

ESEGMAN: A substrate for ACT-R architecture and an Emacs Lisp application

Jong W. Kim (jongkim@psu.edu)

Department of Industrial and Manufacturing Engineering

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

College of Information Sciences and Technology

Richard J. Koubek (rkoubek@psu.edu)

Department of Industrial and Manufacturing Engineering

The Pennsylvania State University, University Park, PA 16802 USA

Introduction

Cognitive architectures provide a framework upon which to build models that emulate human behavior. The model of the user is studied to provide a theoretical and practical understanding of user behaviors and usability of interfaces.

However, there are restrictions placed on the cognitive models regarding access to an external environment. Researchers have studied how to embody a computational cognitive model to interact with a simulated task environment. For example, the Argus system supports an embodied cognitive model interacting with a radar-like target classification task (Schoelles & Gray, 2001; Gray, 2002). In the Argus system, the model and the human subjects use the same interface. It is useful for the development of models including human cognition, human performance or AI agents to have more general access to man-made tasks, task environments, and interfaces.

In this paper, we propose the Emacs substrate system, ESEGMAN (Emacs SubstratE: Gates toward MAN-made world) to help simulate user performance. User performance studies are easy to run with Emacs (including text only web browsing, spreadsheet use, and email use). A model can be connected to the same tasks with a high likelihood of the model being able to see and do the tasks that a user can see and do.

The ESEGMAN World

The ESEGMAN world consists of a cognitive model, a substrate, and a simulated spreadsheet task environment. ESEGMAN instruments the graphic interface system of Emacs. Emacs is an extendable editor that basically functions as an operating system for file editing and related information processing. In the ESEGMAN world, ESEGMAN embodies a cognitive model interacting with a simulated task. This study opens a possibility of a new cognitive modeling paradigm and extends ACT-R's perception and motor capabilities.

Theoretical Backgrounds

Cognitive models generally fail to interact with an external task environment (Ritter et al., 2000). To enable cognitive models to perform interactive tasks, it is necessary for the

models to have visual perception and motor action capabilities. These capabilities allow a cognitive model to perceive what is on the screen and to make some types or mouse movements.

The model's visual perception capabilities should have similar mechanisms to humans. One difference for interactive tasks is that the model's perception in two dimensions is adequate with respect to the interaction with a spreadsheet application in this study.

The model's motor action capabilities should also correspond to the human's motor action mechanisms. A cognitive model would use a mouse or a keyboard when the model interacts with an interface. Possible motor actions include typing a letter, moving a mouse, clicking a mouse button, or moving the eyes, etc.

In general, there are two fundamental approaches to provide models with access to a simulated task environment. One approach is to instrument a graphic language such as MCL, Tcl/Tk, Java, or SL-GMS. For example, this approach is taken by ACT-R/PM (Byrne & Anderson, 1998) and simulated hands and eyes models (Ritter, Baxter, Jones, & Young, 2000). These models know what objects to pass from an interface and how to input simulated user commands.

The other approach is to work with the bitmap taken from the screen and parse the screen into objects. This is very robust approach, once done, because all interfaces within the instrumented language become available to the model. For example, SegMan (St. Amant et al., 2005) provides a fairly robust approach in that it allows any windows interface to be seen by models (e.g., ACT-R or occasional Soar models). SegMan, however, has some limitations. It can be somewhat difficult to use and extend. It does not yet recognize all the objects that people do.

The ESEGMAN World Architecture

As shown in Figure 1, the ESEGMAN world provides a surrounding where a cognitive model subject and human subjects perform spreadsheet tasks in a laboratory setting as part of a study on learning and forgetting.

The model is built on ACT-R 6 cognitive architecture. A model agent interacts with the GNU Emacs spreadsheet called Dismal (Ritter & Wood, 2005). In order for the model

to directly interact with the task environment (Dismal), a substrate (ESEGMAN) represented by the eye and hand in the Figure 1 was implemented.

In the real world, humans can directly perform the Dismal spreadsheet tasks. Human performance is recorded by using RUI denoted by Recording User Input (Kukreja et al., in press).

