15TH ANNUAL
ACT-R WORKSHOP

JULY 18 - 20, 2008

ADAMSON WING
136 A - BAKER HALL

OFFICE OF NAVAL RESEARCH S&T

ONR

Revolutionary Research . . . Relevant Results

FRIDAY, JULY 18

08:00 am Continental Breakfast
08:45 am Welcome
09:00 am The Brain

John Anderson + Angela Brunstein: Learning algebra by discovery: What's ACT got to do with it?
Jelmer Borst, Niels Taatgen, Andrea Stocco, + Hedderik van Rijn: Locating the problem representation bottleneck in the brain.
Daniel Cassenti: Bringing EEG data into the ACT-R fold.
Terrence Stewart + Chris Eliasmith: Implementing the ACT-R production system in spiking neurons.
Andrea Stocco, Christian Lebiere, + John Anderson: Taking the procedural module seriously: A neural model of selection, execution, and learning in the basal ganglia.

10:40 am Break
11:00 am Declarative Memory

Ion Juvina + Niels Taatgen: How do we ignore irrelevant information presented on displays?
Leendert van Maanen: An integrated model of sequential sampling.
Martin Greaves + Richard Young: A modular approach to modeling cognitive processes: Examining the encoding and recall of items in updating working memory.
Jong Kim, Frank Ritter, + Richard Koubek: Explorations of the ACT-R architecture for learning and forgetting performance.

12:20 pm LUNCH
01:30 pm Architecture

Glenn Gunzelmann: Confronting architectural drift in ACT-R.
Frank Ritter, Michael Schoelles, Sue Kase, + Laura Cousino Klein: Simulating pre-task appraisal of serial subtraction.
Christan Lebiere + Bradley Best: Architectural support for adversarial behavior.
Nele Pape + Leon Urbas: The influence of task demands in a model of time estimation.

02:50 pm BREAK
03:10 pm Invited Speaker

John Laird: Cognitive Architecture: Past, Present, and Future.

04:10 pm *John Anderson:* Comments on Cognitive Architecture: Past, Present, and Future.
04:40 pm Open Discussion
06:00 pm ACT-R Workshop Dinner (see back for directions)

SATURDAY, JULY 19

08:00 am Continental Breakfast
08:45 am Welcome
09:00 am Utility Learning and Decision Making

Erik Altmann: Short-term decay of production values for cognitive control.
Christian Janssen, Wayne Gray, + Michael Schoelles: How a modeler's conception of rewards influences a model's behavior: investigating ACT-R 6's utility learning mechanism.
Varun Dutt + Cleotilde Gonzalez: Instance and strategy ACT-R models of choice in a dynamic control task: a model comparison story.
Michael Schoelles, Wayne Gray, + Hansjörg Neth: The Sudoku model: a dynamic decision maker.
Danilo Fum + Antonio Napoli: Putting new wine into old bottles: on the role of markers, instances and utilities in the Iowa gambling task.

10:40 am BREAK
11:00 am Human Computer Interaction

Dario Salvucci: Using ACT-R for rapid prototyping and evaluation of in-vehicle interfaces.
Bonnie John: Making ACT-R typewrite right.
Leonghwee Teo + Bonnie John: Toward a tool for predicting goal-directed exploratory behavior.
Robert West: Building an SGOMS model (Sociotechnical GOMS) using ACT-R: Issues with cognitive modelling and macro cognition.

12:20 pm LUNCH (PROVIDED - in basement of Baker Hall in Coffee Lounge)
01:30 pm Visual Perception + Skill Acquisition

Scott Douglass: ACT-R's answers to six questions about visual routines.
Niels Taatgen + Daniel Dickison: Modeling eye-movement patterns in the Flight Management Task: combining bottom-up and top-down vision.
Matthew Walsh + John Anderson: Statistical learning and anticipatory start-point selection.
Clayton Stanley + Michael Byrne: Processes influencing visual search efficiency in conjunctive search: a rational analysis approach.

02:50 pm BREAK
03:10 pm Symposium

Kevin Gluck, Sue Kase, Glenn Gunzelmann + Brad Best: Large-scale computing resources and ACT-R modeling.

04:30 pm Discussion
The future of ACT-R and the ACT-R workshop

SUNDAY, JULY 20

08:00 am Continental breakfast
08:45 am Welcome
09:00 am Language + Integration

Marc Destefano: The development of Lisp bindings for the D-bus interprocess communication system.
Jerry Ball: Modeling long-distance dependencies in double R language.
Markus Guhe + Ellen Gurman Bard: Adapting the use of attributes to the task environment in joint action: results and a model.
David Reitter, Frank Keller, + Johanna Moore: Structural priming in language production emerging from learning in an ACT-R model.
Mike Matessa: HBA: integrating task network modeling and ACT-R.

10:40 am BREAK
11:00 am Robotics + Theory of the Mind

Gregory Trafton, Magdalena Bugajska, William Kennedy, Anthony Harrison, Benjamin Fransen, + Raj Ratwani: ACT-R/E: E for Embodied.
William Kennedy, Magdalena Bugajska, Anthony Harrison, + Gregory Trafton: Simulation within ACT-R as a theory of mind.
Anthony Harrison, William Kennedy, Benjamin Fransen, + Gregory Trafton: Exploring theory-of-mind components within embodied robotics.
Eric Avery + Troy Kelley: ACT-R on a robot: considerations and extensions.

12:20 pm ADJOURN

THE **BRAIN**

Learning algebra by discovery: What's ACT got to do with it?

John R. Anderson
Angela Brunstein

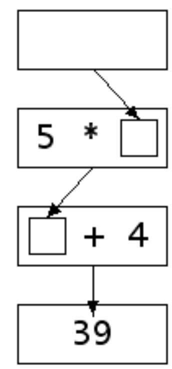
Word Problem (the notation of millennia):

A number is multiplied by 5 and 4 is added to the product. If the result is 39 what was the original number?

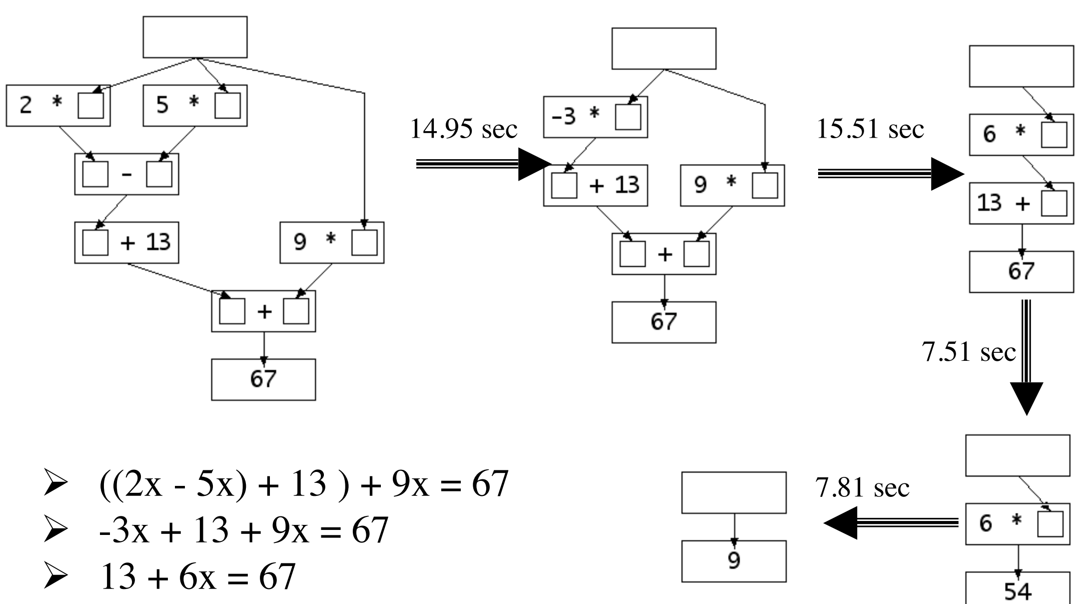
Standard Equation (the notation of Descartes)

$5x + 4 = 39$

Data Flow (CMU notation for the new millennium)

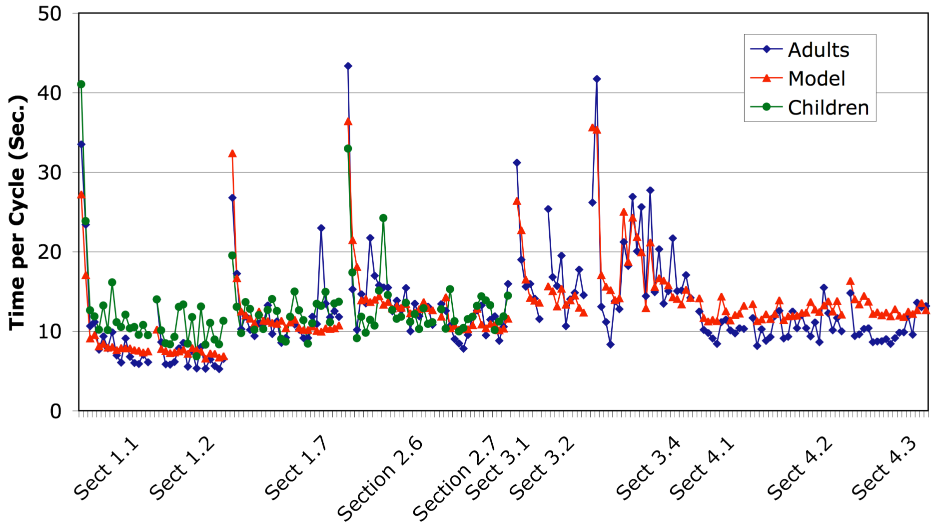


Solving Algebra with an Untrained Eye



- $((2x - 5x) + 13) + 9x = 67$
- $-3x + 13 + 9x = 67$
- $13 + 6x = 67$
- $6x = 54$
- $x = 9$

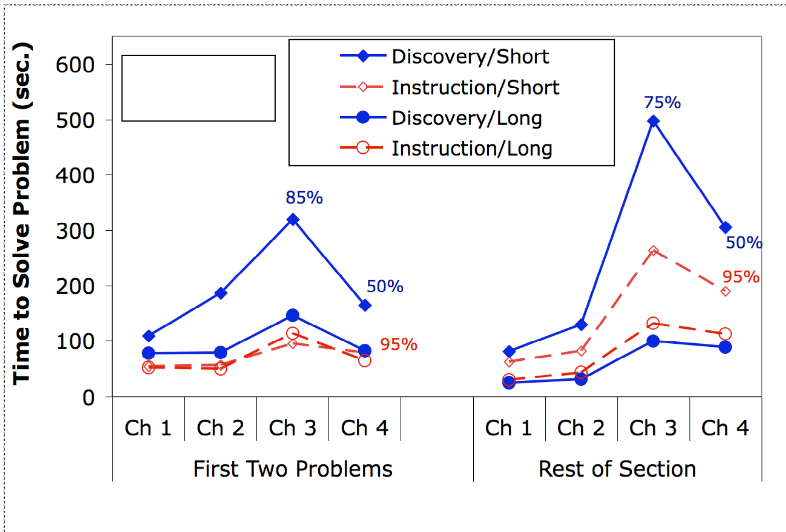
Fit to 181 Problems from Foerster



Effects of Instruction and Practice

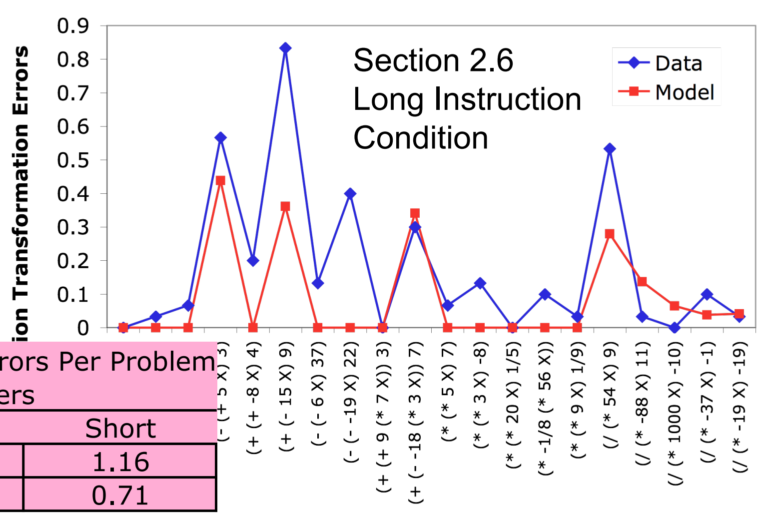
- **2x2 Design**
- **Factor 1:**
  a) **Instruction:** Direction on first one or two problems in a section and guidance available on request for rest.
  b) **Discovery:** No instruction and have to guess transformations. Told if guessed transformations are correct on first one or two problems and whether final answer is correct on rest.
- **Factor 2:**
  a) **Long:** All odd problems (174) for 4 chapters.
  b) **Short:** Subset of 45 that maintains first two and about two others per section Long instruction condition basically replicates earlier study
- Long instruction condition basically replicates earlier study

Time to Solve Problems



➢Half of Subjects dropped out in Short Discovery Condition
➢Subjects in short instruction condition are becoming help abusers
➢Long Discovery students are better than Long Instruction students on later problems
➢Total time for 174 problems (long condition) is 193 minutes for discovery vs. 226 minutes in instruction -- a significant effect.

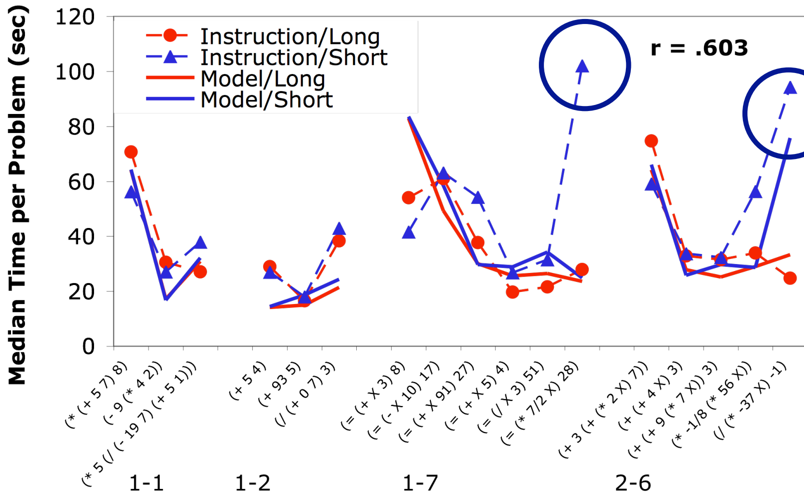
Failure to Semantically Constrain Transformations



Average Transformation Errors Per Problem Over 4 Chapters		
	Long	Short
Instruction	0.37	1.16
Discovery	0.12	0.71

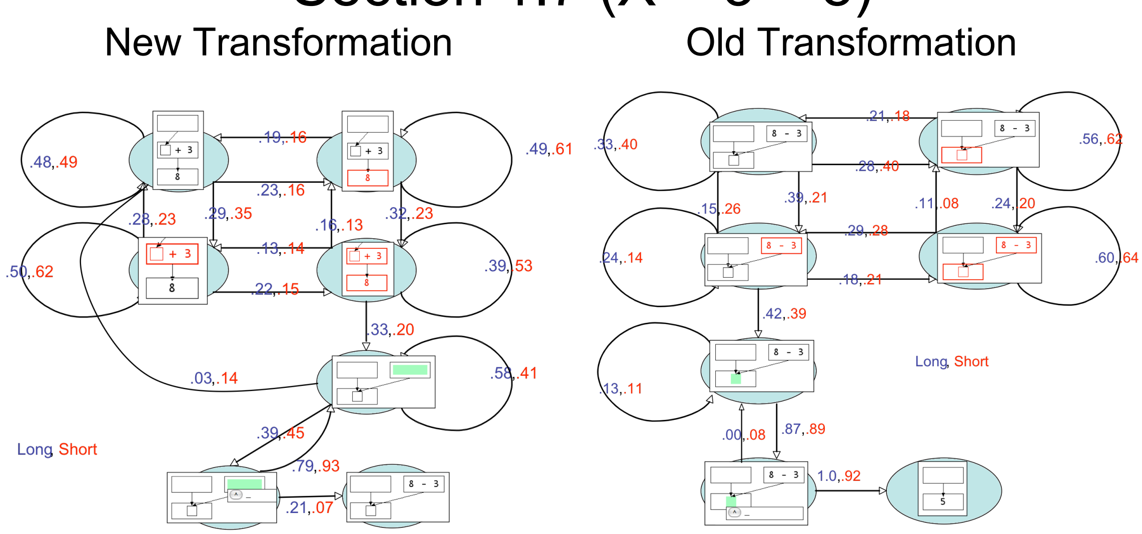
➢Discovery has .01 transformation errors per problem; instruction .20 errors
➢ "(5 + x) – 3" --> 11 of "(5 - 3) – x", 2 of "(5 + 3) – x", 1 of "(5 + 3) + x"
➢ "(15 - x) + 9" --> 12 of "(15 - 9) + x", 5 of (15 - 9) – x", 3 of "(15 + 9) + x"
➢ "(54 * x) / 9" --> 5 of "(54*9) / x", 2 of "(54 / 9) / x", 1 of "(9 / 54) * x"
➢Common feature is preserving main operator
➢General problem of Misinterpretation

Simulating Instruction Condition



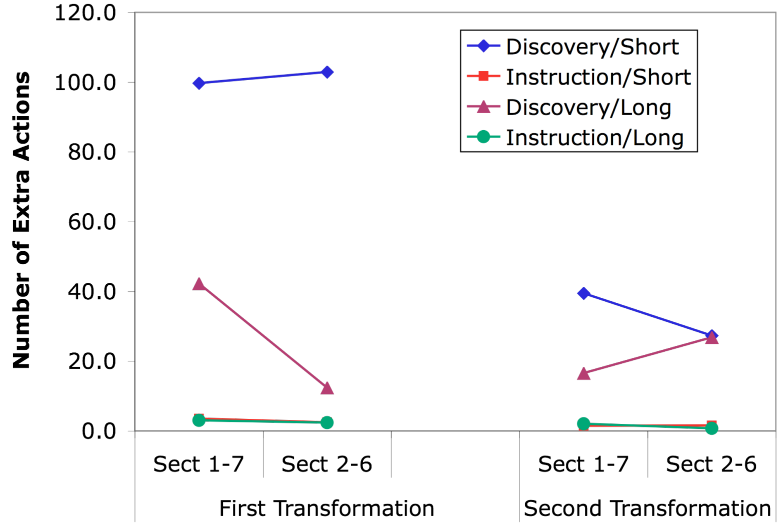
➢Buggy instruction interpretation rules were added that complete with correct rules on a 50-50 basis.
➢In short condition have their first encounter with entering a fraction on last problem in section 1-7. Model starts out knowing how to do it.
➢Last problem of 2-6 is the first opportunity to debug the misconception about the main operator in the short condition.
➢In general short instruction condition is at a disadvantage because there are not enough problems to debug misconceptions.

Discovering New Transformation in a Section 1.7 (X + 3 = 8)



➢Subjects are basically wandering around the search space until they stumble onto the transformation.
➢In the long condition once they have discovered the transformation they don't have problems on later problems.

