In Proceedings of ICCM - 2007- Eighth International Conference on Cognitive Modeling. 233 - 234. Oxford, UK: Taylor & Francis/Psychology Press.

Dynamic Visualization of ACT-R Declarative Memory Structure

Andrea Heiberg[2] (Andrea.Heiberg@mesa.afmc.af.mil)
Jack Harris[1] (Jack.Harris@mesa.afmc.af.mil)
Jerry T. Ball[1] (Jerry.Ball@mesa.afmc.af.mil)
[1]Air Force Research Laboratory / [2]L-3 Communications at Air Force Research Laboratory
6030 S. Kent St.
Mesa, AZ 85212 USA

Introduction

We propose an automated technique for visualizing changes to declarative memory (DM) in the ACT-R 6 cognitive architecture (Anderson et al., 2004; Anderson & Lebiere, 1998). In this technique, DM chunks and the relationships between chunks are displayed graphically in a labeled tree diagram. A series of diagrams, automatically generated during a model run, allows the modeler to easily visualize how DM changes over time. The technique is potentially useful for any ACT-R model with a complex DM structure.

Labeled Tree Diagrams

Labeled tree diagrams are commonly used in theoretical linguistics (e.g., Radford 1988) to represent constituent structure. The structure of "I increased the airspeed" may be represented as in Figure 1. Top-level SENTENCE contains constituents NOUN-PHRASE and VERB-PHRASE; VERB-PHRASE contains VERB ("increased") and NOUN-PHRASE ("the airspeed"), and so on. Figure 1 was generated from labeled bracket notation with a third-party software tool, phpSyntaxTree (Eisenbach & Eisenbach, 2006).

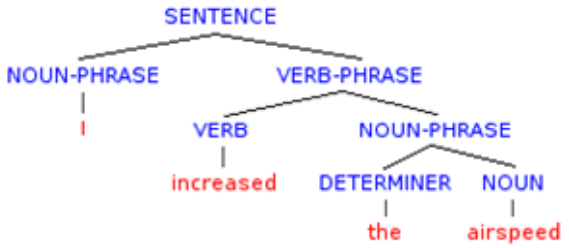

Figure 1 Labeled Tree Diagram

Similar diagrams have long been used for exposition of cognitive models (e.g., Anderson, 1983; Anderson, Budiu, & Reder, 2001).

Declarative Memory Structure Visualization

The automated, dynamic visualization technique is being used in the ACT-R implementation of the Double R model of language comprehension (Ball, 2007; Ball, Heiberg, & Silber, 2007). Figure 2 shows a graphical representation of the final DM structure for "I increased the airspeed". The nodes of the tree are the names of chunks and slots. The tree structure captures the relationships between chunks. For example, chunk PRED-TRANS-VERB (transitive verb predicate) has three constituent slots, SUBJ (subject), HEAD, and OBJ (object); OBJ contains an OBJ-REFER-EXPR (object referring expression) chunk, etc.

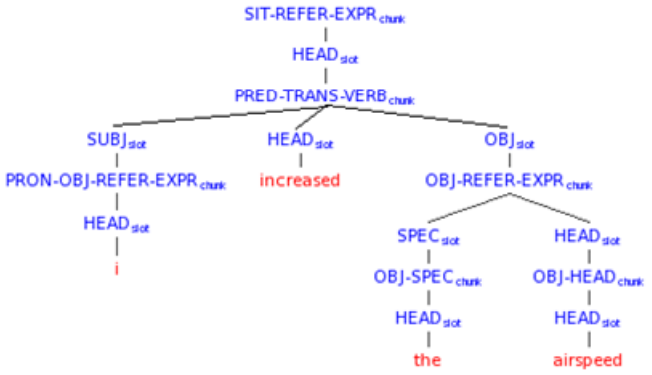

Figure 2 Double R Model DM Structure

The diagrams are also used to visualize changes to DM during the model run. Figure 3 shows a DM snapshot after processing "I"; Figure 4, "increased"; Figure 5, "the"; and Figure 2, the final structure after processing "airspeed".

For the development of the large-scale Double R model, the technique has proven to be greatly more efficient than examining DM by hand. Creating a series of representations takes seconds, as opposed to the minutes required to draw a single diagram by hand.

