems Research, 6(1), 27-40.

Newell, A. (1990) Unified Theories of Cognition. Cambridge, MA: Harvard University Press.

Pavlik, P. I. and Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. Cognitive Science, 29, 559-586.

Roberts, S. and Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. Psychological Review. 107 (2), 358-367.

Stewart, T.C. (2006) Tools and Techniques for Quantitative and Predictive Cognitive Science. 28th Annual Meeting of the Cognitive Science Society.

Stewart, T.C. & West, R. L. (2005) Python ACT-R: A New Implementation and a New Syntax. 12th Annual ACT-R Workshop

Stewart, T.C. & West, R. L. (2006) Deconstructing ACT-R. Seventh International Conference on Cognitive Modelling.

West, R. L., Emond, B., & Tacoma, J. (2005) Simple Object System (SOS) for creating ACT-R environments: A usability test, a test of the perceptual system, and an ACT-R 6 version. Proceedings of the Annual ACT-R Workshop