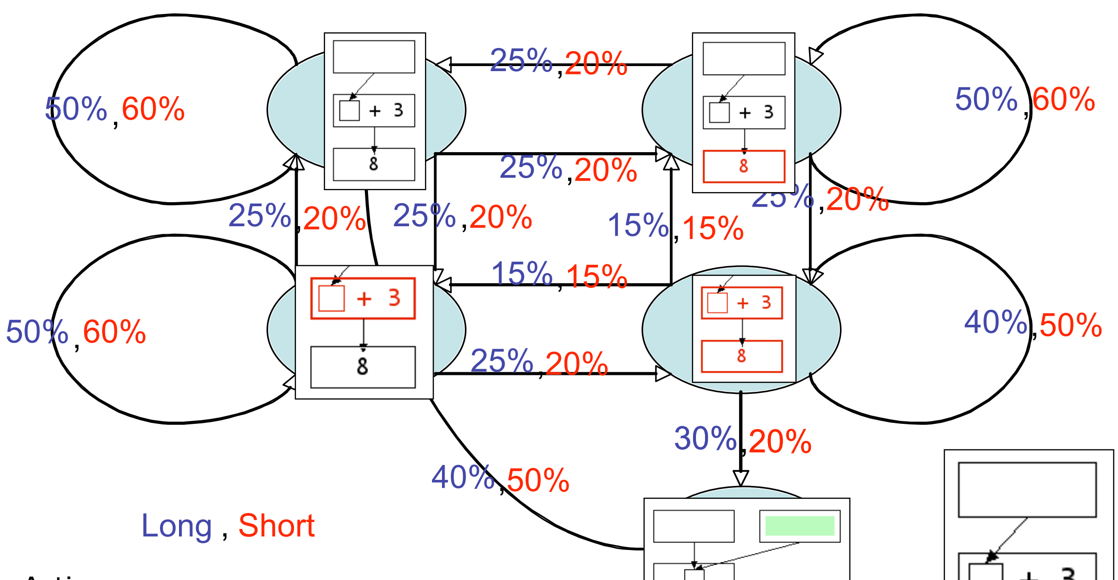
Exp 1&2: First Problems in Section



➢Interaction between transformation, practice and instruction: $F(1,76) = 11.54$; $p < .005$

➢Subjects are showing greater difficulty in the short discovery condition on the first transformation which is new than on the second transformation which is old.

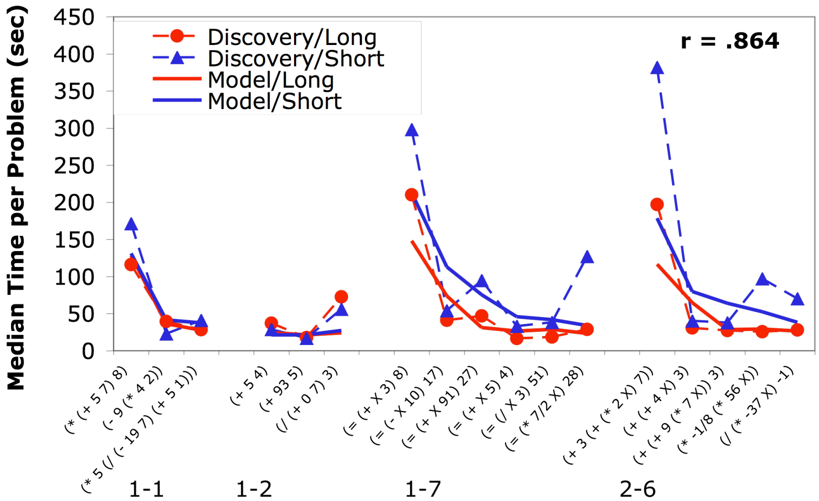
Random Search for Discovery



Actions

	Expected	Observed
Long	34.7	32.1
Short	65.6	63.5

Simulating Discovery



➢Because searches are longer in short condition model cannot often cannot recall correct sequence.
➢As a consequence the model often has to search on later problems and there are not enough problems to always learn the operators.
➢More generally, discovery learning is effective learning is effective as long as the discovery episodes do not get too complex to learn from.

Question: Where is ACT-R?

➢Of course, ACT-R allows us to model the data but where are the ACT-R assumptions contributing to an explanation?

➢The problem with the instruction condition was modeled by programming in misinterpretations.

➢Short instruction is worse because there are not enough examples to debug misinterpretations.

➢Discovery condition at advantage because correct interpretation of discovery episodes programmed in.

➢Short discovery at a disadvantage because similarities are set to make guesses poorer -- this is never really explained.

➢Base-level activation is dropping memory for steps below retrieval threshold in short discovery condition.

Locating the Problem Representation Bottleneck in the Brain

Jelmer Borst (jpborst@ai.rug.nl)[1,2]
Niels Taatgen (taatgen@cmu.edu)[1,2]
Andrea Stocco (stocco@cmu.edu)[2]
Hedderik van Rijn (d.h.van.rijn@rug.nl)[1]

[1]Department of Artificial Intelligence, University of Groningen, The Netherlands
[2]Department of Psychology, Carnegie Mellon University, The Netherlands

The Problem Representation Bottleneck

In their theory of multitasking, Threaded Cognition, Salvucci and Taatgen (2008) presented evidence for two central cognitive bottlenecks: declarative memory and procedural memory. However, in combination with ACT-R (Anderson, 2007), Threaded Cognition suggests a third resource that can act as a bottleneck: the problem representation resource. The problem representation resource is used for mentally maintaining information that is necessary for performing a task. This information is typically not present in the world and often constitutes an intermediate solution to a problem. For instance, if one has to solve a problem like $2x + 5 = 10$, the intermediate step, $2x = 5$, would be stored as the problem representation (e.g., Anderson, 2007). According to ACT-R, the problem representation resource (the imaginal buffer) can hold only one piece of information concurrently. This would mean that if two tasks need to use the problem representation resource at the same time, this would result in interference. In two experiments we have shown that the problem representation indeed acts as a cognitive bottleneck (Borst & Taatgen, 2007; Borst, Taatgen, & Van Rijn, submitted). Cognitive models were developed to account for these results, showing that a problem representation bottleneck can indeed explain the human data.

In the current research, we used functional Magnetic Resonance Imaging (fMRI) to test our model of the problem representation bottleneck further. First, we let our existing ACT-R model make a priori brain activation predictions. Afterwards we tested these predictions in the scanner.

Experiment & Model Predictions

The experiment existed of two tasks that had to be performed concurrently: subtraction and text entry. Both tasks were presented in two versions: an easy version in which maintaining a problem representation was not required and a hard version in which it was. The behavioral data shows a significant interaction effect of subtraction difficulty and text entry difficulty on the reaction times, comparable to the results in Borst et al. (submitted).

We used our existing model of the task to generate brain activation predictions. Basically, when an ACT-R module is active, it will produce a hemodynamic response in the brain region associated with it (for details, see Anderson, 2007). As the model explains the interference effects by using a problem representation bottleneck, we were most interested in the left parietal cortex (Talairach coordinates x = -23, y = -64, z = 34), the region associated with the problem representation resource (Anderson, 2007). The model predicted a strong interaction effect in this region: no activity in the easy subtraction / easy text entry condition, some activation in the easy/hard and hard/easy conditions (the problem representation is then involved in one of the tasks), and very strong activation in the hard/hard condition (the problem representation has to be swapped out constantly, see Borst et al., submitted). The model also predicted activation in five other important regions.

Results

With respect to the problem representation resource, the results were mixed. First of all, the raw fMRI data does not show the predicted interaction effect. The main reason for this is that in the easy/easy condition – where no activation was predicted – we did find high activation levels. However, the problem representation region is known to follow activation in the fusiform gyrus (associated with the visual module in ACT-R). If we discount the visual part of the activation in the problem representation region, so that the activation in the easy/easy condition approaches 0, the activation is reasonably similar to our a priori model predictions (R^2 = .64).

Of the other five regions, the visual, motor and goal areas showed a reasonably good fit. The procedural area was hard to interpret, that is, we did not have strong predictions, and the results are not showing clear effects either. The prefrontal cortex (declarative memory) shows two groups of results: 4 participants showed negative activation, unlike our model, the other 5 participants showed positive activation, showing a similar pattern as the model (R^2 = .76).

References

Anderson, J.R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York: Oxford University Press.

Borst, J.P., & Taatgen, N.A. (2007). The Costs of Multitasking in Threaded Cognition. In *Proceedings of the Eighth International Conference on Cognitive Modeling* (pp. 133-138). Ann Arbor, USA.

Borst, J.P., Taatgen, N.A., & Van Rijn, D.H. (submitted). Problem Representations in Multitasking: An Additional Cognitive Bottleneck.

Salvucci, D.D., & Taatgen, N.A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychol Rev, 115*(1), 101-130.

Bringing EEG Data into the ACT-R Fold

Daniel N. Cassenti (dcassenti@arl.army.mil)
U.S. Army Research Laboratory, AMSRD-ARL-HR-SE, Aberdeen Proving Ground, MD 21005

The ideal for ACT-R is to become a unified theory of cognition (Anderson & Lebiere, 1998) or a system that explains all mental phenomena (see Newell, 1990). ACT-R has a long way to go before it can become a unified theory of cognition, but an initial effort to model neurological phenomena in ACT-R (Sohn, Ursu, Anderson, Stenger, & Carter, 2000) has brought the modeling system much closer to the goal. Neurological phenomena belong to another class of research that is meant to understand cognition – cognitive neuroscience.

The effort by Sohn et al. (2000) represented fMRI (functional magnetic resonance imaging), one of the two most widely used types of neurological measures. The other major source of neurological information comes from electroencephalography (EEG), a practice that collects electro-magnetic data from sensors placed on the scalp. Whereas fMRI provides spatial precision to detect which brain regions are active during task performance, EEG provides temporal precision to determine when brain regions are active. The reverse is also true; fMRI represents task-level activity because fMRI has little temporal resolution, whereas EEG represents large portions of the brain because EEG has little spatial resolution.

Anderson (2007) used fMRI data to assign brain regions to mechanisms, modules, and buffers in the ACT-R cognitive modeling system. Introducing EEG modeling to ACT-R may also have a large impact on ACT-R modeling. In typical cognitive experimentation, two explicit temporal markers appear: onset of stimulus and response. EEG provides temporal markers between these end points. The N100 (a negative charge typically between 100 and 200 ms after stimulus onset) and the P300 (a positive charge typically between 300 and 400 ms after stimulus onset) are two events found during most EEG studies. Although there is some discord in the literature, the N100 is generally considered to mark perceptual encoding of a stimulus while the P300 marks context updating (e.g., classifying the frequency of an event with events in memory).

Two EEG experiments were modeled in ACT-R. Experiment 1 was a simple visual experiment and Experiment 2 was a simple auditory experiment. Instead of being limited by the stimulus onset and response markers, the model was also constrained by the N100. The perceptual encoding portion of the ACT-R models was given the mean and standard deviation parameters from the data of both experiments. The P300 had a different type of effect. The experiments revealed that the P300 often occurs after the response. As such, the context updating that the P300 marks was modeled as a parallel process that is enacted at the same time as the model executes toward the response.

By incorporating EEG into ACT-R, the studies show that the time course of mental processes can have more specificity than with traditional cognitive research. This adds to the capabilities of ACT-R researchers by giving them additional information on timing of mental processes.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Anderson, J. R. & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Newell (1990). *Unified theories of cognition*. Cambridge, MA: Cambridge University Press.

Sohn, M.-H., Ursu, S., Anderson, J. R., Stenger, V. A., & Carter, C. S. (2000). The role of prefrontal cortex and posterior parietal cortex in task-switching. In *Proceedings National Academy of Science*, 97 (24), 13448-13453.

Supplementing Neural Modelling with ACT-R

Terrence C. Stewart (tcstewar@uwaterloo.ca)
Bryan Tripp (bptripp@uwaterloo.ca)
Chris Eliasmith (celiasmith@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo
200 University Ave West, Waterloo, ON N2L 3G1 Canada

There has been significant progress in recent years where modern neuroscience has informed the further development of ACT-R. Modules have been mapped on to brain regions, and detailed fMRI evidence has been gathered. Given this state of affairs, we believe ACT-R can now, in turn, inform neuroscience research.

A common limitation of lower-level neural circuit models is that they lack a realistic environment with which to interact. This limitation is less serious for models of peripheral sensory and motor networks (e.g. a digital image is a reasonable model of bottom-up input to V1). However, models of more central areas, such as prefrontal cortex or basal ganglia, would benefit a great deal from improved models of their connections with surrounding neural systems. From a neural modelling perspective, ACT-R can essentially provide a realistic, interactive test harness.

Neural Modelling

To achieve this, we have integrated a Python implementation of ACT-R (Stewart & West, 2007) with the Nengo neural simulator <http://nengo.ca>. This allows ACT-R models to interact with models of groups of spiking neurons. Nengo supports neuron models as simple as the standard Leaky-Integrate-and-Fire (LIF) model, as well as more detailed conductance-based models. Neural properties such as the post-synaptic time constant and refractory period can be set to match empirically-determined properties of various cell types.

Importantly, Nengo makes use of the Neural Engineering Framework (NEF; Eliasmith & Anderson, 2003). This framework provides an integrated theory of population coding and network dynamics, spanning (1) distributed representation of scalars, vectors, and functions by spike patterns in groups of neurons, (2) transformation of represented values via synaptic connection weights, and (3) a control-theoretic perspective on the way in which these transformations determine network dynamics. This approach has been used to model a variety of sensorimotor systems and cognitive models, including working memory (Singh & Eliasmith, 2006) and the Wason card selection task (Eliasmith, 2005).

In NEF, any group of neurons can be interpreted as representing a vector of values. This vector can be of arbitrary length, and is generally less than the number of neurons in the group. This means that each neuron does not represent a single number in the vector. Instead, the neural spiking patterns form a distributed representation. Increasing the number of neurons improves accuracy and provides robustness to neuron death.

Integrating ACT-R

We wish to use ACT-R to model the various parts of the brain involved in the experimental task, but which are not being modelled neurally. To do this, we need to provide a mapping between neural states and ACT-R states. Since the neural groups can represent vectors and ACT-R states are normally in terms of chunks, we need to transform chunks into numerical vectors and back again. Our current system allows arbitrary mappings, including simple associations between vectors and slot values, and more complex Vector Symbolic Architectures where arbitrary symbol trees can be transformed to a fixed-length vector.

Demonstration

To test this process, we have built a model of an ACT-R goal buffer using 300 LIF neurons. This buffer is based on the NEF working memory model, which can be thought of as a neural integrator, which accumulates and stores input over time. The non-neural portion of the model consists of standard ACT-R with three productions. These simply change the goal buffer so that the model continually cycles between three goal states. Figure 1 shows the spike patterns for the 300 neurons in the goal buffer over 500msec, along with the NEF decoded state information.

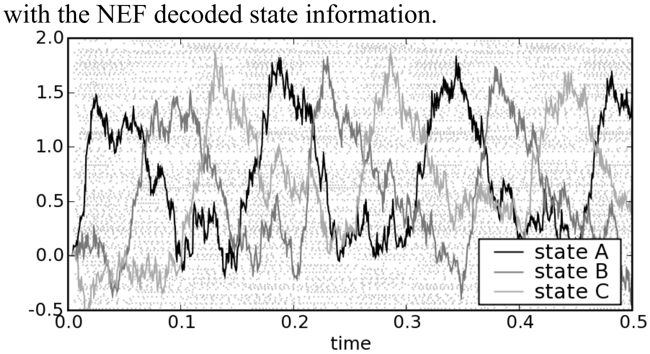


Figure 1: Activity of the 300 LIF neurons in the goal buffer.

References

Eliasmith, C. (2005). *Cognition with neurons: A large-scale, biologically realistic model of the Wason task.* 27th Annual Meeting of the Cognitive Science Society.

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation and dynamics in neurobiological systems.* Cambridge, MA: MIT Press.

Singh, R., & Eliasmith, C. (2006). Higher-dimensional neurons explain the tuning and dynamics of working memory cells. *Journal of Neuroscience*, 26, 3667-3678.

Stewart, T.C., & West, R.L. (2007) Deconstructing and Reconstructing ACT-R. *Cognitive Systems Research.* 8(3), 227-236.

Implementing the ACT-R Production System in Spiking Neurons

Terrence C. Stewart (tcstewar@uwaterloo.ca)
Chris Eliasmith (celiasmith@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo
200 University Ave West, Waterloo, ON N2L 3G1 Canada

There is a strong consensus that the procedural memory component of ACT-R can be identified with specific areas of the basal ganglia. This system must be capable of collecting input (buffer states) from various regions of the cortex, identifying which productions match, selecting one of the matching productions, and then communicating the effects of that production to the relevant brain regions. We present here a method for efficiently implementing this system using spiking neurons. This approach also identifies biological limitations to such a system which limit both the complexity of the production matching rule and the types of patterns that can be matched.

Representing Chunks in Neural Firing Patterns

We base our model on the Neural Engineering Framework (NEF; Eliasmith & Anderson, 2003), which provides an optimal method for representing and manipulating vectors using realistic neural populations. To encode a chunks as vectors, we use Vector Symbolic Architectures (VSAs), a family of techniques for reliably transforming arbitrarily complex symbolic trees into fixed length vectors (see Gayler, 2003 for an overview).

We assume neural projections from various cortical areas provide a representation of buffer contents and module states. The accuracy of these representations depends on the length of the vector, number of neurons, and the neural properties. For example, different neurotransmitters have different re-uptake rates, affecting the time course of the post-synaptic current, and thus how quickly the encoded representation can change.

Production Matching

Given this NEF/VSA representation, the matching for most ACT-R productions can be implemented via a single linear transformation. The slots and values to be matched define a VSA that will be similar to the current representation from the cortex. NEF determines the optimal synaptic connection weights that calculate the degree of similarity for all productions at once. For productions that require negative testing, we can create specialized pseudo-productions that inhibit the basic production when they match.

A more complex system is needed for productions that require the same value in two different slots. This operation could be implemented via a separate specialized group of neurons, but this requires a significant increase in the number of neurons involved. NEF also allows for dendritic nonlinearities to be used instead, but it is unclear whether this approach corresponds to the observed nonlinearities in striatal neurons.

Selection and Execution

After the matching stage, multiple productions may be active. To select which of these to fire requires a winner-take-all mechanism. While this is typically done with mutual inhibition between neurons, we can also implement this as a neural integrator where a production is selected after its represented value has accumulated above a threshold. The temporal dynamics of this process and its interaction with utility learning are still works in progress.

To execute the selected production, these neurons project to the thalamus along a narrow communication channel, which is likely too limited to also include details as to the bound variables for the production. For this reason, the output from the thalamus to the cortex cannot just be encoded symbols indicating values to be placed in slots. Instead, the thalamus must indicate transformations that should be applied to various buffers based on the values in other buffers (e.g. "copy the value from slot A in buffer X to slot B in buffer Y").

To implement this, we note that VSAs allow symbol tree manipulations of this form to be encoded as a vector as well. The output from the thalamus is thus a set of transformations sent to all areas of the cortex, acting as a controlled gating mechanism. Learning and executing these transformations has been previously demonstrated in a model of the Wason selection task (Eliasmith, 2005).

Predictions

While this model is still in the preliminary stages, two predictions are evident. First, we note that using noisy spiking neurons to encode VSAs places limits on the complexity of production that can be consistently matched. Our initial analysis indicates that accuracy will drop below 95% when more than 6 slots are used.

Second, a separate module may be introduced to address the issue of matching only when two slots have the same value. This leads to slightly different timing predictions for situations requiring a production of this form, due to requiring an extra production to indicate the slots of interest.

References

Eliasmith, C. (2005). *Cognition with neurons: A large-scale, biologically realistic model of the Wason task.* 27th Annual Meeting of the Cognitive Science Society.

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation and dynamics in neurobiological systems.* Cambridge, MA: MIT Press.

Gayler, R.W. (2003). *Vector Symbolic Architectures answer Jackendoff's challenges for cognitive neuroscience.* International Conference on Cognitive Science, Sydney.

Taking the Procedural Module Seriously: A Neural Model of Selection, Execution, and Learning in the Basal Ganglia

Andrea Stocco (stocco@cmu.edu)
Christian Lebiere (cl@cmu.edu)
John R. Anderson (ja@cmu.edu)
Department of Psychology, Carnegie Mellon University
Pittsburgh, PA 15213 USA

Introduction

Although crucial, the association between the procedural module and the basal ganglia is probably the least elaborated of ACT-R's module/brain mappings. BOLD predictions derived from the number of productions firing often mismatch the actual measurements in fMRI experiments, and crucial learning and execution mechanisms lack a biological basis.

In order to reduce this gap between computations and their biological substrate, we built a neural model of the basal ganglia and explored how the circuit's anatomy could support the functions of the procedural module. The model is based on the assumption that production rules specify how buffer contents can be moved across modules.

The Model

An overview of the model is presented in Figure 1. In the model, the striatum is the entry point to the circuit, receiving widespread projections from the cortical areas. The thalamus is the output of the system, projecting back to the cortex. Both the striatum and the thalamus are organized into compartments that mirror the organization of cortical areas.

The striatum selects one incoming representation from the cortex, and retrieves an associated destination map for each of them. These two pieces are conveyed separately on the two branches of the direct pathway, one of the two bundles of projections originating in striatum. The wiring between basal ganglia and the thalamus is such that the selected content is eventually routed to the thalamic compartment indicated by the destination map, and therefrom to the cortex.

