

Connecting Cognitive Models to Interact with Human-Computer Interfaces

Farnaz Tehranchi (fjt5064@psu.edu)

Department of Computer Science and Engineering
Penn State, University Park, PA 16802 USA

Frank E. Ritter (frank.ritter@psu.edu)

College of Information Sciences and Technology
Penn State, University Park, PA 16802 USA

Keywords: Cognitive Model; Cognitive Architecture; Human Computer Interface, Interaction.

Introduction

Cognitive modeling goals include understanding and predicting human behavior. In fact, the main objective of cognitive science is understanding the nature of the human mind to develop a model that predicts and explains human behavior. Cognitive architectures are infrastructures for cognitive science theory and provide computational frameworks to execute theories.

ACT-R is an architecture of cognition (Anderson, 2007): a platform that is used to implement cognitive models. ACT-R communicates with the outside world by using modules such as visual perception and motor control. In this paper, we propose a model that will provide an environment for ACT-R to interact with the world. For this purpose, we use EMACS as an interactive text editor that includes extensions to read email, browse the web, and work with spreadsheets (Ritter & Wood, 2005). Both ACT-R and Emacs are written in Lisp. It allows us to extend them and design a bridge between them. This bridge will enable ACT-R to communicate with an interactive environment. Further steps within this work involve expanding the model to be more human-like.

The two main needs for a cognitive model to interact with a task environment are 1) the ability to pass commands, and 2) the ability to access the information on the screen. There are two main approaches for interaction. The first approach to meeting these needs is to use a graphic language library such as MCL, Tcl/Tk, Java, or SL-GMS to define objects and pass them from an interface to simulated hands and eyes (Ritter, Baxter, Jones, & Young, 2000) such as ACT-R/PM (Byrne & Anderson, 1998). The second approach is using the screen's bitmap and transforming the image into objects and symbols that a cognitive model can manipulate. For example, SegMan (St. Amant, Riedel, Ritter, & Reifers, 2005) provides domains in which SegMan applies to support cognitive model-based interaction and evaluation. However, SegMan is not yet user-friendly and easy to extend. It still does not recognize everything on the display.

In addition, there are several other approaches for providing models access to tasks, such as simulating the task in the models' heads, being passed a list of inputs from a program that reads from a file of input sets, and instrumenting a particular task interface.

In this report, we extend a previous attempt to provide ACT-R access to the world (Kim, Ritter, & Koubek, 2006) by enabling ACT-R to interact with the Emacs text editor. In this work, ACT-R can communicate with a task environment by instrumenting a graphical library, in this case the Emacs text editor.

The ESegman Approach

ACT-R has been introduced by Anderson (1993) for implementing cognitive models. In fact, it is an architecture of cognition that makes precise predictions about behaviors. Any ACT-R cognitive model has a loop with three components: the task environment, the model, and the results (see Figure 1). The cognitive model receives its perspective as input from the visual (and auditory) module and outputs actions through the motor module. The cognitive architecture within this process understands the external world this way.

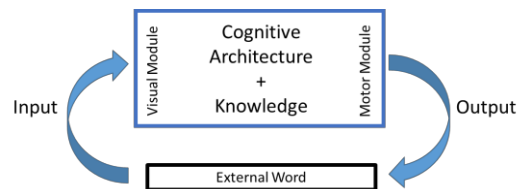


Figure 1: Cognitive model structure.

Figure 2 shows that the first step in the modeling process could be collecting data from the real world (although some would argue for and start at block 3). This process can be performed using the Dismal spreadsheet task. Humans have directly performed the Dismal spreadsheet tasks (Kim & Ritter, 2015), and their performance is recorded using the Recording User Input (RUI) software (Kukreja, Stevenson, & Ritter, 2006; Morgan, Cheng, Pike, & Ritter, 2013).

In particular, for our connection between the architecture and the world, we will use Emacs functions to take the user commands from the main process (ACT-R will be a sub-process of the Emacs process) and insert them into the target buffer. Therefore, the model will be aware of the user commands and can execute them. Whether it is a keystroke command or a mouse action, the model will be capable of using the Emacs Lisp language to execute them. After collecting the commands from ACT-R and receiving requests to 'look' and move the eye, we will feed ACT-R