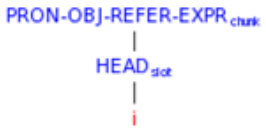

Figure 3 DM Snapshot after Processing "I"

Figure 4 DM Snapshot after Processing "increased"

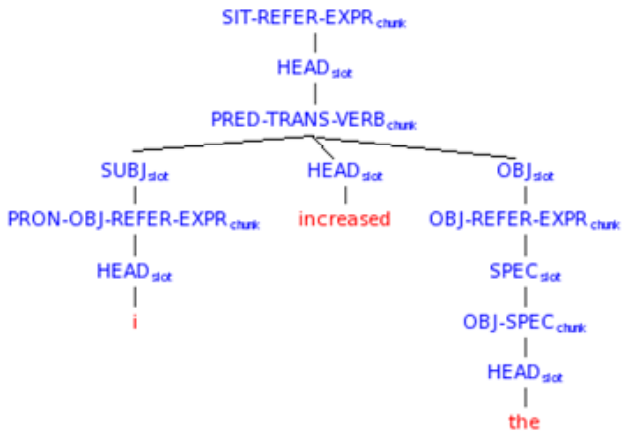

Figure 5 DM Snapshot after Processing "the"

The technique may also be applied to non-linguistic models, to help visualize complex DM structures. An example from the ACT-R 6 tutorial (http://act-r.psy.cmu.edu/actr6) is the Siegler child addition (Siegler & Shrager, 1984) model. The chunks for that model include:

(two isa number value 2 name "two")
(three isa number value 3 name "three")
(five isa number value 5 name "five")
(f23 isa plus-fact addend1 two addend2 three sum five)

Figure 6 shows a graphical representation of chunk f23:

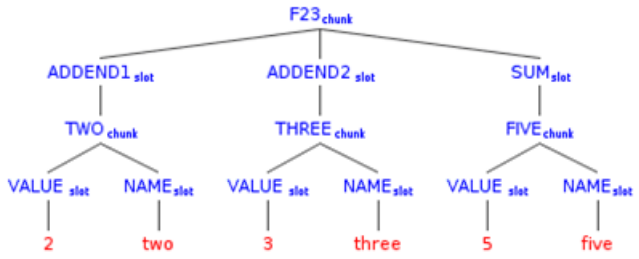

Figure 6 Siegler Model DM Structure

Implementation

During a model run, snapshots of DM are created by invoking image generation from ACT-R production rules. DM is traversed from a starting chunk; slots and chunks are recursively examined to produce a labeled bracket representation, which is then input to an image generator (phpSyntaxTree) that is integrated with the system. The code is written in Lisp; ACT-R 6 functions are used to traverse DM. The implementation is generic, and may be used with any ACT-R 6 model.

A DM chunk may ultimately refer to itself. To avoid infinite processing, traversal stops at any previously visited chunk. For example, in the communication model (Matessa, 1999; Matessa & Anderson, 2000) shown in Figure 7, chunk C5 appears at the top of the tree and in the BELOW slot of chunk C6. However, C5 is expanded only once.

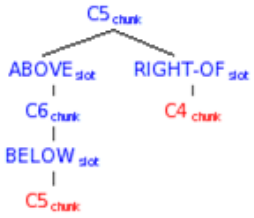

Figure 7 Communication Model DM Structure

Summary

The automated, dynamic visualization technique proposed here may be used to help understand the DM structure of an ACT-R model. Relationships between chunks are displayed graphically in a labeled tree diagram. A series of diagrams is automatically created during a model run to show how DM changes over time. The technique has proven to be particularly useful for the development and exposition of a large-scale model. The implementation of the technique is general, and so may be used with any ACT-R 6 model. The clear view of DM provided by the technique helps make assumptions about a model explicit; it is hoped that this will help provide a better understanding of cognitive modeling.

Acknowledgments

This research is funded by the Warfighter Readiness Research Division of the Air Force Research Laboratory.

References

Anderson, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C. & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review* 111 (4), 1036-1060.

Anderson, J. R., Budiu, R. & Reder, L. M. (2001). A theory of sentence memory as part of a general theory of memory. *Journal of Memory and Language* 45, 337-367.

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

Ball, J. (2007). Construction driven language processing. In *Proceedings of the Second European Cognitive Science Conference.*

Ball, J., Heiberg, A. & Silber, R. (2007). Toward a large-scale model of language comprehension in ACT-R 6. In *Proceedings of the Eighth ICCM.*

Eisenbach, A. & Eisenbach, M. (2006). phpSyntaxTree tool, http://ironcreek.net/phpsyntaxtree.

Matessa, M. P. (1999). Communication in collaborative problem solving. Ms.

Matessa, M. & Anderson, J. R. (2000). An ACT-R model of adaptive communication. In *Proceedings of the Third ICCM*, 210-217, University of Groningen, Netherlands.

Radford, A. (1988). *Transformational Grammar: A First Course*. Cambridge: Cambridge University Press.

Siegler, R. S. & Shrager, J. (1984). Strategy choices in addition and subtraction: How do children know what to do? In C. Sophian (ed.), *Origins of Cognitive Skills*. Hillsdale, NJ: Erlbaum.