The network is capable of binding and moving variables by allowing cortical representations to pass through, and transfer them across different buffers. It also accounts for some of ACT-R's assumptions (e.g., the serial bottleneck) and some aspects of procedural learning, (e.g, the transformation of variables into constants with practice).

The Indirect Pathway and Learning

The second bundle of striatal projections is called indirect pathway. It is longer than the direct one and runs lateral to it. In the model, it is hypothesized to hold a delayed memory of the previous destination map. This memory is used to detect circumstances where two consecutive productions can be compiled. This triggers the release of dopamine in the striatum, which fosters learning and leads to the compilations of two consecutive steps into a single operations.

The model was tested on a simple task: The aural-vocal part of Schumacher et al (2001) dual-task paradigm. It is shown that it succeeds in modeling the initial, unskilled execution of the task as well as skilled performance after production compilation.

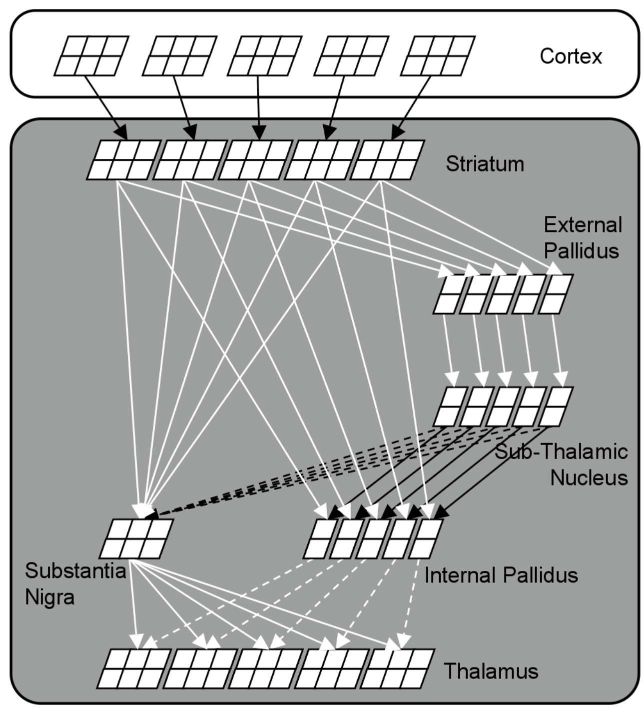


Figure 1: An overview of the architecture of the basal ganglia model. White arrows represent inhibitory connections; black arrows represent excitatory projections; solid lines represent one-to-many connections; and dotted lines represent one-to-one connections.

DECLARATIVE **MEMORY**

How do we ignore irrelevant information presented on displays?

Ion Juvina (ijuvina@andrew.cmu.edu)
Niels Taatgen (taatgen@cmu.edu)
Department of Psychology, Carnegie Mellon University 5000
Forbes Avenue, Baker Hall 336A, Pittsburgh PA, 15213 USA

Abstract

We will present empirical and computational work aiming to contribute toward a coherent theory of how humans ignore irrelevant information presented on displays.

An experiment was conducted with an isomorph of the classical Stroop task. A treatment condition included to-be-ignored extra-stimuli beside the typical Stroop stimuli. The treatment block was preceded and followed by control blocks composed of typical Stroop trials. Half of the participants received three color-patches and the other half received three words as extra stimuli. Some of the extra stimuli were randomly set to coincide with the target or distracter dimension of the main stimulus.

The results show that adding to-be-ignored stimuli to the Stroop task can be disruptive or facilitative depending on the nature of these stimuli. When extra-stimuli are of the same kind as (but not identical to) the distracter dimension of the main stimulus (words) they are facilitative. When one of the extra stimuli matches (either visually or semantically) the distracter dimension of the main stimulus there is a disruptive effect on performance. These results alone would be best explained by a "lateral inhibition" account. A visual stimulus activates its own mental representation and inhibits representations of similar stimuli.

However, a lateral inhibition account would also predict effects of the extra-stimuli on the target dimension of the main stimulus (color). Adding extra colors should disrupt performance on the primary Stroop task (color naming), except when one of the extra colors coincides with the target color, in which case the effect should be facilitative. None of these latter effects have been observed empirically.

These results pose interesting challenges to modeling cognitive control in display-based tasks. Some modeling explorations will be presented and discussed at the Workshop.

An Integrated Model of Sequential Sampling

Leendert van Maanen
(leendert@ai.rug.nl)
Dept. of Artificial Intelligence University
of Groningen P.O.Box 407 9700 AK
Groningen, the Netherlands

Human behavior involves continuously making simple decisions, such as remembering the name of someone you meet at a conference ("was that John or Jay I was just talking to?"), or reading the title of this abstract and deciding on the meaning of the words. Many of these simple decisions comprise a retrieval of a relevant chunk from declarative memory. The study of simple decision making has led to a class of models that is often referred to as *sequential sampling models* (Ratcliff & Smith, 2004). Although the details vary from model to model, they have in common that the decision process is considered as a gradual accumulation of evidence for each of the units of information (e.g., chunks) under consideration. The decision depends on whether the activation of a chunk crosses a certain threshold criterion (Ratcliff & Smith, 2004; Usher & McClelland, 2001).

Retrieval by Accumulation Evidence (RACE, Van Maanen & Van Rijn, 2007), can also be classified as a sequential sampling model. However, RACE differs from most sequential sampling models in that it is integrated in a cognitive architecture (ACT-R, Anderson, 2007). This enables the possibility to develop cognitive models of complete tasks that involve simple decision making, whereas most sequential sampling models only model the decision-making aspects of the task. In this talk, I will argue that such a isolated approach may not be enough for understanding the processes underlying task performance in these simple decision tasks. I will present data from a picture-word interference study that can only be accounted for when taking the full cognitive process into account (Van Maanen & Van Rijn, 2008). The model demonstrates that taking advantage of the strengths of sequential sampling as well as the strengths of a cognitive architecture can clarify the underlying cognitive process for this particular task, while neither of the approaches in isolation can account for the data.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford UP. Ratcliff, R., & Smith, P. L. (2004). A comparison of sequential sampling models for two-choice reaction time. *Psychological Review, 111*(2), 333-367. Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: The leaky, competing accumulator model. *Psychological Review, 108*(3), 550-592. Van Maanen, L., & Van Rijn, H. (2007). An accumulator model of semantic interference. *Cognitive Systems Research, 8*(3), 174-181.

Van Maanen, L., & Van Rijn, H. (2008). The picture-word interference effect is a Stroop effect after all. In V. Sloutsky, B. Love & K. McRae (Eds.), *Proceedings of the 30th Annual Meeting of the Cognitive Science Society*. Washington DC.

A Modular Approach to Modelling the Updating of Items in Working Memory

Martin Greaves (m.greaves@herts.ac.uk)
Psychology Department
University of Hertfordshire
Hatfield, Herts AL10 9AB, UK

Richard M. Young (r.m.young@ucl.ac.uk)
UCL Interaction Centre
University College London
Gower Street, London WC1E 6BT, UK

This talk argues for and illustrates the use of 'modular' models as a tool to help understand people's cognition in cases where different people adopt different strategies for a task, or whereone person adopts different strategies at different times or under different conditions.

The standard rhetoric says that a cognitive model is supposed to be a computer program which illuminates the cognitive processes involved in a task by performing it in the same way as peopledo. But if people do the task in different ways, then there is no single "same way" for the model to mirror. Ideally in such cases, a completely inclusive, self-contained model would adapt itsstrategy to the circumstances in a way that predicts what people do (though that still leaves problems if people differ in ways that cannot be captured in simple parametric variation of asingle model).

But such ideal, complete models are hard to attain. In their absence, we propose that useful insight into the cognitive phenomenon can be provided by a modular toolkit of models, fromwhich specific models are assembled, corresponding to particular strategies, rather as a Lego model is assembled from standard components. The purpose of such models is not to "predict what people will do"— because there is no single thing that people do — but instead to exploreand understand the space of possible strategies, and to make more circumscribed predictionsrelative to those strategies.

We propose such modular models for the task of *running memory span* (RMS). In the RMS task, a subject is presented with a long series, of unpredictable length, of items such asmonosyllabic words. When the presentation ends, the S has to try to report the last N items from the list. The RM span is defined as the length of the consecutive block of such items, including the final item, all of which are correctly reported.

The task is surprisingly difficult. Compared to the standard result of *immediate memory span* of 7 ± 2 items, typical results for RMS report a mean of 3-5 items for digits. For bi-syllabic words, RM span is typically 2 -3 items. There is controversy in the literature over how Ss perform thetask, specifically whether Ss can employ a special process of directly *updating* information in working memory as items arrive (e.g. Morris & Jones, 1990), or whether instead the familiarprocesses of attention and rehearsal are sufficient to explain behaviour (e.g. Ruiz, Elosúa &Lechuga, 2005).

In a series of experiments (Greaves, 2008), we examined the effect on RMS of varying the details of the task: rate of presentation, grouping, immediate *vs* delayed recall, the number ofitems to be reported (N), whether Ss were instructed to use specific strategies, and so on. Wereported use of a variety of strategies for performing the task. These include *N-item rehearsal*, in which S tries to rehearse a sliding list of up to the N most recent items as they arrive; *single-item rehearsal*, in which S rehearses, perhaps repeatedly, just the most recent item to arrive; and so-called *passive rehearsal* (which we here call *no-rehearsal*) in which S does not actively rehearseat all, but just focuses on the items one by one as they come in.

The findings reported here rely also on analysis of the temporal patterns of recall: the order in which items were reported, both from within the RMS and other items both recent (but outsidethe RMS as defined) and from farther back in the list.

Our approach to modelling focuses on the cognitive resources and how they are deployed in thetask. The primary relevant resources are

- A serial phonological buffer, identified with Act-R/PM's audicon, with a capacity of around 2-3 seconds of speech-like items. With additional control processes, this buffer can be used as the basis of a "phonological loop" able to rehearse internally generated items as well as external input.
- A declarative memory, subject to decay, and sensitive to recency and frequency,identified with Act-R's DM.
- A single-item aural buffer, corresponding to echoic memory.

The first two of these resources provide essentially two separate memory systems for performingthe RMS task (or many other memory tasks), but with very different properties. The audicon provides good order information and good item recall, but its capacity is sharply delimited intime, both in duration and in delay. Declarative memory, on the other hand, is effectivelyunlimited in raw capacity and is longer lasting than the audicon, but provides poor orderinformation (at least for independent items, as in these tasks) and is unreliable for iteminformation. The main interest in strategies concerns how these resources are utilised to greateror lesser effect.

The space of modelled strategies has two main dimensions: what is done during input(encoding); and what is done during output (recall). For encoding and rehearsal, the main options are

- No-rehearsal: the item is simply retrieved from the audicon, acting as an input buffer.
- Single-item rehearsal: following retrieval from the audicon, the item is subvocally repeated once, which results in its being re-encoded into the audicon.

For recall of items, the main options are

- Recall from audicon, no-rehearsal: simply retrieves the earliest item from the external sources, and reports it.
- Recall from audicon, single-item rehearsal: retrieves the earliest item from an internal source, and reports it.
- Recall from declarative memory: an item is recalled from DM (thereby strengthening it).Repeated recall of the same item is blocked by holding recalled items in the goal chunk (a

technique described by the present authors at a previous Act-R workshop), but this limits the number of items that can be recalled from DM.

Data from the experiments, including especially patterns of recall order, show clear evidence of the use of more than one memory structure. (See Greaves, 2008, for details.) Signature patterns include:

- Short runs of recent items in *forwards order*, indicative of recall from the phonological buffer.
- Separated items in *backwards order*, not necessarily consecutive, indicative of recall from declarative memory.
- When the first two patterns both occur, the runs in forwards order tend to come beforethose in backwards order, reflecting the short-lived nature of the phonological store.
- Initial recall of the final item, usually followed by one of the above, indicative of recallfrom echoic memory.

Test runs of the models largely support these interpretations and provide further understanding ofpossible strategies. For example, the model for no-rehearsal coupled with recall from DM, on 60% of its runs provided a reporting pattern of (P1 P2) or (P1 P2 P3), i.e. backwards recall of thelast two or three items. (Pi means the i-th item counting backwards from the last.) There is also a scattering (around 12% each) of (P1 P3), (P1 P3 P2), and (P1 P2) followed by earlier items.Broadly similar patterns appear in the data from Ss instructed to use the no-rehearsal strategy. Similarly, with single-item rehearsal and retrieval from DM, the two most common patterns(around 40% each) were (P1 P2 P3 …) and (P1 P2 P4 …), i.e. starting with backwards recall ofthe last three items, or the last two followed by the fourth. These again are found among the dominant patterns of experimental Ss instructed to use the single-item strategy (although ofcourse the human data are mixed in with other patterns such as forward recall of the last 3 or 4items, indicative of recall from the phonological store). With recall from the audicon, no-rehearsal leads to recall of the last 2-3 items in forwards order for the default decay-time of 3 seconds in the audicon. With single-item rehearsal, consistent recall of the final 3 items (P3 P2 P1) is observed.

Further to these findings, an interesting and unexpected finding from the modelling is the existence of micro-strategies. As Gray & Boehm-Davis (2000) point out, small shifts in thetiming of cognitive process steps can make a significant impact on the overall performance. For example, with single-item rehearsal, instead of clearing the aural buffer at the end of theencoding step, the clearing can be left to the subsequent cycle of rehearsal. This leaves the contents of the aural buffer available for immediate recall following rehearsal, corresponding tothe residual availability of an item in echoic memory. Another possibility is to defer rehearsal ofan item until the arrival of the following item. This delays recoding of the first item until as lateas possible, minimising the time during which the rehearsed item is subject to decay. This again allows the immediate recall of the final item. These micro-strategic variations can, under theright conditions, contribute a whole extra item to the RMS, raising it for example from 2 to 3, orfrom 3to 4.

Overall, the study showed that people performing the RMS task adopt a variety of strategiesdrawing on different cognitive resources, and that their performance can be accounted for bystandard processes of attention and rehearsal with no need to invoke a special updating process.

The role of modelling has been, firstly, to test and refine our intuitive analysis of the behaviouralconsequences of the different strategies; and secondly, to help us interpret and dissect theempirical data, and to understand the necessarily compound and messy patterns of recall seen in the data as being composed of distinct contributions from the different resources and strategies.The modular structure of the models allows one component to be changed leaving othercomponents unchanged. This approach to modelling allows alternative accounts to be tested within a single framework, showing whether specific components can account for a particularaspect of behaviour, or whether they should be ruled out.

Looking to the future, we see further ways in which these "toolkit" models can be used:

(1) Like some other modular models, this one offers hints as to how the strategies might arise.If each module makes local sense in terms of the task requirements, then it is plausible thatSs would choose to do it, at least up to the point where there is evidence that it doesn'twork. For example, attending to each incoming item has the automatic side-effect of storing the item in DM. So DM is always available as a resource for recall. Likewise, if an item is available for recall from a very fragile store, such as the aural buffer, it makes sense to report it before moving on to less ephemeral resources.

(2) These models may not be able to predict "what people do" (even allowing that there is noone thing that they *do* do), but Act-R, as a model of the human cognitive architecture,should excel at telling us which potential strategies are humanly executable and which arenot. For example, we devised a counter-intuitive strategy for RMS in which each incoming item is added to the *beginning* (instead of the end) of the list being rehearsed, and later partsof the list are simply abandoned when it gets too long. In principle, because this strategy allows S to focus on the first few elements of the list, which are the easiest ones to rehearse,it should have played to the strengths of both human cognition and of the task requirementsby constantly maintaining a list of the most recent, say, six items. In practice however, Ss found the strategy too complex, and were losing incoming items while still struggling to work out what they were supposed to do with a previous item.

(3) Another question where Act-R should be in a position to predict the answer, is the matter ofwhat happens if a strategy is practised and used repeatedly over a long time. To continue with the same example, if the "backwards encoding" strategy just described were practised repeatedly, perhaps with a gradually increasing presentation rate, it is still possible that itwould turn out to be effective. Act-R ought to be able to tell us.

Finally, we mention briefly two other related attempts at modelling working memory tasks.David Huss, working with Mike Byrne, several years ago built an implementation of thephonological loop using the Act-R/PM audicon. In the end our modelling took a differentdirection, but we studied his model closely and learned from it. And recently, after the work reported here was finished, we became aware of a PhD by Krawitz (2007) which presents an EPIC model of a number of updating-style working memory tasks. His findings largely agreewith or complement ours, but he has far less interest in multiple strategies and, so far as we can see, none at all in patterns of order of recall.

References

Gray, W. D. & Boehm-Davis, D. A. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied, 6* (4), 322-335.

Greaves, M. H. (2008). Rehearsal-based strategies for updating verbal working memory. PhD thesis. School of Psychology, University of Hertfordshire, UK.

Krawitz, A. (2007). Computational modeling of updating operations in the performance of basic verbal working-memory tasks. PhD thesis. University of Michigan.

Morris, N. & Jones, D. M. (1990). Memory updating in working memory: The role of the central executive. *British Journal of Psychology, 81* (2), 111-121.

Ruiz, M., Elosúa, M. R. & Lechuga, M. T. (2005). Old-fashioned responses in an updating memory task. *Quarterly Journal of Experimental Psychology, 58A* (5), 887-908.

Explorations of the ACT-R Architecture for Learning and Forgetting Performance

Jong W. Kim (jongkim@psu.edu)
Frank E. Ritter (frank.ritter@psu.edu)
Richard J. Koubek (rkoubek@psu.edu)
The Pennsylvania State University
University Park, PA 16802, USA

We explored the learning and forgetting performance of 30 participants in an office work task, using a relatively unknown spreadsheet. Participants were randomly assigned to three different retention intervals (6-, 12-, and 18-days) to investigate forgetting. The Power Law of Learning again matched human learning behavior. Retention intervals (6-day, 12-day, or 18-day) showed clear effects on the amount of forgetting. The ACT-R theory, which is used as the main theoretical background, was tested against human data with regard to learning and forgetting. The skill retention model in ACT-R was developed to predict a mouse user's learning and forgetting performance in one subtask. The model predicted the learning performance on this a half minute task with $r^2 = 0.8$ and *RMSSD* = 1.4, compared with human data. The model showed that an ACT-R model is able to predict learning. The endeavor of human performance modeling using ACT-R can be used to evaluate efficacy of a training regimen by predicting learning. We note the advantages and disadvantages of using a current cognitive architecture in predicting training and provide suggested directions toward exploration of smart training.

ARCHITECTURE

Confronting Architectural Drift in ACT-R

Glenn Gunzelmann (glenn.gunzelmann@mesa.afmc.af.mil)
Air Force Research Laboratory; 6030 South Kent Street; Mesa, AZ 85212 USA

The ACT-R architecture continues to experience vibrant development, resulting in theoretical progress and a continuous stream of implementation changes that improve upon the details of the architecture. Since the initial release of ACT-R 6.0 at the 2005 ACT-R Workshop in Trieste, Italy, over 500 ACT-R code updates have been committed to the Subversion server. The modifications implement incremental changes that address a variety of issues, from correcting bugs and extending the capabilities of ACT-R so that the implementation better reflects the theory, to updating documentation and "extras" (e.g., the Environment interface, contributed files). In general, major theoretical additions and changes have been rare, although p* productions, reinforcement learning for production utilities, and the new vision module are notable exceptions.

Development of the architecture does not come without a cost, however. As the implementation is improved and theoretical details are refined, there are potentially important, though often subtle, changes in ACT-R's behavior. This issue became more salient in recent efforts to update a model of the impact of fatigue from ACT-R 5 to ACT-R 6. The model performs a simple task called the Psychomotor Vigilance Test (PVT), a sustained attention task where participants are asked to monitor a known location and respond to the onset of a stimulus by pressing a response button. The model itself is straightforward, but there are a number of subtleties in the human data, including the impact of sleep loss and circadian rhythms, which we account for in our model in ACT-R 5 (see Gross, Gunzelmann, Gluck, Van Dongen, & Dinges, 2006).

Because of the simplicity of the model, updating the syntax of the task delivery code and model to run in ACT-R 6 was straightforward, though care was required because the effects of some operations in ACT-R 6 are different in influential ways from ACT-R 5. Using the "old" (ACT-R 5) utility mechanism for production selection, the behavior of the ACT-R 6 model is roughly equivalent to the ACT-R 5 model when the fatigue mechanisms are disengaged in both. However, as the impact of the fatigue mechanisms increases, the performance of the ACT-R 6 model diverges from the performance observed in ACT-R 5 (Figure 1).