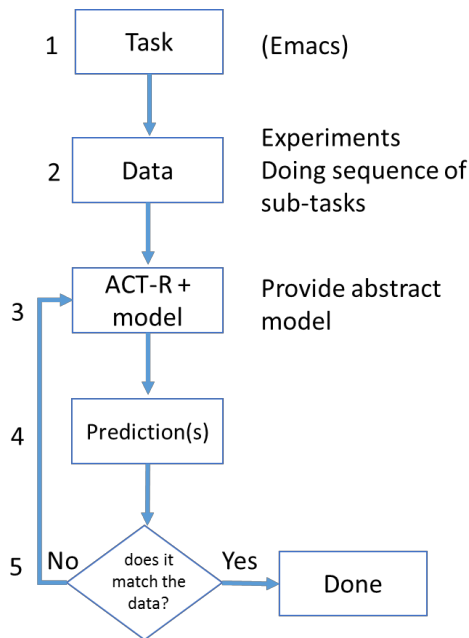


Figure 2: Model process.

with the information about the world by inserting target buffer contexts into its visual module. This implements parts of block 3 in Figure 2 and the connection in Figure 1.

We will also be able to use an existing ACT-R model to interact with Emacs. Paik et al. (2015) created this set of ACT-R models to perform the Dismal spreadsheet task. These models generate the actions to perform the task, but the actions are not implemented in the world—they are simply generated. Enacting them in the world (Emacs in this case) would help test the timing, and with the actions passed in the world, one could turn on errors in the perceptual-motor component in ACT-R and start to explore how errors are generated, noticed, and recovered from.

The main challenge will be the communication step. In particular, ACT-R can load data into Lisp. For sending and receiving data it needs to communicate through the main Lisp application process with Emacs.

In our approach, we are going to load ACT-R within a buffer managed by SLIME, which is an Emacs mode for Common Lisp development. Our objectives will be connecting the ACT-R process with the Emacs that is already managing ACT-R. One of the potentials of Emacs would be allowing an ACT-R model to edit files in the Emacs environment that it is running in.

Another method would be sending a request to an already running Emacs that is waiting for a socket request (using the Swank library), and setting its output as one of the parameters of make-network-process function for opening a client TCP/IP connection.

From a security perspective, we should consider a Swank address that is not wide open to the world because it can make the computer vulnerable to intrusion attacks. However, in our connection scenario, we will not access Swank from another computer. Therefore, establishing a secure channel to the Swank server will not be necessary.

Conclusion and Further Research

Cognitive models have struggled connecting to the world. This approach of connecting through and with Emacs will provide another opportunity for providing models access to more interesting tasks.

Acknowledgments

This work was funded by ONR (N00014-11-1-0275 & N00014-15-1-2275). David Reitter has provided useful comments on Emacs and Aquamacs, (the Emacs version for the Mac). Jong Kim provided the idea.

References

- Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.
- Anderson, J. R. (2007). *How can the human mind exist in the physical universe?* New York, NY: Oxford University Press.
- Byrne, M. D., & Anderson, J. R. (1998). Perception and action. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought*. Mahwah, NJ: Erlbaum.
- Kim, J., Ritter, F. E., & Koubek, R. J. (2006). ESEGMAN: A substrate for ACT-R architecture and an Emacs Lisp application. In *Proceedings of ICCM - 2006- Seventh International Conference on Cognitive Modeling*, 375. Edizioni Goliardiche: Trieste, Italy.
- Kim, J. W., & Ritter, F. E. (2015). Learning, forgetting, and relearning for keystroke- and mouse-driven tasks: Relearning is important. *Human-Computer Interaction*, 30(1), 1-33.
- Kukreja, U., Stevenson, W. E., & Ritter, F. E. (2006). RUI—Recording User Input from interfaces under Windows and Mac OS X. *Behavior Research Methods*, 38(4), 656–659.
- Morgan, J. H., Cheng, C.-Y., Pike, C., & Ritter, F. E. (2013). A design, tests, and considerations for improving keystroke and mouse loggers. *Interacting with Computers*, 25(3), 242-258.
- Paik, J., Kim, J. W., Ritter, F. E., & Reitter, D. (2015). Predicting user performance and learning in human-computer interaction with the Herbal compiler *ACM Transactions on Computer-Human Interaction*, 22(5), Article No.: 25.
- Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Supporting cognitive models as users. *ACM Transactions on Computer-Human Interaction*, 7(2), 141-173.
- Ritter, F. E., & Wood, A. B. (2005). Dismal: A spreadsheet for sequential data analysis and HCI experimentation. *Behavior Research Methods*, 37(1), 71-81.
- St. Amant, R., Riedel, M. O., Ritter, F. E., & Reifers, A. (2005). Image processing in cognitive models with SegMan. In *Proceedings of HCI International '05*, Volume 4 - Theories Models and Processes in HCI. Paper # 1869. Erlbaum: Mahwah, NJ.