Evaluation of Cognitive Architectures: A Software Evaluation Perspective

Bilge Say

Cognitive Science Program
Informatics Institute
Middle East Technical University, Ankara, Turkey
bsay@ii.metu.edu.tr

Abstract

In this position paper, the benefits and a possible roadmap for evaluation of cognitive architectures via a perspective of software evaluation is outlined. Development of such architectures is a distinctive software development activity; and their evaluations may benefit from detailed surveys into their usage from a software evaluation point of view.

Introduction

Cognitive architectures[1] have been evaluated from various respects: Gluck and Pew (2005) detail a comprehensive evaluation exercise of cognitive architectures within the Agent-Based Modeling and Behavior Representation (AMBR) model comparison project; Cooper (2006) and Cooper and Shallice (1995) evaluate the theoretical value of methodologies and realizations of cognitive architectures in Newell's sense. There are also less direct evaluations that for example, direct attention to pitfalls of statistical model evaluations (Roberts and Pashler, 2000). To my knowledge, there is no detailed generic evaluation exercise that treats and evaluates cognitive architectures as pieces of software from software engineering perspectives, apart from testing of specific cognitive architectures (Ritter, 1992). Such an exercise could actually act as a supplementary basis for working on more fundamental questions such as developing functional and theoretical evaluation methods for cognitive architectures. Moreover, such an exercise will also be an evaluation by itself, since it will give specific characterizations of how cognitive architectures fare on software evaluation issues, such as learnability, individualization and dealing with modeler errors. Rest of the paper will be dealing with a clarification of this proposal, along with some tentative roadmap onto how to realize it.

[1] I will be using *cognitive architectures* broadly, not necessarily instantiating "Unified Theories of Cognition" (Newell, 1990), and synonymous to architectures for intelligence, not necessarily assuming that relationship between intelligence and cognition is worked out.

Cognitive Architectures as Software

Cognitive architectures *are* pieces of software irrespective of whether they have been designed to do cognitive modeling for cognitive science or act as cognitively-inspired artificial intelligence agents in games, simulations etc. How could we characterize their characteristics as different from other software? And, would such a characterization help us to evaluate them?

One outstanding characteristics in developing cognitive architectures is we do not and will not in near future, have a precise, unambigous, well-defined software requirements specification for a cognitive architecture, unless we understand human cognition better in a unified way. Although Newell's vision to do so is still a valid attempt and a topic of discussion (Anderson and Lebiere, 2003), the whole enterprise of developing unified theories of cognition seems not to be at the point where we can readily export requirements into cognitive architectures in the general sense.

Another implication of the above problem that it seriously affects the validation issues of cognitive architectures, as well as our answers to the question "How do we go about building a cognitive architectures?". In a sufficiently constrained artificial intelligence subtask, such as "How do we build part-of-speech taggers for natural languages?", we can start with general information about various methods and evaluation of such methods, and then boil the discussion down to the details of the specific techniques. In the enterprise of building cognitive architectures, we can give details of the design and usage of several cognitive architectures along with the issues they have, but it seems hard to abstract away from examples and generalize for methods for building and using cognitive architectures. An evaluation exercise from the perspective of software evaluation may help us build this abstraction to a certain degree, as well as giving us ideas about remedying the requirements-validation conflict by drawing opinions from a wide range of cognitive architecture designers and users.

This brings us to another issue, namely, how existing cognitive architectures fare as pieces of well-developed and usable software. Software development is a complex task in itself; the complex issue of building cognitive architectures complicates the issue further. Most existing cognitive architectures (except for commercial attempts) seem to be predominantly used by research teams that actually designed them. Does that affect the usability of cognitive architectures? Again, most cognitive architectures, by definition, are likely to be built and used by interdisciplinary teams of researchers and programmers. Is that the case? If yes, again how does that affect usability? How much good design and implementation practices in software engineering are being complied to? For example, are there clear design specifications? How modular and reusable are cognitive architectures? How does cognitive modularity affect design modularity? To what degree is this reflected in implementation modularity? To even risk being somewhat sarcastic, could it be the case that no cognitive scientist tested the cognitive load of cognitive architectures?

A Proposal for a Comprehensive Survey

Even if we accept the need for one, a survey of the current state of cognitive architectures from a software point of view will not be straightforward. Most existing usability questionnaires are geared towards software dealing well-defined tasks such as word processing. Still, they could be at least partially adapted or inspired from.[2] Existing surveys in related software areas, such as a recent survey on Discrete Event Simulation, could again provide inspirations (Taylor and Robinson, 2006). Clearly, other cognitive architecture evaluation work will provide input. Yet, certainly this will require to be a multistage, interdisciplinary, team project.

Initial stages could encompass a classification of current cognitive architectures on multiple dimensions (commercial vs research, software only vs robotics, cognitive-science-oriented vs artificial-intelligence-oriented, mature vs new), and ensure that an adequate sampling is covered in a comprehensive survey. Multiple survey techniques such as pilot questionnaires, interviews and focus group discussions with both designers and users of selected cognitive arhitectures, preinformed with software evaluation methods, could be the next step. Third step, can actually involve reaching a wider sample of cognitive architecture designers and users with reasonably valid and reliable survey instruments.

[2] As part of an ongoing MSc study in METU, Can Bican and I have been partially using IsoMetrics (Gediga, Hamborg and Duntch, 1999) to measure the atttitudes of cognitive modeling tool users.

Conclusion

In summary, trying to bring a software evaluation perspective to both design-implementation and deployment-usage processes of cognitive architectures may aid in clarification of some issues in the evaluation of cognitive architectures. In addition, this will also result in an evaluation of cognitive architectures as complementary to other perspectives such as cognitive and biological plausibility and statistical evaluation methods. By means of surveying a wide range of cognitive architectures, we could highlight the commonalities and differences between cognitive science oriented approaches and artificial intelligence approaches; this may in turn help the interaction of these two communities.

References

Anderson, J., and Lebiere, C. (2003). The Newell Test for a Theory of Cognition. *Behavioral and Brain Sciences* 26: 587-640.

Cooper, R. (2006). Cognitive Architectures as Lakatosian Research Programs: Two Case Studies. *Philosophical Psychology* 19(2): 199-220.

Cooper, R., and Shallice, T. (1995) Soar and the Case for Unified Theories of Cognition. *Cognition* 55: 115-149.

Gadiga, G.; Hamborg, K.C. and Duntsch, I. 1999. The Isometrics Usability Inventory: An Operationalization of ISO 9241-10 Supporting Summative and Formative Evaluation of Software Systems. *Behaviour and Information Technology.* 18(3):151-164.

Gluck, K. A., and Pew, R.W. eds (2005). *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation and Validation.* New Jersey: Lawrance Erlbaum.

Newell, A. (1990). *Unified Theories of Cognition.* Harvard University Press

Ritter, F. 1992. A Methodology and Software Environment for Testing Process Model's Sequential Predictions with Protocols. Ph. D. Diss., School of Computer Science, Carnegie Mellon University, Pittsburgh.

Roberts, S., and Pashler, H. (2000) How Persuasive is a Good Fit? *Psychological Review* 107 (2): 358-367.

Taylor, S.J.E. and S. Robinson. 2006. So Where to Next? A Survey of the Future for Discrete-Event Simulation. *Journal of Simulation,* 1: 1-6.