An investigation of this issue has led to some possible answers. The most likely appears to be the interplay between the "run" command and the ACT-R clock and scheduler. Variations in how the clock is managed and events are scheduled on the boundaries of run calls appear to result in small but significant changes to the model predictions. This illustrates that the behavior of ACT-R 6 can differ from ACT-R 5 depending upon the task context and the details of the implementation. In cases where the microstructure of ACT-R's behavior can impact the conclusions drawn from models – where milliseconds matter (e.g., Gray & Boehm-Davis, 2000) – these small differences may have important implications.

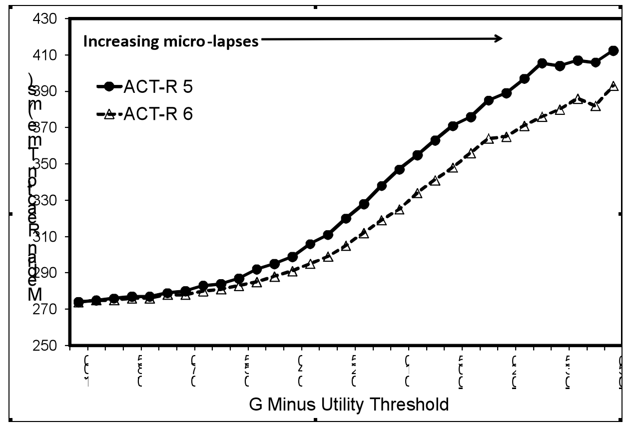


Figure 1: Median alert response time on the PVT as fatigue mechanisms are engaged (increasing micro-lapses).

Dealing with this issue could involve locking the model in a particular version of the architecture. However, the goal is to develop a cumulative theory that can be used to generate predictions of human performance in novel contexts by using tasks like to PVT to baseline fatigue parameters. To take advantage of emerging capabilities in the architecture, bug fixes, and extensions, it is important to remain current. Thus, it is imperative to carefully validate the model as it is run in new and different versions of ACT-R. Here, this requires careful analysis of ACT-R traces to ensure that action sequences and event timings remain consistent. The right approach to addressing architectural drift like this depends on the model and application, but it is something that modelers must confront somehow in their research.

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was sponsored by the Air Force Research Laboratory's Warfighter Readiness Research Division and by grant 07HE01COR from the Air Force Office of Scientific Research. Thanks to Dan Bothell, Kevin Gluck, and Rick Moore for helpful discussion.

References

Gray, W. D. & Boehm-Davis, D. A. (2000). Milliseconds Matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied, 6(4)*, 322-335..

Gross, J. B., Gunzelmann, G., Gluck, K. A., Van Dongen, H. P. A., & Dinges, D. F. (2006). Prediction of circadian performance during sleep deprivation. In R. Sun & N. Miyake (Eds.), *Proceedings of the Twenty-Eighth Annual Meeting of the Cognitive Science Society* (p. 297-302). Mahwah, NJ: Lawrence Erlbaum Associates.

Simulating Pre-Task Appraisal of Serial Subtraction

Frank E. Ritter (frank.ritter@psu.edu)[1], Michael Schoelles (schoem@rpi.edu)[2], Sue Kase (skase@ist.psu.edu)[1], and Laura Cousino Klein (lxk18@psu.edu)[3]

[1] College of IST, Penn State; [2] Cognitive Science, RPI; [3] Biobehavioral Health, Penn State

Not too long ago ACT-R was little more than a memory system. Perception and motor processes were added to embody ACT-R. One of the next development goals is the addition of emotions. Incorporating the effects of stress and other behavioral moderators into the ACT-R architecture represents our contribution to meeting this goal.

As a starting point we looked at existing theories of stress and how to incorporate them into the ACT-R architecture in the form of overlays (Ritter et al, 2007). An overlay is a change to the existing architecture to implement some feature and may be as simple as a change to a default parameter or as complex as a new module. We were able to identify some potential overlays but we often found that the existing theories were not precise enough for encoding into a cognitive architecture. The next step was to conduct an experiment on the effect of behavioral moderators and develop a model that would incorporate the overlays we had identified if possible and identify new ones.

The Trier Social Stressor Task (TSST) (Kirschbaum, Pirke & Hellhammer, 1993) has been widely used to study the physiological effects of stress. Part of the TSST is a serial subtraction task that lends itself to computational cognitive modeling. It includes a pre-task appraisal measurement that allows the experimenter to group participants into "threatened" (little control over performance, less able to cope) and "challenged" (in control, can more than cope) groups. These pre-task appraisals can moderate stress responses to the task and performance on this task (Quigley, Feldman Barrett, & Weinstein, 2002).

In the TSST, serial subtraction consists of subtracting 7 or 13 from a 4-digit number, speaking the answer, and then proceeding with the next subtraction for the answer just given. This is done for 120 s at which time the experimenter interjects a stress inducing comment. The task then continues for 120 s. The participant is given feedback for incorrect responses and cannot proceed to the next number until the correct response is spoken. In our empirical study (Ritter et al., submitted) we found a high variance in number of subtraction attempts and accuracy. We also found several strategies for speaking the answer.

Our effort at modeling these distinctions is an ACT-R model with overlays to the architecture. Our ACT-R model of this task integrates declarative, procedural, and vocal processes. Several different vocalization strategies are implemented. These strategies contribute substantially to the model's ability to simulate the threatened/challenged distinction. This suggests that stress can be simulated in ACT-R not only through overlays but through strategy selection, which is a different type of control than changing the architecture. The overlays that we use to model this difference are parameter setting overlays. That is, different settings of parameters are used for simulating a threatened and a challenged participant. We are currently comparing the model to human performance on the number of subtraction attempts, accuracy, and error types.

Developing the parameter setting overlays required searching a vast parameter space. For example, Wickens proposes a reduced "working memory" under high stress. In ACT-R there are several parameters involved in "working memory" such as the retrieval threshold, the base level constant, the activation noise, and the latency factor. To date we have looked at activation noise and the base level constant and are in the process of incorporating retrieval threshold and the latency factor into an overlay. To find settings for these requires new ways of using high performance computers to run models. We have developed a methodology using genetic algorithms for running the serial subtraction model on high performance computing systems (Kase, Ritter, & Schoelles, 2008). An interesting finding of this effort has been the types of errors made by the different groups—the vocalization strategy seems to play a large part in this. We hypothesize that activation may be spreading from the vocal buffer inducing memory errors. We are currently investigating how to accomplish this in ACT-R.

References

Kase, S., Ritter, F. E., & Scholles, M. (in press). From modeler-free individual data fitting to 3-D parametric prediction landscapes: A research expedition. In *Proceedings of the 2008 Conference of the Cognitive Science Society*. Kirschbaum, C., Pirke, K.-M., & Hellhammer, D. H. (1993). The Trier Social Stress Test—A tool for investigating psychobiological stress responses in a laboratory setting. Neuropsychobiology, 28, 76-81. Quigley, K. S., Feldman Barrett, L. F., & Weinstein, S. (2002). Cardiovascular patterns associated with threat and challenge appraisals: A within-subjects analysis. Psychophysiology, 39, 292-302. Ritter, F. E., Bennett, J. M., Kase, S. E., Rodrigues, I., Klein, L. C. (submitted). Cognitive performance on a serial subtraction task. Ritter, F. E., Reifers, A. L., Schoelles, M., & Klein, L. C. (2007). Lessons from defining theories of stress for architectures. In W. Gray (Ed.), Integrated models of cognitive systems (pp. 254-262). New York: Oxford.

Disclaimer and Acknowledgments: This project was supported by ONR (N000140310248, FER & LCK), the National Science Foundation (SBR 9905157, LCK), and from the Penn State University College of Health and Human Development (223 15 3605; LCK). The services provided by the General Clinical Research Center of The Pennsylvania State University to help run the study are appreciated (NIH Grant M01 RR 10732).

Architectural Support for Adversarial Behavior
ACT-R Workshop Presentation

Christian Lebiere (cl@cmu.edu)
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Bradley J. Best (bjbest@adcogsys.com)
Adaptive Cognitive Systems
Boulder, CO 80302, USA

Adversarial behavior is a pervasive aspect of human activity. It happens every day in naturalistic settings as diverse as driving, work, and interpersonal relationships. In more abstract settings, it is featured prominently in paradigms ranging in complexity from simple games (e.g. board games) to first-person shooter video games (e.g. Doom, Quake, etc) to command-and-control simulations in military settings (e.g. Command and Conquer) to massive multi-player online environments (e.g. EverQuest). Adversarial behavior has been a central application area of Artificial Intelligence from its early days (e.g. Samuels' checker player) and a key form of performance benchmark (e.g. the Friedkin Prize for chess, Robocup). In economics, a formal theory (game theory) has been developed to account for the unique aspects of behavior that it brings forth.

In stark contrast, adversarial behavior has received comparatively little attention in cognitive psychology, for a number of reasons both practical and theoretical. However, we have shown that, despite that lack of interest, cognitive models of adversarial behavior can provide both high-levels of functionality (e.g. Roshambo competition, ICCM Poker symposium) as well as close correspondence to human performance (e.g. PRS player, baseball batting, backgammon player). Moreover, those models have exposed the flaws in game-theoretic accounts of adversarial behavior (e.g. PRS, Prisoner's Dilemma) as well as the functional advantages of cognitive architectures over machine learning approaches (e.g. baseball batting).

However, as one moves from those simple, formal domains to more naturalistic ones (e.g. MOUT), complexities arise. Central characteristics of adversarial behavior, including constant unpredictiveness and adaptivity, stress some underlying assumptions of cognitive architectures. Key aspects of the practice if not the theory of cognitive modeling such as the ability to anticipate and reflect in the design of the cognitive model the structure of the problem-solving process and the representation of information received from the environment that are mainstays of experimental psychology designs are under stress in adversarial environments. The change in the nature of task from a static experimental design to constantly changing, dynamic adversaries raises the bar for flexibility and adaptivity in cognitive models.

We will illustrate those challenges using a task involving teams of adversarial agents in a synthetic environment. Key aspects of needed functionality for cognitive agents in that environment are the needs to flexibly and robustly represent the current context, access declarative information such as plans and strategies, manage conflicting goals, and reconcile goal-directedness and reactivity in mapping context to strategies. We will describe approaches at providing that functionality within the context of the ACT-R cognitive architecture and attempts at validating it using traditional modeling methods. We propose a 3-level validation methodology for cognitive architecture mechanisms that include Functionality (which capability is added or improved?), Fitting (which results are accounted for in a better or more natural way?) and Compatibility (impact on existing models as per Simon's methodology of backward-fitting: simpler/more robust accounts (great), unaffected (good), or requiring change (good or bad)).

The Influence of Task Demands in a Model of Time Estimation

Nele Pape (nele.pape@zmms.tu-berlin.de))
GRK Prometei, ZMMS, Technische Universität Berlin
Franklinstr. 28-29, FR 2-7/2, 10587 Berlin Germany
Leon Urbas (leon.urbas@tu-dresden.de)
Dept. of. Electrical and Computer Engineering, Technische Universität Dresden
Helmholtzstrasse 10, 01069 Dresden Germany

Abstract

A model of prospective time-estimation is introduced which explains the interplay of working memory demands on duration estimation. The approach is integrated into ACT-R and tested by estimating the duration of a task that varied coordinative and sequential demands on working memory. The comparison with experimental data shows that the model is able to simulate the influence of these demands on human time-estimation.

Introduction

The cognitive ability to be aware of the passage of time is beneficial in dynamic environments. Time-judgments are important to stay tuned to this environment, to plan steps in a task, and to identify problems (e.g. after an expected duration of booting a computer the monitor stays blank).

In the context of human-machine interaction, the knowledge of temporal dependencies is of great interest. For example, in order to drive safely, drivers need to divide their visual attention in a reasonable way between traffic and secondary tasks such as In-Vehicle-Information-Systems. Operators can deduce a malfunction from the system's temporal behavior in comparison to the temporal properties of a functioning system (Schulze-Kissing, 2007).

Here we introduce a computational model of time-estimation that shows how a demanding task disrupts the ability to judge time. In this model, the need to maintain and update information (e.g. a number in arithmetic) during a task distorts the construction of time representation during this period. The approach is integrated into ACT-R (atomic components of thought – rational analysis; Anderson et al., 2004). In this way the influence of cognitive processes and demands on the construction of time representations can be explored in a cognitive context. For cognitive architectures, it is valuable to have an integrated component that simulates temporal human behavior. This is especially important for modeling switching tasks, multitasking and tasks under time-pressure.

The integrated timing-model is tested within a counting task (Dutke, 1997) with varying demands to compare human data to the performance of the model.

Psychological Models of Time-Estimation

The research field of human time-estimation explains differences in estimates on a number of factors such as the duration of the interval, the kind of instruction given to the subjects, when and how an interval is estimated (production, reproduction), or the number of incidents experienced during a given interval.

It is generally found that a demanding task affects time-estimation. Time-estimates are shorter when compared to less demanding conditions (Zakay, 1993; Dutke, 1997; Brown, 1997). A number of authors (e.g. Block & Zakay, 1996; Brown & West, 1990) assume that attention-allocation is the responsible factor for the interference between task and time-estimates. A number of other authors assume a strong influence of working memory on time-estimation.

Quantitative Time-Estimation Model

The proposed model of prospective time estimation consists of four parts: a pacemaker that generates pulses, an accumulator which collects pulses for short durations, a process of construction which updates the time representation, and a procedure which finally estimates time, e.g. by comparing an old time representation with a new interval as in the reproduction task. The first two parts are modeled by adding a timing-module to the architecture. The third and forth parts of the approach integrate the output of the new timing-module with already existing processes and modules of the cognitive architecture.

Discussion

The approach under discussion explains the way working memory demands effect duration estimation. For a time-estimation, the temporal representation during an interval has to be updated continuously. In order to do this the latest representation has to be maintained in working memory. A task that calls upon working memory mechanisms interferes with the working memory mechanism of maintaining the latest time-representation. Compared to other theoretical accounts of duration estimation, this model is more parsimonious in that no additional elements like an attentional gate or processes of a central executive are necessary to explain observable distortions of the estimation process.

Acknowledgments

This work was sponsored by VolkswagenStiftung (Research Group *Modelling of User Behaviour in Dynamic Systems*) and Deutsche Forschungsgemeinschaft (DFG Research Training Group *Prospective Design of Human Technology Interaction*, GRK 1013).

References

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

Baddeley, A. (1986). *Working memory*. Oxford, UK: Clarendon Press.

Block, R.A., & Zakay, D. (1996). Models of psychological time revisited. In H. Helfrich (Eds.), *Time and mind. Proceedings of the International Symposium on Time and Mind* (pp. 171-195). Kirkland, WA: Hogrefe & Huber.

Brown, S.W. (1997). Attentional resources in timing: Interference effects in concurrent temporal and nontemporal working memory tasks. *Perception & Psychophysics*, 59, 1118-1140.

Brown, S.W., & West, A.N. (1990). Multiple timing and the allocation of attention. *Acta Psychologica*, 75, 103-121.

Dutke, S. (1997). *Erinnern der Dauer: Zur zeitlichen Rekonstruktion von Handlungen und Ereignissen* [Remembering duration: The temporal reconstruction of actions and events]. Lengerich, Germany: Pabst Science Publishers.

Dutke, S. (2005). Remembered duration: Working memory and the reproduction of intervals*. Perception & Psychophysics*, 67, 1404-1422.

Schulze-Kissing, D. (2007). *Eine Mikroanalyse des zeitlichen Fehlers in der Mensch-Maschine-Interaktion*. VDM Verlag.

Taatgen, N.A., van Rijn, H., & Anderson, J.R. (2007). An Integrated Theory of Prospective Time Interval Estimation: The Role of Cognition, Attention and Learning. *Psychological Review*, *114*(3), 577-598.

Zakay, D., (1993). Relative and absolute duration judgments under prospective and retrospective paradigms. *Perception & Psychophysics* 54, pp. 656–664.

The Influence of Task Demands in a Model of Time Estimation

Nele Pape, Technische Universität Berlin

prometei graduiertenkolleg

ZMMS TU DFG

Time-judgments are important to:

- stay tuned to the environment
- to plan steps in a task
- to detect delays, i.e. absence of events

- In the context of human-machine interaction, the knowledge of temporal dependencies is of great interest, e.g.:
  - in order to drive safely
  - to deduce a malfunction from the system's temporal behavior

Why is time estimation important for cognitive architectures?

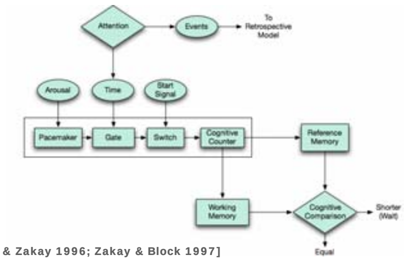
- It is valuable to have an integrated component that simulates temporal human behavior:
  - switching tasks
  - multitasking
  - tasks under time-pressure (subjective time on task / remaining time)
  - timing of action without external cues

Psychological Model: Attentional-Gate

Problems:

→What is attention-to-time?

→What corresponds to this in ACT-R?



[Block & Zakay 1996; Zakay & Block 1997]

Models of prospective time-estimation…

… are already implemented in ACT-R
(Byrne, 2006; Taatgen, van Rijn & Anderson, 2007)

- Byrne: Attentional Gate
  - Attention can either be devoted to time (which opens the "gate," allowing pulses to be counted), or to other tasks
  - In ACT-R terms, this means:
    - a production has to fire to increment the counter
    - pulses can be missed, producing underestimates

- Taatgen: Slow down of pacemaker (duration < 15 sec)
  - If a task is very demanding people forget to estimate time and have to restart their estimation
  - → no account what kind of task demands are responsible for distortions in time-estimates

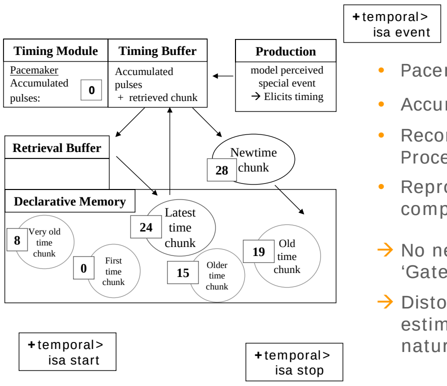
---

The '**Ta**sk **S**ensitive **T**ime **E**stimation Module' (TaSTE Module)

- Target:
  - Prospective time estimation for longer durations
- Foundations:
  - Empirical findings of influence of task demands on time estimation (Brown, 1997; Dutke, 2005)
  - Time Traces in Episodic Memory (Baddely, 2002)
- No Magic Doctrine:
  - Use regular memory processes such as Spreading Activation and the concept of Working Memory already implemented in ACT-R to „Emerge" human time estimation effects

(Lovett, Reder & Lebiere, 1999)

The 'Task Sensitive Time Estimation Module' (TaSTE Module)

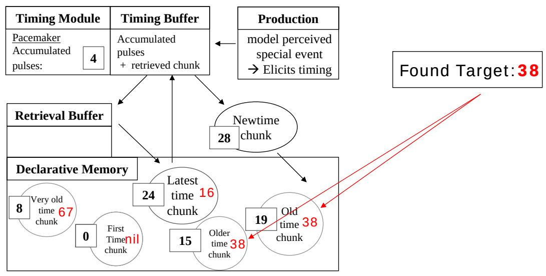


- Pacemaker
- Accumulator
- Reconstruction Process
- Reproduction or comparison

→ No need for a 'Gate'!