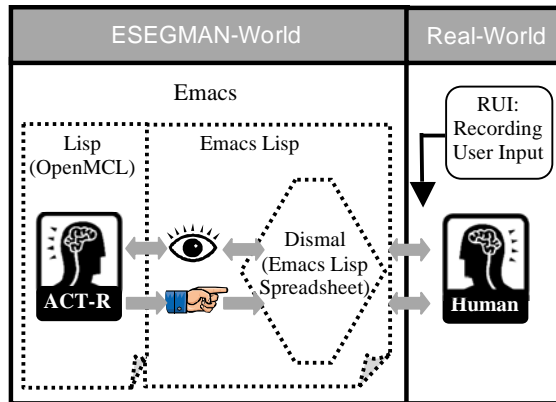


Figure 1: ESEGMAN architecture.

Mechanisms of ESEGMAN

ESEGMAN is layered on the operation of Emacs and allows a model to see and to touch a task environment. Dismal spreadsheet was implemented on the Emacs Lisp language. ESEGMAN is built on both Common Lisp and Emacs Lisp languages. Thus, ESEGMAN provides an ACT-R model with a gate to interact with a man-made world of spreadsheet tasks.

ESEGMAN works in the following way. An Emacs shell process is spawned, and a model is loaded within that process. For example, a shell is started in Emacs to invoke OpenMCL that is a Lisp implementation. Then, ACT-R 6 is loaded into OpenMCL. An ACT-R model can send commands to ESEGMAN, such as to move the mouse, to type a letter, or to get the contents of Dismal as fovea.

In Emacs, there is a set of functions to take the output from the shell and insert them into the associated buffer. This approach allows a natural place for the ESEGMAN to inspect what is sent, and if a command is sent, to execute it.

If the command is to type a letter or to execute a keystroke command, this can be done directly using the extension language of Emacs Lisp. If the command is to move the mouse, a model mouse pointer is moved, probably shown in the mode line of the buffer being used by the model. If the command is to execute a mouse action, the corresponding process as for keystrokes is executed.

When the model wants to look at the screen, ESEGMAN takes the current fovea location and sets up a data structure to be processed and sends this back to the ACT-R model. ACT-R, after sending the fovea look command, has a read that follows the incoming information and puts it into the ACT-R's visual iconic memory. ESEGMAN can create a

file, or it can pass back through the process to an associated buffer.

Conclusion and Further Research

Gray (2002) stated that simulated task environments are one solution to the researcher's dilemma that there is a tradeoff between too much complexity in field research and little complexity in laboratory research. The ESEGMAN world provides a continuum of real spreadsheet tasks between the human and model subjects in a laboratory setting. Also, it provides easy data collection and appears to require less than an hour of training for experiments while providing models complete access to a complex task environment.

Various types of cognitive models under the ACT-R 6 architecture including knowledge acquisition or degradation on a spreadsheet task can be developed and utilized to provide a better understanding of user behavior.

Acknowledgments

Dan Bothell at Carnegie Mellon University provided useful comments on the current ACT-R 6 architecture and its perceptual/motor mechanisms. William Stevenson provided comments that improved this paper.

References

- Byrne, M. D., & Anderson, J. R. (1998). Perception and action. In J. R. Anderson & C. Lebiere (Eds.), *The Atomic Components of Thought* (pp. 167-200). Mahwah, NJ: Lawrence Erlbaum.
- Gray, W. D. (2002). Simulated task environments: The role of high-fidelity simulations, scaled worlds, synthetic environments, and microworlds in basic and applied cognitive research. *Cognitive Science Q.*, 2(2), 205-227.
- Kukreja, U., Stevenson, W. E., & Ritter, F. E. (in press). RUI: Recording user input from interfaces under Window and Mac OS X. *Behavior Research Methods, Instruments, & Computers*.
- Ritter, F. E., & Wood, A. B. (2005). Dismal: A spreadsheet for sequential data analysis and HCI experimentation. *Behavior Research Methods, Instruments, & Computers*, 37(1), 71-81.
- 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.
- Schoelles, M. J., & Gray, W. D. (2001). Argus: A suite of tools for research in complex cognition. *Behavior Research Methods, Instruments, & Computers*, 33(2), 130-140.
- St. Amant, R., Riedl, M. O., Ritter, F. E., & Reifers, A. (2005). *Image processing in cognitive models with SegMan*. Paper presented at the HCI International 2005, Las Vegas, Nevada.