→ Distortions of time estimates emerge naturally

Context information and Spreading Activation



- Context dependent
- Explains influence of WM demand (Lovett, Reder & Lebiere, 1999)
- Episodic Memory Traces (Baddley, 2002)

An example task

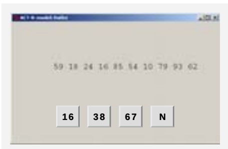
Counting task (Dutke, 2005) :

Subjects had to scan the list for:

(1) one target (16) or…

(2) one of three targets (16, 38, 67)

… and say "yes" at the third occurrence



- High coordinative demands (coor+)

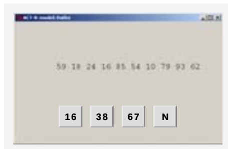
  The WM system has to coordinate storing, refreshing, updating, and responding – without confusing the tallies
- High sequential demands (seq+)

  The system has to deal with a higher rate of updating the results of the counting process (35% vs. 68%) (14 vs. 27 targets)

	coor-	coor+
seq-	cond. 1	cond. 3
seq+	cond. 2	cond. 4

An example task

Subject had to reproduce the duration they experienced (400sec)



Hypotheses:

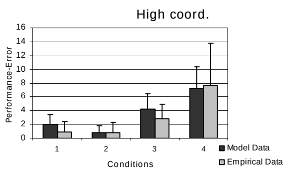
Both demands will distort time estimation if the Attentional Gate Model is true

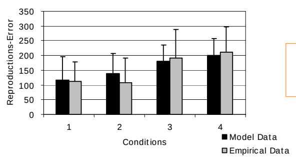
Just coordinative demands will distort time estimation if they depend on specific wm resources

	coor-	coor+
seq-	cond. 1	cond. 3
seq+	cond. 2	cond. 4

Results counting Task

	coor-	coor+
seq-	cond. 1	cond. 3
seq+	cond. 2	cond. 4





Sign. difference for coordinative demands p<.01

Outlook

- Next steps:
  - Evaluate predictive power of the model
  - Variation of task
  - Influence on well learned durations
  - Different durations
  - Different ways of using time representations (reproduction with/without task)

INVITED **SPEAKER**

Cognitive Architecture: Past, Present, Future

John Laird
University of Michigan

ACT-R Workshop
July 18, 2008

What is Cognitive Architecture?

Fixed mechanisms and structures that underlie cognition
- – Processors that manipulate data
- – Memories that hold knowledge
- – Interfaces that interact with an environment

- Sharp distinction between
  - – task-dependent knowledge and
  - – task independent architecture

2

Cognitive Architecture Types

Goal

Modeling

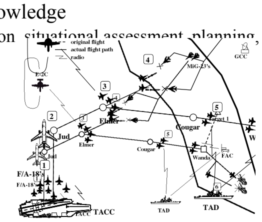
Compare to human data:
- Generality
- Reaction time
- Error rates
- Learning rates
- Changes in brain physiology
…

Functionality

Solve problems:
- Generality
- Efficiency
- Robustness
- Flexibility
- Instructability
…

Cognitive Architecture: Creating Complete Agents

- Coarse-grain integration
  - – Connecting all capabilities, from perception to action
- Fine-grain integration of capabilities/knowledge
  - – Dynamic intermixing of perception, reaction, situational assessment, planning, meta-reasoning, language, …
- Ubiquitous learning that
  - – is not deliberately cared for and controlled
  - – is incremental and real-time
  - – doesn't interfere with reasoning
  - – *impacts everything an agent does*
- Long-term existence
  - – Behave for hours or days, not minutes
  - – Scaling to tasks employing large bodies of knowledge
  - – Generation of goals, drives, internal rewards, …
- Turing equivalence isn't sufficient
  - – Architectures have different *complexity profiles*



History of Cognitive Architecture 1969-2000

1970 1975 1980 1985 1990 1995 2000

- GPS (Ernst & Newell, 1969) Means-ends analysis, recursive subgoals
- ACT (Anderson, 1976) Human semantic memory
- CAPS (Thibadeau, Just, Carpenter) Production system for modeling reading
- Soar (Laird, & Newell, 1983) Multi-method problem solving, production systems, and problem spaces
- Theo (Mitchell et al., 1985) Frames, backward chaining, and EBL
- PRS (Georgeff & Lansky, 1986) Procedural reasoning & problem solving
- BB1/AIS (Hayes-Roth & Hewitt 1988) Blackboard architecture, meta-level control
- Prodigy (Minton et al., 1989) Means-ends analysis, planning and EBL
- MAX (Kuokka, 1991) Meta-level reasoning for planning and learning
- Icarus (Langley, McKusick, & Allen,1991) Concept learning, planning, and learning
- 3T (Gat, 1991) Integrated reactivity, deliberation, and planning
- CIRCA (Musliner, Durfee, & Shin, 1993) Real-time performance integrated with planning
- AIS (Hayes-Roth 1995) Blackboard architecture, dynamic environment
- EPIC (Kieras & Meyer, 1997) Models of human perception, action, and reasoning
- APEX (Freed et al., 1998) Model humans to support human computer designs

5

History of Cognitive Architecture 1969-2000

Psychological Modeling

1970 1975 1980 1985 1990 1995 2000

- GPS (Ernst & Newell, 1969) Means-ends analysis, recursive subgoals
- *ACT (Anderson, 1976) Human semantic memory*
- *CAPS (Thibadeau, Just, Carpenter, 1982) Production system for modeling reading*
- ~~*Soar (Laird, & Newell, 1983) Multi-method problem solving, production systems, and problem spaces*~~
- Theo (Mitchell et al., 1985) Frames, backward chaining, and EBL
- PRS (Georgeff & Lansky, 1986) Procedural reasoning & problem solving
- BB1/AIS (Hayes-Roth & Hewitt 1988) Blackboard architecture, meta-level control
- Prodigy (Minton et al., 1989) Means-ends analysis, planning and EBL
- MAX (Kuokka, 1991) Meta-level reasoning for planning and learning
- Icarus (Langley, McKusick, & Allen,1991) Concept learning, planning, and learning
- 3T (Gat, 1991) Integrated reactivity, deliberation, and planning
- CIRCA (Musliner, Durfee, & Shin, 1993) Real-time performance integrated with planning
- ~~AIS (Hayes-Roth 1995) Blackboard architecture, dynamic environment~~
- *EPIC (Kieras & Meyer, 1997) Models of human perception, action, and reasoning*
- ***APEX (Freed et al., 1998) Model humans to support human computer designs***

6

History of Cognitive Architecture 1969-2000

Active Architectures

1970, 1975, 1980, 1985, 1990, 1995, 2000

- GPS (Ernst & Newell, 1969) Means-ends analysis, recursive subgoals
- *ACT (Anderson, 1976) Human semantic memory*
- *CAPS (Thibadeau, Just, Carpenter, 1982) Production system for modeling reading*
- ~~*Soar (Laird, & Newell, 1983) Multi-method problem solving, production systems, and problem spaces*~~
- Theo (Mitchell et al., 1985) Frames, backward chaining, and EBL
- PRS (Georgeff & Lansky, 1986) Procedural reasoning & problem solving
- BB1/AIS (Hayes-Roth & Hewitt 1988) Blackboard architecture, meta-level control
- Prodigy (Minton et al., 1989) Means-ends analysis, planning and EBL
- MAX (Kuokka, 1991) Meta-level reasoning for planning and learning
- Icarus (Langley, McKusick, & Allen,1991) Concept learning, planning, and learning
- 3T (Gat, 1991) Integrated reactivity, deliberation, and planning
- CIRCA (Musliner, Durfee, & Shin, 1993) Real-time performance integrated with planning
- ~~AIS (Hayes-Roth 1995) Blackboard architecture, dynamic environment~~
- *EPIC (Kieras & Meyer, 1997) Models of human perception, action, and reasoning*
- *APEX (Freed et al., 1998) Model humans to support human computer designs*

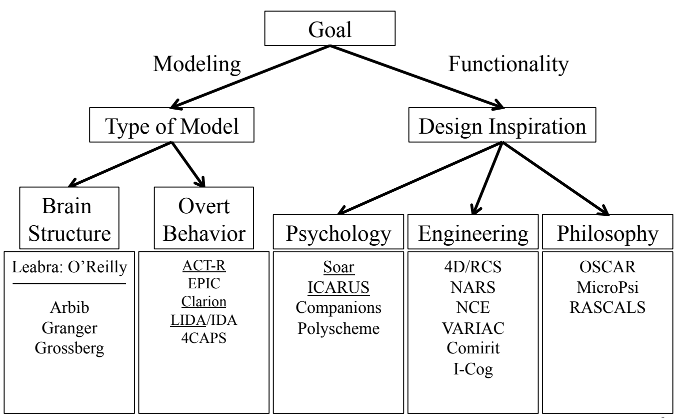
7

Current State of Cognitive Architecture

- Explosion of different architectures
  - Developed with different goals in mind
- Lots of different components
- But some significant commonalities

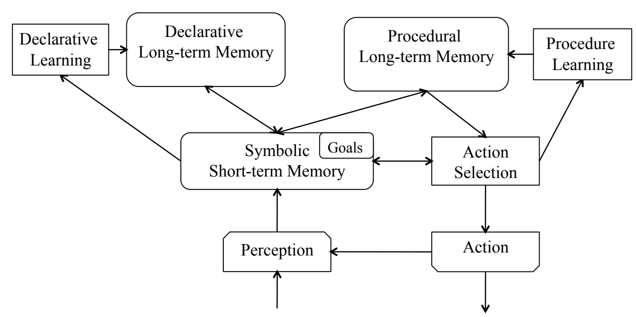
8

Classification of Active Architectures



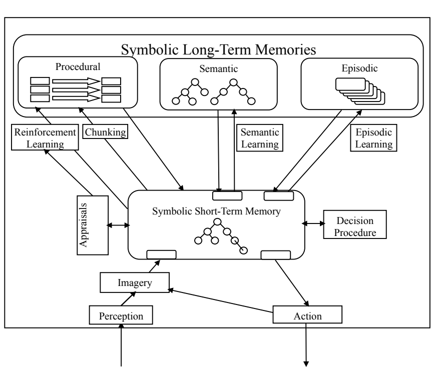
9

Common Architectural Structure

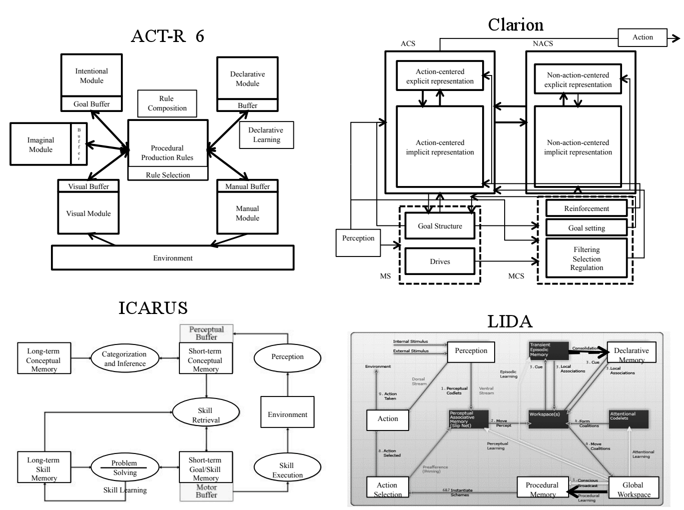


10

Soar



11



Common Processing & Representations Across Many Architectures

- Complex behavior arises from sequence of simple decisions over internal and external actions controlled by knowledge
  - No monolithic plans
  - Significant internal parallelism, limited external parallelism
  - For cognitive modeling, ~50msec is basic cycle time of cognition
- <u>Knowledge access must be bounded to maintain reactivity</u>
- Symbolic long- & short-term knowledge representation
  - Procedural & semantic (Clarion also has non-symbolic)
  - Relational representations (-Clarion)
- Non-symbolic representation for action selection
- Learning is incremental & on-line (-LIDA)

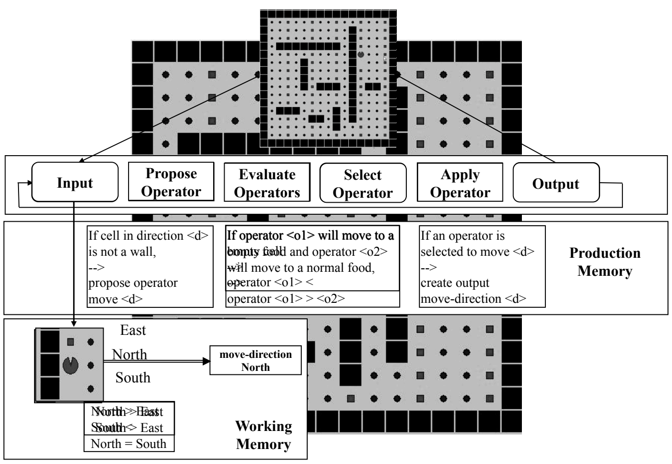
13

Unique Components

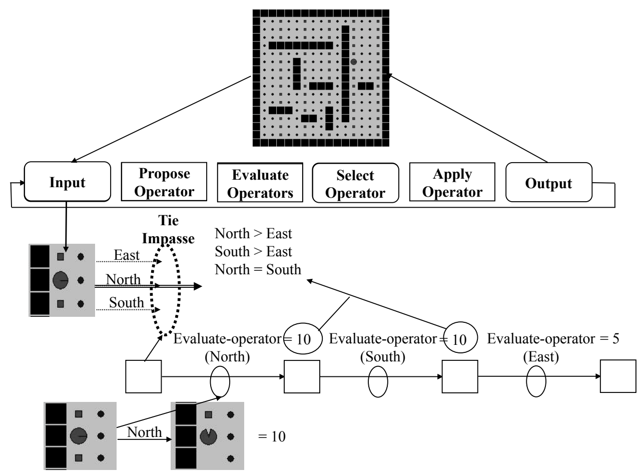
- Categorization [ICARUS & LIDA]
- Episodic Memory [Soar & LIDA]
- Attention [LIDA]
- Mental Imagery [Soar]
- Appraisals – emotion [Soar]
- Drives [Clarion]

14

Soar 101: Eaters



Soar 102: Substates



Key Differences: Act-R and Soar

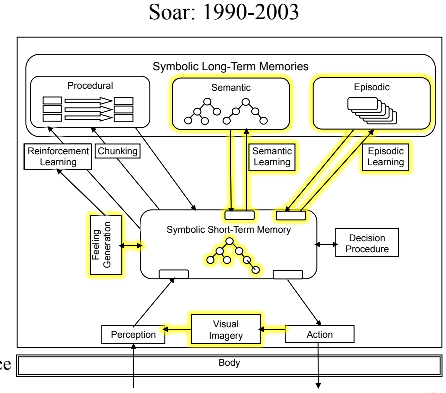
- Working memory representation & size
  - Fixed buffers vs. unbounded graph structure
- Persistence of working memory elements
  - Persistent vs. persistent and non-persistent
- Encoding of procedural knowledge
  - Rules vs. operators/parallel rules + parallel elaborations
- Goal structures
  - Deliberate vs. architectural substates
- Meta-data
  - ?? vs. architectural impasses/substates

17

Extending Soar

Soar: 1990-2003

- Learn from rewards
  - Reinforcement learning
- Learn facts
  - What you know
  - Semantic memory
- Learn events
  - What you remember
  - Episodic memory
- Basic drives and …
  - Emotions, feelings, mood
- Non-symbolic reasoning
  - Mental imagery
- Working memory relevance
  - Activation



18

Theoretical Commitments

Stayed the Same

- Problem Space Computational Model
- Long-term & short-term memories
- Associative procedural knowledge
- Fixed decision procedure
- Impasse-driven reasoning
- Incremental, experience-driven learning
- No task-specific modules

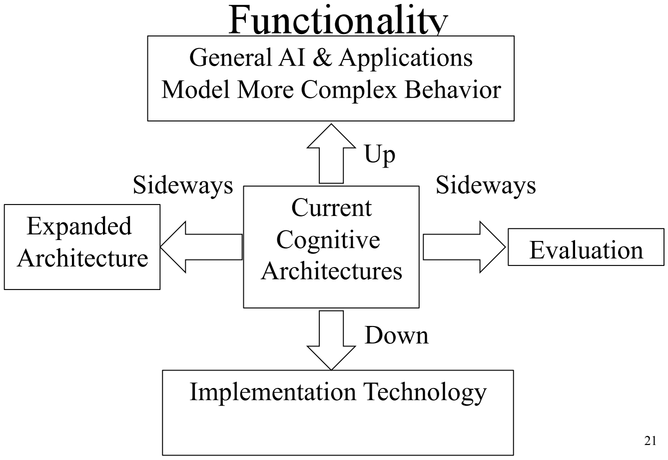
Changed

- Multiple long-term memories
- Multiple learning mechanisms
- Modality-specific representations & processing
- Non-symbolic processing
  - Symbol generation (clustering)
  - Control (numeric preferences)
  - Learning Control (reinforcement learning)
  - Intrinsic reward (appraisals)
  - Aid memory retrieval (WM activation)
  - Non-symbolic reasoning (visual imagery)

Cognitive Modeling with Soar

- Negatives
  - Unbounded working memory
  - Unbounded goal stack
- Positives
  - Incorporates many additional components
    - Episodic memory, mental imagery, appraisals
    - Easy to distinguish current situation from hypotheticals
  - At longer time scales, knowledge dominates behavior, not architecture
    - Indistinguishable from ACT-R for Pyramid problems & rat mazes
  - Easy and fast to explore complex behavior
    - ~1,000 times cognitive real time on simple tasks (no learning)
  - Scales up to large and long problems
    - > 8,000 rules for some tasks
    - > 5,000,000,000 decisions = 7 years real time (no learning)

Future of Cognitive Architecture: Functionality

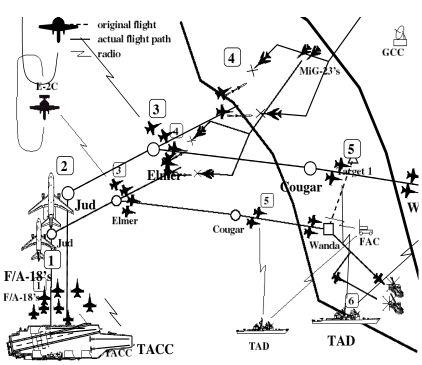


Up:
Toward General Intelligent Behavior

- Many more complex, knowledge-rich capabilities
  - Natural language, planning, spatial, temporal, meta-reasoning, reflection to improve performance, develop strategies
- Long-term behavior
  - Multiple interacting goals where history matters
- Interactions between those capabilities
  - Natural language interaction to aid planning
  - Planning during natural language generation
- Social agents that perform many different tasks
- Learning is everywhere (*wild learning*)
  - From imitation, instruction, experience, reflection, …
  - *Transition from programming to training, learning by experience*
- Real world applications
  - Intelligent assistants, robots, training & education, computer games
- Connect to the rest of psychology

TacAir-Soar

[1997]



Controls simulated aircraft in real-time training exercises (>3000 entities)

Flies all U.S. air missions

Dynamically changes missions as appropriate

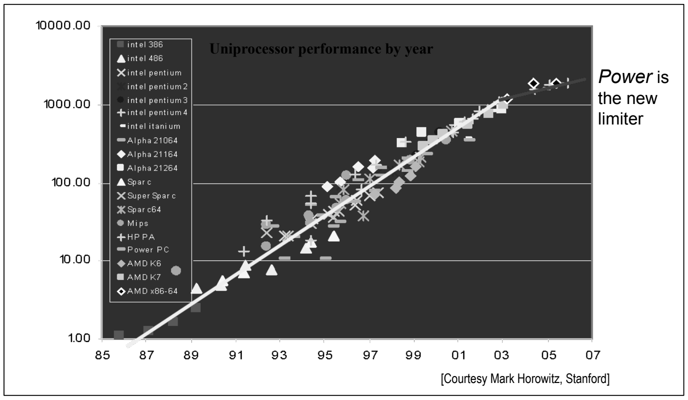
Communicates and coordinates with computer and human controlled planes

>8000 rules

Down - Functionality

- Challenge
  - *Real* systems run for days, weeks, months, …
  - *Real* applications will require *huge* knowledge bases
    - 8,000 rules in TacAir-Soar
    - 3,000,000 facts in OpenCyc
  - *Real* learning leads to lots of knowledge
  - Architectures assume constant time memory retrieval
- Common response:
  - "Don't worry, Moore's Law will save us."

Moore's Law



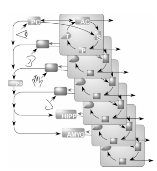
Down
Parallelism for Scaling

- Coarse-grain:
  - Multi-core & multi-processor clusters [Companions]
  - But Amdahl's law – still stuck with most costly process
- Fine-grain: New hardware architectures
  - FPGAs for memories
  - GPUs for imagery
  - ???
- Available technology can (should?) impact cognitive architecture design

26

Down - Modeling

- Extend models to the brain phenomena (fMRI)
  - Anderson et al. (problem solving)
  - Mitchell et al. (nouns)
- Neural models: Leabra (O'Reilly)
- Circuit models: Arbib, Grossberg, Ganger
- More brain structures and structure



27

Sideways
New Architectural (?) Capabilities

- Low-level vision & motor control (learning)
- Categorization, classification, …
- Development
- Prediction
- Emotion
- Drives and Motivation [origin of goals]
- Non-symbolic representations, reasoning, learning
  - Mental Imagery
  - Probability

28

Architecture vs. Knowledge?

- Advantages of architecture
  - Stable & efficient
  - Has access to architectural data
  - Available for all tasks independent of learning
- Advantages of knowledge
  - Can be task-specific
  - Can change over time
  - Simplifies architecture (RISC vs. CISC)

29

Sideways:
Evaluation of Functional Architectures

- No "gold standard" for comparison
- No common tasks or metrics
- No agreed upon evaluation methodology

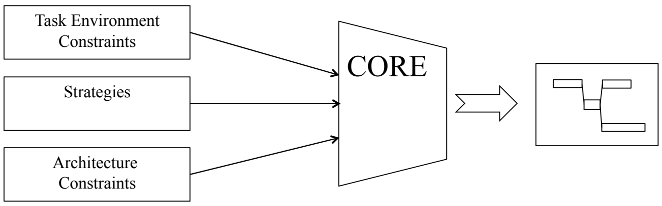
- Need series of tasks that require more and more capabilities
  - Where generality and learning are necessary

30

Sideways

New Approaches to Cognitive Modeling

- Cognitive Constraint Modeling: CORE
  - Howes, Lewis, Vera



31

Conclusion

- We are in a "Golden Age" of cognitive architecture
- Even after 25 years, lots of exciting research ahead
- Many challenges:
  - Performance:
    - Scaling to large knowledge while maintaining reactivity over long times
  - Applications
    - Putting cognitive architectures to work
  - Evaluation
    - How do we get people to work on common problems and compare
  - Consolidation
    - Bring together best ideas
  - Connect to the Brain
  - Connect to rest of AI and psychology

- Prediction
  - Prediction is the current/next "big" thing.

32

UTILITY **LEARNING**
+ DECISION MAKING

Short-term decay of production values for cognitive control

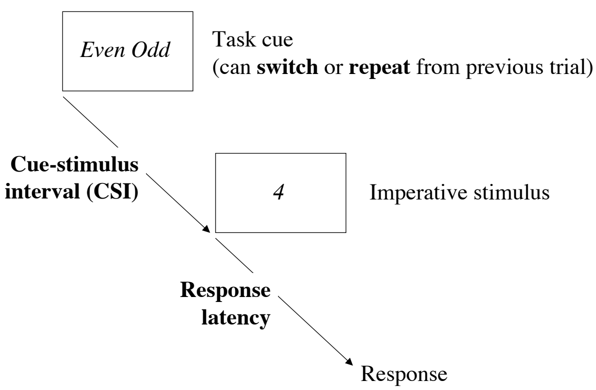
Erik M. Altmann
Michigan State University

Decay in cognitive control

(Altmann & Gray, in press, *Psychological Review*)

- Declarative control information
  - *Task codes*
  - Indicate what to do *now*, as opposed to last time
  - Decay limits proactive interference
- Procedural control information
  - Grabs system cycles to encode a task code
  - Those cycles are lost to other processes
  - But decay prevents lockout (see Salvucci & Taatgen, 2007)

Trial line



Encoding productions

identify-cue

When a task cue has onset, retrieve its meaning, and use it to create a new task code in episodic memory

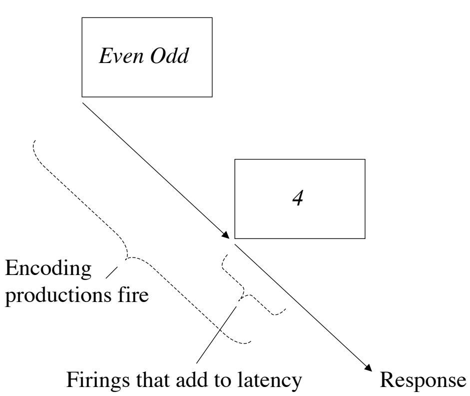
Source of switch cost: repeat cues are repetition-primed

activate-task-code

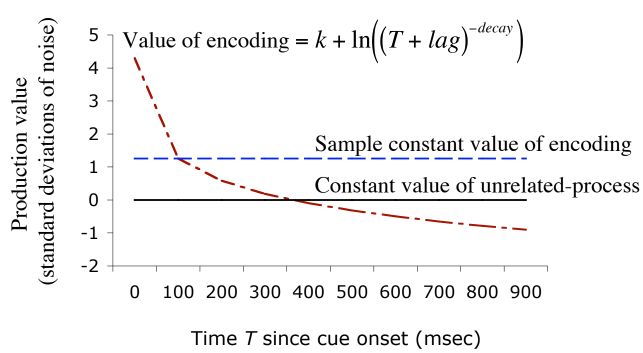
When the new task code is not active enough yet, increment its activation (fires iteratively)

- Values start the trial high, at cue onset
  - Decay during cue-stimulus interval
  - Relative to the *unrelated-process* production

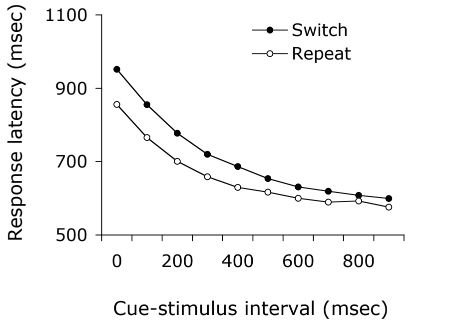
Trial line



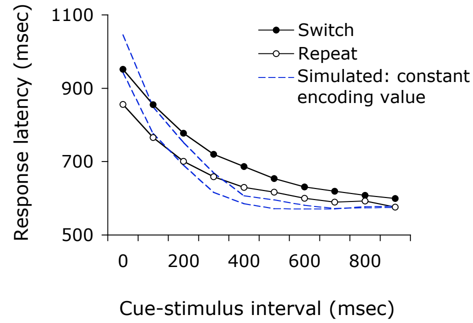
Production values



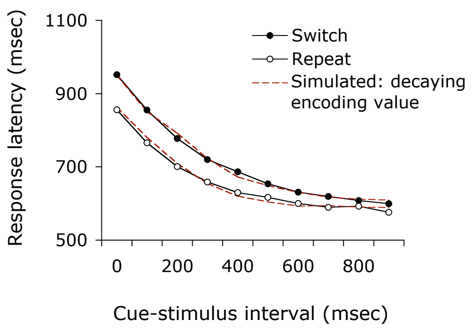
Empirical results



Simulation results



Simulation results



A general pattern

- Main effect of cue-stimulus interval is too gradual to fit with constant production values
  - Seems that either production values decay, or the production approach is wrong, here
- May offer an account of alertness effects (e.g., Luce, 1986; Posner & Boies, 1971)
  - Aversion to maintaining full preparedness = architectural decay of encoding productions

Implications

- A basic mechanism of endogenous control:
  - Spike the value of encoding productions
  - Basic: Applies to any input
  - Output (task code) projects control into the future
- Decay then frees the central processor
  - Automatically, unlike inhibition; safety feature
  - Quickly, so other processes have access
  - Preserves cognitive flexibility and reactivity

How a Modeler's Conception of Rewards Influences a Model's Behavior: Investigating ACT-R 6's Utility Learning Mechanism

Christian P. Janssen[1,2], Wayne D. Gray[2] and Michael J. Schoelles[2]
cjanssen@ai.rug.nl, grayw@rpi.edu, schoem@rpi.edu
[1]: Department of Artificial Intelligence, University of Groningen
[2]: Cognitive Science Department, Rensselaer Polytechnic Institute

There's a lot to Learn about Utility Learning

Temporal difference learning has recently been introduced as the new utility learning mechanism in ACT-R 6 (e.g., Fu & Anderson, 2004). Common practices for using it still have to emerge. In this study we take a first step by investigating two critical aspects of utility learning: the location and size of rewards.

As a case study we use the Blocks World task (Gray et al., 2006). In this task subjects have to copy a pattern of eight blocks, depicted in a target window, by moving blocks from a resource window to a workspace window. Information in each of the windows is covered by a gray rectangle and only becomes available when subjects move the mouse cursor into the window area. In addition, the information in the target window only becomes available after waiting for a lockout time of 0, 400 or 3200 milliseconds (manipulated between subjects). As the size of the lockout time increases, subjects tend to study and place more blocks per visit to the target window.

Previous attempts in modeling the task in ACT-R 5 did not provide good fits to human data. Analysis indicated that this might be because ACT-R 5's expected value equation can only handle binary feedback (Gray, Schoelles, & Sims, 2005). As ACT-R 6's utility learning mechanism is not limited to binary feedback, its use seems more promising.

ACT-R 6 Models of the Blocks World task

The ACT-R 6 models of the Blocks World task are kept close to the ACT-R 5 models, but also take benefit of new ACT-R 6 features. Crucially, the model has eight encode-x productions that determine its strategy: the number of blocks (x, ranging between 1 and 8), which the model will study during a visit to the target window. Using utility learning, the model tries to learn which encode-x strategies/productions lead to the best overall performance.

We tested six different models that have the same parameter settings and production rules, but differ in two aspects: the location of the reward and the size of the reward. The location of the reward can either be *once* per trial after completing the whole trial or *each* time that the model has tried to place blocks in the workspace window and either finishes the trial, or starts studying blocks in the target window again. The location of the reward is important for utility learning, as the utility of a production converges towards the size of the experienced reward minus the average time between the firing of that production and the time the reward was triggered (Anderson, 2007).

The second manipulation between models is the size of the rewards. Due to space limitations we will not describe each of these models in detail, but fundamentally the manipulations differ in what aspect of the task the model conceptualizes as a reward. On the one hand, a reward can be expressed in *how good* the model performs the task itself: how many blocks does it place after a visit to the target window, and how many blocks does it study but forget? Different models have different reward functions, but in general the rewards range between -8 (all blocks studied, none placed) and 8 (all blocks placed). A totally different conception is to express rewards in terms of *how fast* the model performs the task (or specific parts of it). Different models have different reward functions, but in general the rewards are negative: the more time the model spends on the trial (or on specific parts of it), the more negative the reward is. In this case rewards range between 0 and about -80.

Results and Discussion

As shown in our example above, the modeler's conception of the rewards of a task has a big influence on the reward size. The reward size has a big influence on the utility of productions, and this has a big influence on the behavior of a model. In the end, the modeler's conception of rewards has a big influence on the model's behavior.

In our simulations of the Blocks World task, each model behaves different from others. Despite the broad exploration of model types, none provides a good fit to the human data. Some models seem to be at least as good as the best ACT-R 5 models (Gray et al., 2005). Unlike the ACT-R 5 models, these models do not require changes to the architecture. They require a different conception of what a reward is.

References

Anderson, J. R. (2007). How Can the Human Mind Occur in the Physical Universe? New York: Oxford University Press.

Fu, W. T., & Anderson, J. R. (2004). Extending the Computational Abilities of the Procedural Learning Mechanism in ACT-R. Proceedings of the 26th annual meeting of the cognitive science society, 416–421.

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

Gray, W. D., Sims, C. R., Fu, W. T., & Schoelles, M. J. (2006). The soft constraints hypothesis: A rational analysis approach to resource allocation for interactive behavior. Psychological Review, 113(3), 461-482.

Instance and Strategy ACT-R models of choice in a dynamic control task: a model comparison story

Varun Dutt (vdutt@andrew.cmu.edu)
& Cleotilde Gonzalez (coty@cmu.edu)
Dynamic Decision Making Laboratory
Carnegie Mellon University
Pittsburgh, PA 15213 USA

In ACT-R, choice has traditionally been modeled by representing strategies into productions. Each production in ACT-R has a utility value and ACT-R learns to choose the production with the maximum utility, among a possible set of applicable productions by using the conflict resolution and reinforcement learning procedures. A second approach to modeling choice in ACT-R is instance-based, by representing alternatives into chunks. Each chunk in ACT-R has an activation value, and ACT-R learns to choose the chunk with the highest activation, determined by frequency, recency, decay, and similarity of the chunk to the goal. These effects of chunk activation clearly reproduce characteristics of human memory.

We implemented two models of choice using Strategy-Based Learning (SBL) and Instance-Based Learning (IBL) approaches in ACT-R. The models interacted in real-time with a dynamic control task, Dynamic Stocks and Flows (DSF). The goal in DSF is to maintain the level of a stock (water in a tank) at a target level through repeated trials while external flows remove or add water into the tank. The two models were compared in different dimensions: (1) Applicability: how well each model fits human data; (2) Robustness: how well each model with its own parameters with which it fits human data, is able to fit a new data set; and (3) Adaptability: how well each model is able to reproduce the way humans learn in one scenario of the task (i.e. training) and transfer to a new scenario.

The results demonstrate that both models fit human data well. However, the results also show that the IBL model is more robust and adaptable than the SBL model. This exercise opens up for a discussion of how we evaluate the goodness of our models, and the advantages and disadvantages of traditional representations of choice in ACT-R.

The Sudoku Model: A Dynamic Decision Maker

Michael J. Schoelles, Wayne D. Gray and Hansjörg Neth
Cognitive Science Department, Rensselaer Polytechnic Institute

Routine Behavior and Decision Making.

While decision-making is often thought to be a product of high-level cognition, we contend that routine interactive behavior at the 300–1000 ms time scale can affect human decisions made at longer time scales. The ACT-R 6.0 Sudoku model is being developed to explore how resource allocation strategies (Gray et al., 2006) used in routine interactive behavior influence–decision-making strategies. The model is being developed in conjunction with a series of empirical studies.

Sudoku is a 9-row by 9-column matrix subdivided into 9 3-row by 3-column matrices (called boxes). Each row, column and box must be filled with numbers 1 through 9 with each number appearing only once in each row, column, and box. The puzzle starts out with some of the cells filled in and the solver must fill in the rest. Sudoku is well-suited to study the impact of interactive behavior on decision making, since it requires many decisions based on the interactions of human memory and vision. It is a popular game so the training time is minimal, and it is repetitive. The repetitive nature is important for electroencephalography (EEG) data collection, which we will do in future studies. Another advantage of Sudoku is that the level of difficulty of different games can be determined.

Sudoku has been well studied in the Artificial Intelligence (AI) community as a constraint satisfaction problem or logic problem (Simonis 2005). The solutions methods studied include the constraint of difference method, integer programming and graph theory. Little attention has been given to the study of the psychology of Sudoku, except for Lee et al. (2006). They report three experiments to determine what types of deductions people use to solve certain Sudoku configurations.

Empirical Studies

We are conducting a series of studies. In the first study, participants played a total of eight games each. The first three were practice games, the next three were games where the difficulty was manipulated and the last two were normal games. The difficulty manipulation was covering up 0, 6 or 8 boxes. The manipulation required clicking on the box cover to remove it and see the cells in the box. This added a motor component to the task plus increased the amount of interaction between memory and vision. All the games in this study were rated as easy, which means that the game can be solved through constraint satisfaction without any search. We have also written an AI program to solve any Soduku puzzle and verified that these games require only propagation of constraints. This program can determine the level of difficulty of any Sudoku game in terms of the number of constraints propagated and the amount of search required.

The Model

The Sudoku model shows how the difficulty manipulation of the first empirical study influences the decision-making strategy. The current status of the model is that the first version of the model has been written and is being tested on different games. The current model can solve very easy and easy puzzles. The time the model takes is longer than humans but the model can make errors. The current model has been developed only for the manipulation where 0 boxes are covered.

To describe the operation of the model, the term unit will be used to refer to a row, column, or box. The current model scans the puzzle looking for a unit with 4 or fewer empty cells. For each empty cell in the unit it determines what are the possible values and encodes them. To determine a value the model scans the row, column and box for every possible value. If a value is not encountered it can encode it as a possible value. If the model has exhausted all possible values and recalls only one possible value then it will enter the recalled value into the cell. The model uses simple heuristics, for example, if there is only 1 possible value then only the unit needs to be scanned to determine the value. The current model always looks at a cell to determine its value if one exists. A change that we are currently implementing to interact more with memory by trying to recall a cell's value instead of looking at it.

The next steps in the model development will be to improve the match of model performance to human performance in terms of time and error rates and to solve the puzzle when 6 or 8 boxes are covered. Since the task is very visual, the number of eye movements and time to move visual attention are crucial to model performance. Analysis of eye data collected during the empirical study will be used to improve the design of the model.

References

Gray, W. D., Sims, C. R., Fu, W. T., & Schoelles, M. J. (2006). The soft constraints hypothesis: A rational analysis approach to resource allocation for interactive behavior. Psychological Review, 113(3), 461-482.

Lee, N.Y. Louis, Goodwin, G. P. & Johnson-Laird, P.N. (2006). The Psychology of Su Doku Problems. In Sun, R., & Miyake, N. (Eds.). Proceedings of the 28th annual meeting of the Cognitive Science Society, 2543.

Simonis, H. (2005). Sudoku as a Constaint Problem. In Hnich, B., Prosser P., and Smith B., (eds.),Proceedings 4th Int. Works. Modelling and Reformulating Constraint Satisfaction Problems (pp. 13–27).

Putting New Wine into Old Bottles: On the role of markers, instances and utilities in the Iowa Gambling Task

Danilo Fum (fum@units.it)
Antonio Napoli
University of Trieste
Trieste, Italy, I-34127

While the Iowa Gambling Task has been traditionally considered as the main experimental testbed for the Somatic Markers Hypothesis put forward by Antonio Damasio and coworkers, it can be included among a set of experimental paradigms—comprising, for instance, probability matching, *n*-bandits, and dynamic choice tasks—which require a series of iterative decisions, and in which the only (or the most important) information available to the decision maker is constituted by the outcome of previous selections.

Two main kinds of explanation have been provided for the experimental findings obtained with these tasks: the first one relying on the retrieval of previous outcomes from memory to drive new decisions, the second one based on the progressive learning of the utilities of different choice options.

In the paper we present the results of three experiments carried out with the Iowa Gambling Task which clearly highlight the critical role played by the frequency of contingent events (e.g., losses in the case in which each selection is always associated with a gain) in comparison with alternative factors such as the dichotomy between immediate outcome and long term results privileged by the proponents of the Somatic Markers Hypothesis.

In the first experiment—which adopted the classic IGT scenario in which each choice gave always rise to a win, while losses were contingent to the different selections—variations in the expected value of possible options did not influence the behavior of participants who were, on the contrary, extremely responsive to the frequency of losses associated with each choice. Similar results were obtained in the second experiment in which each selection was invariably associated with a loss, while wins were the contingent event.

A procedural model, relying essentially on the new ACT-R utility learning mechanism, was capable of explaining these findings. The model made also some interesting and unexpected predictions in the case in which the frequency of the contingent event was kept constant. According to the model, with low-frequency events the participants should be unable to discriminate among the expected value of the different options while they should be able to concentrate their selections on the most profitable ones in case the high-frequency contingent events. These predictions were confirmed in the third experiment.

The relevance of this work for the Somatic Markers Hypothesis, on one hand, and for the instance vs. productions issue, on the other, is discussed, and some speculation about the neurobiological and the neuropsychological ramifications of the proposed mechanism are made.

HUMAN **COMPUTER** INTERACTION

Using ACT-R for Rapid Prototyping and Evaluation of In-Vehicle Interfaces

Dario D. Salvucci (salvucci@cs.drexel.edu)
Department of Computer Science, Drexel University
3141 Chestnut St., Philadelphia, PA 19104, USA

Driver distraction from in-vehicle secondary tasks has become a great concern for researchers and citizens alike. While the community has traditionally used experimental testing (in driving simulators or on real roads) to evaluate the distraction potential of new interfaces and tasks, such testing is typically very time-consuming and very expensive. Thus, there is much to gain from a system that can provide predictions of distraction potential — good engineering approximations that can guide design and development of in-vehicle interfaces.

In this talk I will discuss a recent major redevelopment of Distract-R (Salvucci et al., 2005), a system that allows designers to rapidly prototype and evaluate new in-vehicle interfaces. The core engine of the system relies on a rigorous cognitive model of driver behavior (Salvucci, 2006) implemented in the ACT-R cognitive architecture (Anderson et al., 2004). When integrated with other models of secondary-task behavior, the combined model can generate predictions of driver performance and distraction (see Salvucci, 2001; Salvucci, 2007). Distract-R also utilizes Threaded Cognition theory (Salvucci & Taatgen, 2008) to predict multitasking in the interleaved execution of the primary and secondary tasks.

Distract-R allows a designer to prototype basic interfaces, demonstrate possible tasks on these interfaces, specify relevant driver characteristics and driving scenarios, and finally simulate, visualize, and analyze the resulting behavior as generated by the cognitive model. To date we have applied the system to several modeling studies to explore the space of possible interfaces and to validate various aspects of the system. The paper includes three modeling studies that demonstrate the system's ability to account for various aspects of driver performance for several types of in-vehicle interfaces.

Acknowledgments

This work was supported by Office of Naval Research grant #N00014-03-1-0036 and National Science Foundation grant #IIS-0426674.

References

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

Salvucci, D.D. (2001). Predicting the effects of in-car interface use on driver performance: An integrated model approach. *International Journal of Human-Computer Studies*, *55*, 85-107.

Salvucci, D.D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, 48, 362-380.

Salvucci, D. D. (2007). Integrated models of driver behavior. In W. D. Gray (Ed.), *Integrated Models of Cognitive Systems* (pp. 356-367). New York: Oxford University Press.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, *115*, 101-130.

Salvucci, D. D., Zuber, M., Beregovaia, E., & Markley, D. (2005). Distract-R: Rapid prototyping and evaluation of in-vehicle interfaces. In *Human Factors in Computing Systems: CHI 2005 Conference Proceedings* (pp. 581-589). New York: ACM Press.

Towards a Tool for Predicting Goal-directed Exploratory Behavior

Leonghwee Teo Bonnie E. John
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh PA 15213 USA
{teo, bej}@cs.cmu.edu

MOTIVATION

MESA (Miller and Remington, 2004), ACWW (Blackmon, Kitajima and Polson, 2005) and Bloodhound (Chi, et al., 2003) predict user search through webpages. While their predictions correlated with human data, none of them capture or consider the spatial positions of links on a webpage in their predictions. However, the positions of links may matter because a link not seen and evaluated is not likely to be chosen, while competing links seen before the correct link might be chosen instead. Choosing incorrect links can greatly increase the number of interaction steps and time spent exploring the wrong branches in a large and complex website. To investigate this, we reanalyzed the data from ACWW Experiment 2 and found that participants indeed were more likely to choose the correct link if it appeared in the left column of the webpage than if it appeared in the right column, suggesting a predominant left-to-right visual scan pattern by those participants. Unfortunately, MESA, ACWW and Bloodhound would have predicted no difference.

SOLUTION APPROACH

More accurate predictions can inform designers if a user-interface (UI) design might help or hinder successful exploration. Towards this end, we are developing CogTool-Explorer and used it to model ACWW Experiment 2 (Teo and John, 2008). Using the ACT-R architecture (Anderson and Lebiere, 1998), CogTool-Explorer employs a process model, SNIF-ACT 2.0 (Fu and Pirolli, 2007), to evaluate links one step at a time. The model evaluates the semantic relatedness of a link to the goal, then decides to either satisfice and choose the best link read so far on the webpage, or continue to look at and read another link. Because the model may satisfice and not evaluate all links on a webpage before making a choice, the order in which links are evaluated affects its choices. CogTool-Explorer uses ACT-R's vision system and a visual search strategy adapted from the Minimal Model of Visual Search (Halverson and Hornof, 2007). The strategy starts in the upper-left corner and proceeds to look at the link closest to the model's current point of visual attention without replacement. For visual search to be meaningful, CogTool-Explorer uses a spatially accurate model of the UI (on-screen position, dimension and text label of every link on the webpage) created with CogTool (John, Prevas, Salvucci and Koedinger, 2004). CogTool converts this representation into an ACT-R *device model*, with which the process model can interact. Given the device model, CogTool-Explorer moves its visual attention to a link, encodes the text label, and evaluates its semantic relatedness to the goal. When the model decides to satisfice, it looks back at the best link, moves a virtual mouse pointer over it and clicks on it. Each run of the model can be different because of noise, thus, the path of the model through the webpages on each run is analogous to predicting the exploration choices of a single human trial.

RESULTS

We compared human data on 22 tasks in ACWW Expt. 2 to predictions by CogTool-Explorer and ACWW, a publically accessible tool. We set CogTool-Explorer to use the same semantic relatedness scores as ACWW, and fitted a parameter *k*, the model's "readiness" to satisfice, to human data. Figure 1 shows CogTool-Explorer's predictions aligned with human data, while ACWW predicted no significant difference. This more accurate prediction is a direct consequent of using a satisficing process model with a visual search strategy on a spatially accurate device model in concert.

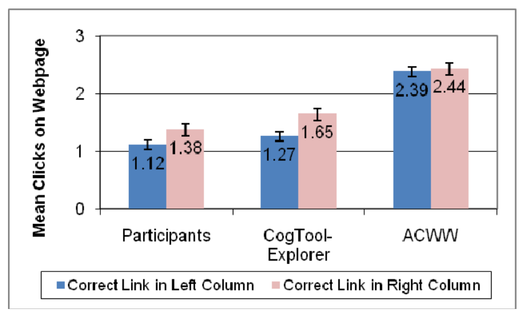


Figure 1. Mean clicks on webpage to click on correct link over 22 search tasks, by target column (Std. Err. shown)

REFERENCES

Anderson, J. R., and Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Blackmon, M. H., Kitajima, M., and Polson, P. G. (2005). Tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '05. ACM, 31-40.

Chi, E. H., Rosien, A., Supattanasiri, G., Williams, A., Royer, C., Chow, C., Robles, E., Dalal, B., Chen, J., and Cousins, S. (2003). The bloodhound project: automating discovery of web usability issues using the InfoScent™ simulator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '03. ACM, 505-512.

Fu, W.-T., and Pirolli, P. (2007). SNIF-ACT: A cognitive model of user navigation on the World Wide Web. *Human-Computer Interaction, 22,* 355-412.

Halverson, T. and Hornof, A. J. (2007). A minimal model for predicting visual search in human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. 431-434.

John, B. E., Prevas, K., Salvucci, D. D., and Koedinger, K. (2004). Predictive human performance modeling made easy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. 455-462.

Miller, C. S., and Remington, R. W. (2004). Modeling Information Navigation: Implications for Information Architecture. *Human-Computer Interaction, 19*, 225-271.

Teo, L. and John, B. E. (2008). Cogtool-Explorer: towards a tool for predicting user interaction. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*. CHI '08. ACM, 2793-2798.

Building an SGOMS model (Sociotechnical GOMS) using ACT-R: Issues with Cognitive Modelling and Macro Cognition

Robert L. West (robert_west@carleton.ca)
Department of Psychology
Carleton University
British Columbia, Canada

Macro cognition is an area of study concerned with understanding higher level, complex, real world human cognition and behaviour. People involved with macro cognition refer to cognitive psychology and other experimentally based approaches as micro cognition. Cognitive modelling architectures, such as ACT-R, are based on findings from micro cognition but are increasingly being used to model higher level, complex, real world behaviours. Therefore, cognitive architectures can be viewed as scaling micro cognition up to the level of macro cognition. West and Nagy (2007) have agued that specific theories about how micro cognition scales up are important. To this end they proposed a system called SGOMS (Sociotechnical GOMS), which is a theory about how the mechanisms involved in GOMS modelling get scaled up to deal with complex, social/team orientated behaviours. In this presentation I will discuss how modelling SGOMS using ACT-R provides theoretical constraints and leverage. Some comparisons to studying macro cognition in the absence of a cognitive model may be made.

VISUAL **PERCEPTION**
+ **SKILL** ACQUISITION

ACT-R's Answers to Six Questions about Visual Routines

Scott Douglass (scott.douglass@mesa.afmc.af.mil)
Air Force Research Laboratory, 6030 S. Kent St., Mesa, AZ 85212 USA

Motivation

Each ACT-R model that uses the vision module to identify visual locations and move attention, tells a story about how central cognition interacts with a task environment through attention. ACT-R models that use the vision module to interact with a task can: (a) encode the entire visual context into a representation and then never revisit the world; or (b) encode aspects of the visual context "on demand" and remain situated in the world. On the one hand, models that attempt to comprehensively encode and represent the visual context must answer the question "How can capacity limits and the impact of sudden context changes be avoided?" On the other hand, models that take an incremental approach to information acquisition must repeatedly answer a different challenging question; "What determines the destination of the next attention shift?"

An ACT-R model that incrementally encodes aspects of the visual context as it completes a task states what determines the destination of each attention shift in a single task. A framework in ACT-R that generally explained how task-relevant aspects of the environment are encoded "on demand" would be preferable. The presentation will describe such a framework.

Approach

According to Bajcsy (1988), vision is not passive and static; instead it is dynamic, probing, exploratory, and guided by the interleaving of visual processing and context assessment. Bajcsy proposed that rather than trying to represent the entire visual field using a complete amodal representation scheme, vision allows the environment to be its own representation. Rather than representing the world *before* cognition proceeds, the visual system encodes parts of the visual field *as* cognition proceeds. Active vision is the process through which the visual system selectively encodes and represents aspects of the world as they are required by visual and central cognitive processes. On-going activities and current goals define the situational demands that frame intelligent visual information seeking during active vision.

In order to avoid computational complexity, the visual system employs composable primitive operations during visual recognition (Ullman, 1983). These elemental operations are sequenced into *visual routines*. Researchers attempting to model active vision (Ballard & Rao, 1995; Hayhoe, 2000; Rao & Ballard, 1995; Sprague, Ballard, & Robinson, in press) using visual routines have addressed only subsets of six critical questions that guide the application of a visual routines framework:

1. How are visual routines represented?
2. What are the primitive visual routines underlying active vision?
3. How are macro-behaviors composed and learned?
4. How are visual routines and macro-behaviors arbitrated?
5. How are visual routines and macro-behaviors selected and executed?
6. What integrates behavior during active vision?

The presentation will show that an active vision framework in ACT-R can explain the incremental use of attention in various tasks and contexts while answering *all* six questions. The active vision framework is interesting because it: (a) illustrates how the major components of ACT-R combine to enable active vision; (b) describes a type of visual learning based on dynamic pattern matching, composition, and proceduralization; and (c) suggests how top-down and bottom-up processes contribute to dynamic, probing, exploratory, and goal-driven active visual cognition.

The views expressed in this paper are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE , 76* (8), 996-1005.

Ballard, D. H., & Rao, R. (1995). A computational model of human vision based on visual routines. In G. Sagerer, S. Posch, & F. Kummert (Ed.), *17th DAGM-Symposium* (pp. 646-667). Berlin: Springer.

Hayhoe, M. (2000). Vision using routines: A functional vision account. *Visual Cognition , 7* (1), 43-61.

Rao, R. P., & Ballard, D. H. (1995). An active vision architecture based on iconic representations. *Artificial Intelligence Journal , 78*, 461-505.

Sprague, N., Ballard, D., & Robinson, A. (in press). Modelind attention with embodied visual behaviors. *ACM Transactions on Action and Perception* .

Ullman, S. (1983). *Visual routines.* A.I. Memo (AIM-732), 1-65, Artificial Intelligence Laboratory: Massachusetts Institute of Technology.

Modeling eye-movement patterns in the Flight Management Task: combining bottom-up and top-down vision

Niels Taatgen and Daniel Dickison

In the discussion of our model of the Flight Management Task (Taatgen, Huss, Dickison & Anderson, in press), we argue that task control is largely bottom-up, that is, the decision of the next step to take is prompted by what is perceived in the environment, and not so much by internal planning. The shortcoming of that model is that perception is grossly oversimplified: the state of the display is summarized in a single attribute-value pair. In our new model, we have mended this short-coming, and have it gather information from the display using ACT-R's perceptual system. Gathering information is partially a bottom-up process: task-independent production rules roam the display and process information on it. Consistent with theories of visual attention like Rensink (2007), this process retains very little information, but it does retain a map of "where things are". A supplemental, more top-down, visual process can use this map to find information it needs. This combination produces model behavior in which information on the display is attended exactly when it is needed, making it unnecessary to build an internal representation of the world.

We use this model to explain the data from an eye-movement experiment we have done with the FMS task. The eye-movement data show that subjects indeed employ a "just-in-time" method of finding information, and the model is accurate in fitting the data. The model is implemented in ACT-R/Lisa, an extension (under construction) of ACT-R that makes it easier to model learning from instructions. The visual strategy from the FMS model will also be built into ACT-R/Lisa, making it easy to reuse.

References

Rensink, R.A. (2007). The modeling and control of visual perception. In: W. Gray (Ed.), *Integrated Models of Cognitive Systems* (pp. 132-148). Oxford: OUP.

Taatgen, N. A., Huss, D., Dickison, D. & Anderson, J. R. (in press). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology: General*.

Modeling eye-movement patterns in the FMS task

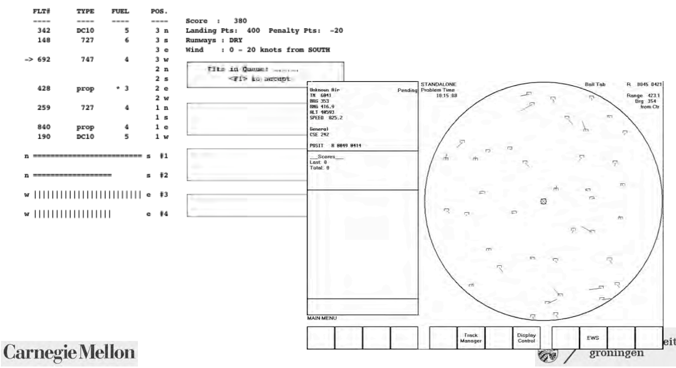
Combining bottom-up and top-down vision

Niels Taatgen and Daniel Dickison

Carnegie Mellon

rijksuniversiteit groningen

Vision in ACT-R



Carnegie Mellon

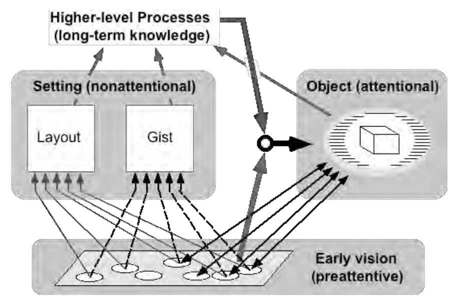
Problems

- Vision in ACT-R is too “programmed”
- More autonomous support of the visual system to help processing
- There is a lot of information on the screen, and it is unlikely that we internalize all that information in memory

Carnegie Mellon

rijksuniversiteit groningen

Rensink’s triadic architecture



Carnegie Mellon

rijksuniversiteit groningen

- Rensink argues that it is often faster to recover information from the world than from memory
- Others have made similar arguments, e.g. Tetris

Carnegie Mellon

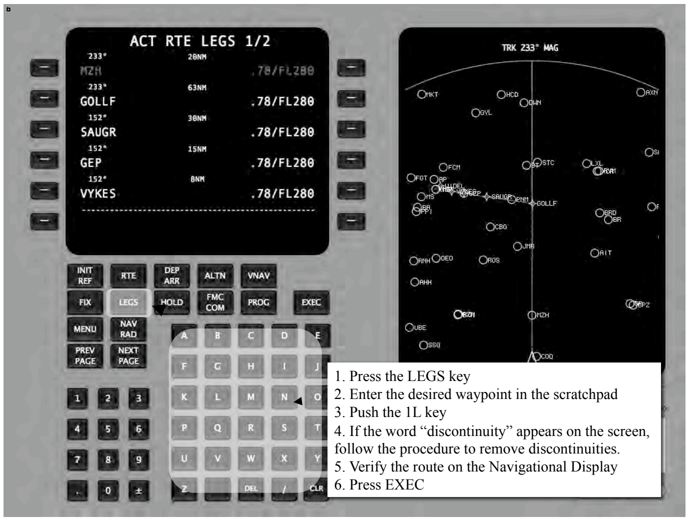
rijksuniversiteit groningen

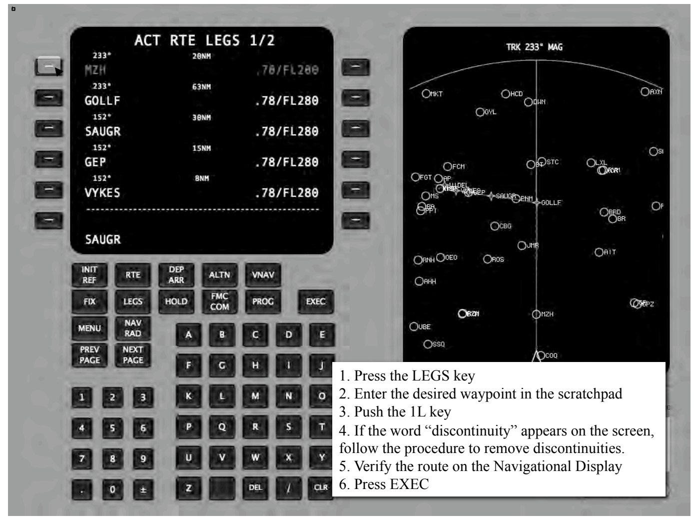
Task: Flight Management System

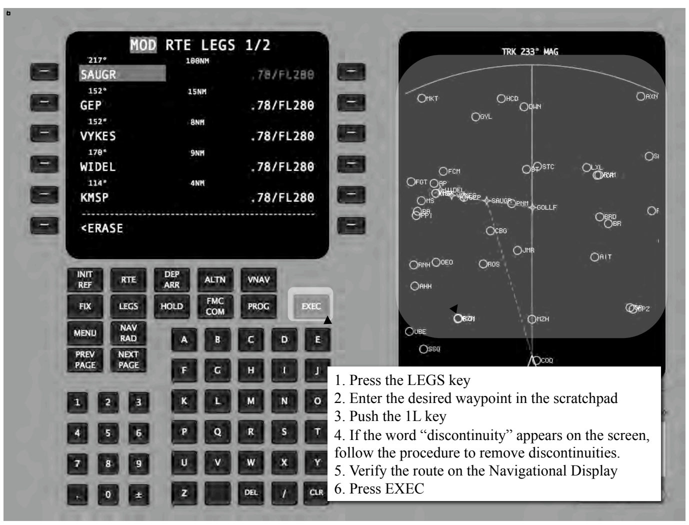
- We have a model of this task that bypasses visual perception (Taatgen et al., in press)
- The goal of the current project is to model the perception part in order to model eye-movements

Carnegie Mellon

rijksuniversiteit groningen







Implementation in ACT-R

- Bottom-up (goal-independent) productions scan items in the visual field.
- The last two of these are stored in a buffer, but older objects are lost.
- However, everything that has been attended gets an entry in an internal visual map
- When cognition needs a particular piece of information, and its location is in the visual map, an eye-movement is initiated to recover it.

Carnegie Mellon

rijksuniversiteit groningen

1. When you are not yet on the LEGS page, **Press the LEGS key** to get to the LEGS page.
2. When you are on the LEGS page, **Enter the desired waypoint in the scratchpad** so you can put it into the flight plan
3. When you have your new destination in the scratchpad, **Push the 1L key** to move the waypoint from the scratchpad into the first line of the flightplan.

Page Init | Press "Legs" | Page Legs

Page Legs | Press line with dest | Dest in Scratch

Dest in Scratch | Press line 1 | Mod done

Mod done | Check ND | Check done

Check done | Press "Exec" | Goal achieved

ACT-R/Lisa

- Page Legs | Press "Legs"

Page Legs, Scratch empty | Press line with dest

Scratch destination | Press line 1

Nav ok, - L2 Discontinuity | Press "Exec"

Carnegie Mellon

rijksuniversiteit groningen

ACT-R version

PAGE LEGS

L1 minde
L2 acx
L3 ksfo
L4 fgx

NAV DONE

SCRATCH EMPTY

KEYBOARD X

DESTINATION minde

state212
isa state
page nil
l1 nil
l2 nil
scratch empty
nav done
keyboard nil
destination nil

rijksuniversiteit groningen

Bottom-up vision

```
(p* attend-anything
  =visual-location>
    isa visual-location
  ?visual>
    state free
    buffer empty
==>
  +visual>
    isa move-attention
    screen-pos =visual-location
  =visual-location>)
```

```
(p* attended-item-to-state
  =visual-location>
    isa visual-location
    content =slot
  =visual>
    isa text
    value =text
  =state>
    isa state
==>
  =state>
    =slot =text)
```

```
(p continue-bottom-up-vision
  ?visual-location>
    buffer empty
    - state error
  ?visual>
    state free
    buffer empty
==>
  +visual-location>
    isa visual-location
    :nearest current
    :attended nil)
```

Carnegie Mellon

rijksuniversiteit groningen

ACT-R version

PAGE LEGS

L1 minde
L2 acx
L3 ksfo
L4 fgx

NAV DONE

SCRATCH EMPTY

KEYBOARD X

DESTINATION minde

state212
isa state
page nil
l1 nil
l2 nil
scratch empty
nav done
keyboard nil
destination minde

rijksuniversiteit groningen

ACT-R version

PAGE LEGS

L1 minde
L2 acx
L3 ksfo
L4 fgx

NAV DONE

SCRATCH EMPTY

KEYBOARD X

DESTINATION minde

state212
isa state
page nil
l1 nil
l2 nil
scratch nil
nav done
keyboard nil
destination minde

rijksuniversiteit groningen

PAGE LEGS

L1 minde
L2 acx
L3 ksfo
L4 fgx

NAV DONE

SCRATCH EMPTY

KEYBOARD X

DESTINATION minde

state212
isa state
page nil
l1 nil
...

Retrieved operator

- Page Legs | Press "Legs"

+visual-location>
content legs

Carnegie Mellon

rijksuniversiteit groningen

Operator retrieval

```
(p retrieve-operator
   =goal>
     isa goal
     task =task
     state =s
     action nil
   ?retrieval>
     state free
     buffer empty
==>
   +retrieval>
     isa operator
     task =task
     state =s
     :recently-retrieved nil)
```

```
(p* revisit-location-on-precondition
   =goal>
     isa goal
     task =task
   =retrieval>
     isa operator
     pre =slot
     val =value
   =state>
     isa state
     =slot nil
==>
   +visual-location>
     isa visual-location
     content =slot
```

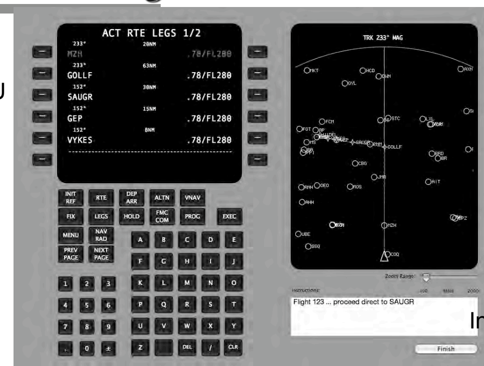
Carnegie Mellon — rijksuniversiteit groningen

Four Regions of Interest

CDU

NAV

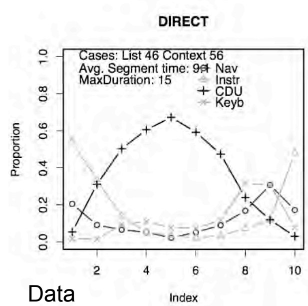
Key board



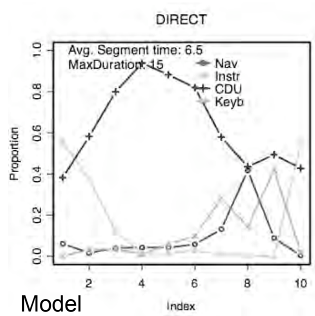
Instructions

Carnegie Mellon — rijksuniversiteit groningen

Simple direct-to trials



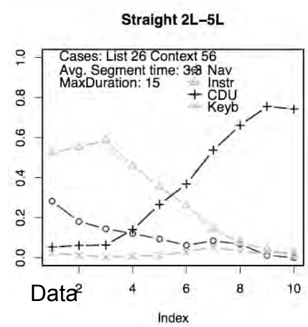
Data



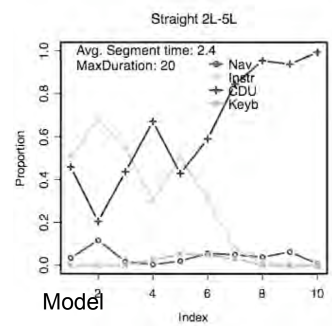
Model

Carnegie Mellon — rijksuniversiteit groningen

Some detail: start of a trial



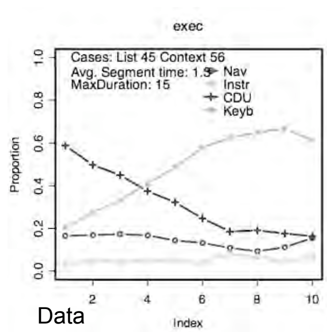
Data



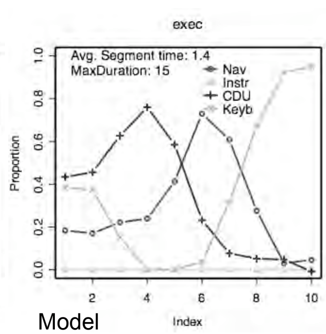
Model

Carnegie Mellon — rijksuniversiteit groningen

Before pressing Exec



Data



Model

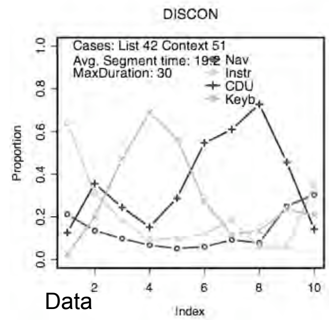
Carnegie Mellon — rijksuniversiteit groningen

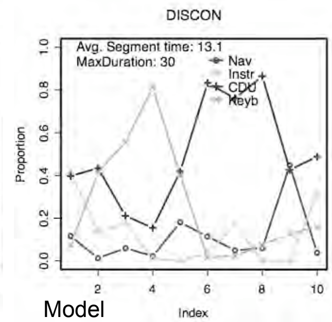
Discontinuity problems

- Type the first waypoint
- Then press 1L key
- See that there is a discontinuity
- Enter a second waypoint
- Put this second waypoint into the discontinuity

Carnegie Mellon — rijksuniversiteit groningen

Discontinuity trials

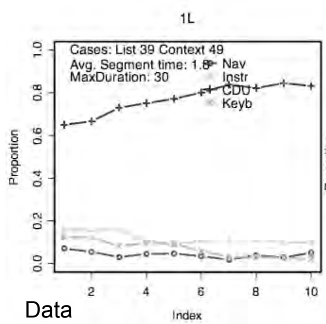


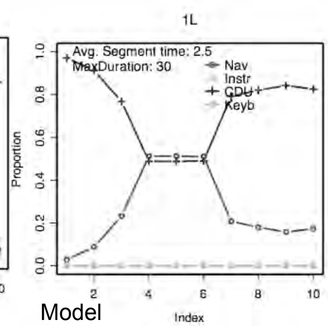


Carnegie Mellon

rijksuniversiteit groningen

Details: sometimes the model is wrong

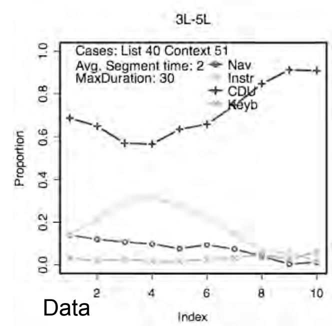


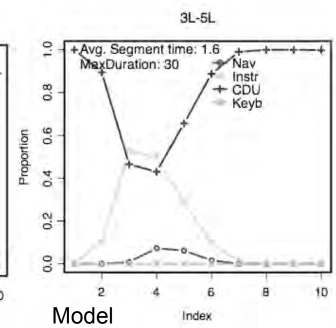


Carnegie Mellon

rijksuniversiteit groningen

Reading the second waypoint in the instructions

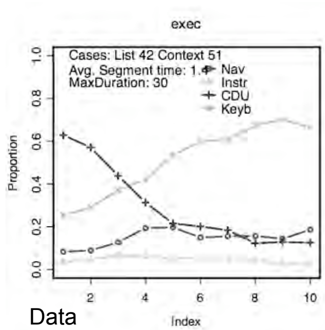


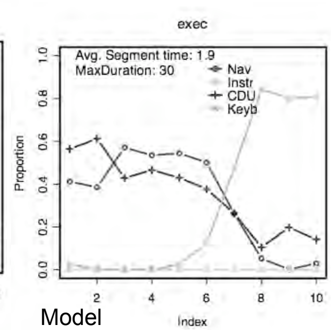


Carnegie Mellon

rijksuniversiteit groningen

Before EXEC





Carnegie Mellon

rijksuniversiteit groningen

Conclusions

- ACT-R/Lisa implementation
  - Looking up information on the display does not have to be modeled explicitly
- Looking up information when needed:
  - Fits the eye-movement data
  - Frees up the "working memory"
  - Tracks changes to display

Carnegie Mellon

rijksuniversiteit groningen

Statistical learning and anticipatory start-point selection

Matthew M. Walsh (mmw187@andrew.cmu.edu)
John Anderson (ja@cmu.edu)
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213 USA

Many theories seek to explain skill acquisition. One class of theories ascribes practice-related speedup to quantitative changes in task efficiency. These theories propose mechanisms like production compilation (Newell & Rosenbloom, 1981) and production strengthening (Anderson, 1982). A second class of theories attributes skill acquisition to qualitative changes in solution-method selection (Logan, 1998). While both classes of theories account for higher-level learning in complex tasks, researchers have paid less attention to changes in lower-level, perceptual-motor performance.

To explore this issue, we conducted an experiment using a modified version of Freidman's classic probability learning task (1964). In our experiment, participants placed a mouse cursor between two circles presented on a computer monitor. After they positioned the cursor, one circle turned green. Participants were required to move to and click in the green circle as quickly as possible. Within each experimental block, each circle was programmed to turn green at a fixed probability. Between blocks, these fixed probabilities changed. If participants are sensitive to the fixed probabilities, we reasoned that they would select the cursor starting point that enabled the most rapid movement to the anticipated target. This prediction was met. From selected start points, the relative proximity to each circle equaled the probability of that circle serving as the target.

To better understand participants' performance, we have developed competing ACT-R models. Our first model utilizes ACT-R's utility learning mechanism. After each trial, a reward determined by movement completion time propagates to the earlier productions associated with the selected cursor starting point. Our second model incorporates an additional declarative component in which the base-level activations of different target probabilities reflect trial history.

In sum, our behavioral results show that people anticipate events, and that they take preparatory actions to facilitate forthcoming movements. As seen in our modeling, this pattern of decisions is accounted for by a mixture of utility and base-level learning.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89*, 369–406.

Friedman, M. P., Burke, C. J., Cole, M., Keller, L., Millward, R. B., & Estes, W. K. (1964). Two-choice behavior under extended training with shifting probabilities of reinforcement. In R. C. Atkinson (Ed.), Studies in mathematical psychology (pp. 250-316). Stanford, CA: Stanford University Press.

Logan, G. D. (1988). Toward an instance theory of automatization. Psychological Review, *95*, 492–527.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), Cognitive skills and their acquisition (pp. 1–55). Hillsdale, NJ: Erlbaum.

Processes influencing visual search efficiency in conjunctive search
A rational analysis approach

Clayton T. Stanley and Michael D. Byrne
Department of Psychology, Rice University
Houston, TX

When deploying the eyes, how does the human visual system decide where to look next? Although extensive progress has been made towards an answer, current visual search theories are as much unified as they are unique, arguing solutions based heavily on neurodynamical research (Deco & Rolls, 2004), low-level spatial frequency filtering (Itti & Koch, 2000; Rao, Zelinsky, Hayhoe, & Ballard, 2002), or combinations of top-down and bottom-up attentional processes (Wolfe, 1994; Logan, 1996). Additionally, ACT-R currently employs only a minimal model to determine where visual attention should move, which doesn't handle bottom-up salience nor err on conjunctive searches. Although numerous and varied visual search theories do exist, each is capable of predicting various psychological phenomena present in visual search tasks. Therefore it should at least be helpful to outfit ACT-R with a more sophisticated visual search model, even if some attributes of that model are currently being challenged. Byrne (2006) developed a theory of visual salience computation based heavily on rational analysis of the visual system and provided the capability of implementing the model in ACT-R. However, the model has not yet been rigorously tested and verified.

We therefore tested Byrne's visual salience computation theory on a conjunctive visual search task similar to that studied by Shen, Reingold, and Pomplun (2003). Participants were able to search across both color and orientation of a rectangular object, and visual search efficiency was therefore dependent on the relative frequency of the two types of distractors. After calibrating, the ACT-R model successfully reproduced this distractor ratio effect, correlating strongly with participant data.

Calibrating the ACT-R model to fit participant's data required a stronger weighting across the color than orientation dimension for bottom-up activation, as well as giving top-down activation only for color. Implementing Byrne's visual salience theory in ACT-R and modeling this task thus allowed one to argue specific attentional processes and weightings involved that caused the observed reaction times. Utilizing this approach, the results found by Shen et al. (2003) can be directly compared and linked to the results found here, and relationships explained by a simple manipulation of these parameters.

Processes influencing visual search efficiency in conjunctive search

A rational analysis approach

Clayton T. Stanley & Michael D. Byrne
Department of Psychology
Rice University
Houston, TX 77005
clayton.stanley@rice.edu
byrne@rice.edu
http://chil.rice.edu

RICE

chil
THE COMPUTER-HUMAN INTERACTION LABORATORY AT RICE UNIVERSITY

5

Problem

- When deploying the eyes, how does the human visual system decide where to look next?
- Since its inception, the ACT-R visual system hasn't really addressed these issues
  - Currently doesn't handle bottom-up salience nor err on conjunctive searches
- Here is a first attempt to address such concerns

7

Salience Computation

Salience of feature i in the visual icon

bottom-up weightings

noise

$$L_i = \sum_{k=1}^{\#attr(i)} \log_2 \frac{1}{p_i(v_k)} \gamma_k + \log_2 \frac{1}{p_i(v_s)} + \sum w_j S_{ji} + \varepsilon$$

bottom-up salience

top-down spatial guidance

top-down value guidance

8

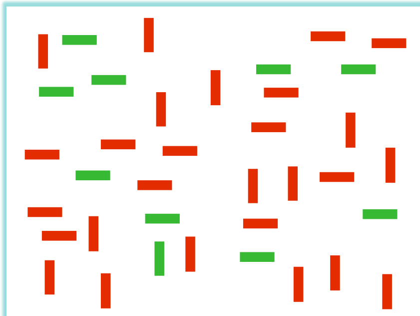
Methods

- 16 subjects
- 4 possible rectangular targets
  - (Red, Green) color
  - (Horizontal, Vertical) orientation
- 2 target conditions
  - (Present, Absent)
- 21 manipulated cells
  - Alters relative frequency of the two types of conjunctive distractors as well as number of disjoint distractors
- 3 repetitions of each configuration
- 504 shuffled trials per run
- Number of objects
  - 36 for target absent condition; 1 additional when present (target doesn't replace an object)

9

Methods

Target Present (GV)
Distract RV 15
Distract GH 9
Disjoint Distract RH 12



Search trial ex. a

11

Methods

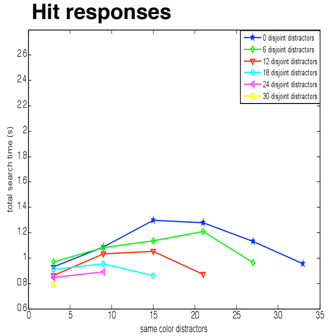
# same color distractors	# same orientation distractors					
	3	9	15	21	27	33
3	30 (d)	24	18	12 (e)	6	0
9	24 (c)	18	12 (a)	6	0	-
15	18 (b)	12	6	0	-	-
21	12	6	0	-	-	-
27	6	0	-	-	-	-
33	0	-	-	-	-	-

- Hyphens represent cells that were not tested. Numbers within a cell indicate the number of disjoint distractors placed on the screen

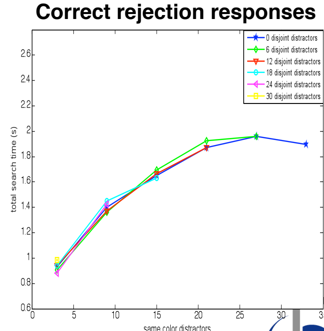
16

Results

Hit responses



Correct rejection responses



17

Discussion

- Average miss & false alarm rate was .06 & .013 respectively
- Asymmetrical curves for hit & correct rejection responses
  - Two components
    - First, the shape of the curves
      - Strong quadratic component of trend for hit responses
      - Almost asymptotic for correct rejections
    - Next, the 'squishiness' of the curves
      - Evidence for strong serial search component in hit responses
      - Evidence non-existent for correct rejections
- Interpretation
  - Difficult looking only at these data without a strong understanding of the underlying processes involved

18

3

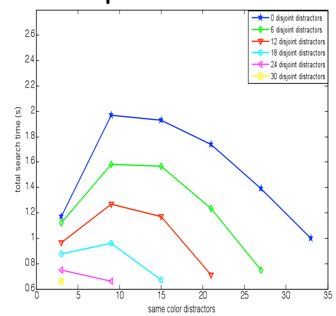
ACT-R Model

- Target rectangle encoded and placed in goal buffer
- +visual-location> requests cause model to find object with highest activation
  - Includes a slot only for target color
  - If object activation is greater than *salience-thresh*, chunk is placed in buffer; else nothing is returned
- If nothing is returned, model concludes that target is absent
  - Analogous to a memory retrieval failure
- If an object is returned and it is the target object, model concludes that target is present; else the model keeps looking
- If an object has been looked at, the object won't be looked at again

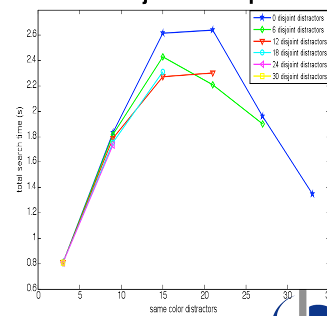
19

ACT-R Model Results

Hit responses



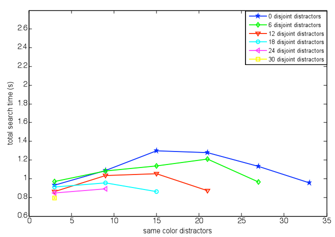
Correct rejection responses



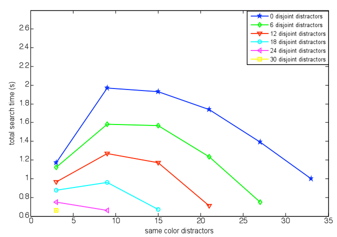
20

Model Fit: Hits

Participant data



ACT-R data

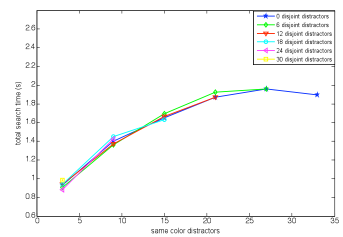


$R^2(19) = .74 \quad MAD = 245ms$

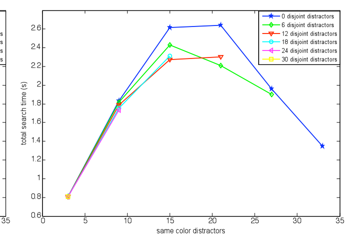
21

Model Fit: Correct Rejections

Participant data



ACT-R data

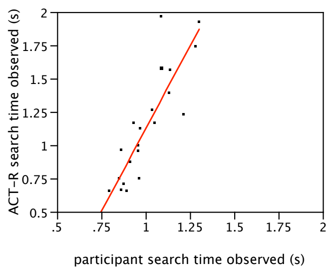


$R^2(19) = .70 \quad MAD = 367ms$

22

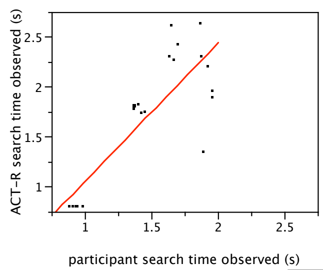
Model Fit: *r* Scatter

Hit responses



$R^2(19) = .74 \quad MAD = 245ms$

Correct rejection responses



$R^2(19) = .70 \quad MAD = 367ms$

23

Model Fit: Incorrect Responses

- Miss responses
  - Salience threshold calibrated to match miss rate
  - Therefore consistent miss rate for ACT-R (.07) and participant (.06) data
- False alarms
  - Small (but non-zero) for participants (.013)
  - Not modeled with ACT-R currently

24

Discussion: Asymmetrical Shape

- Subjects utilizing color primarily to guide their search
  - High bottom-up activation percentage for color relative to orientation
  - Top-down guidance only for color
  - High ratio of top-down/bottom-up activation
  - However, a bit of bottom-up activation for orientation still necessary to produce the strong quadratic present in the hit responses

25

Discussion: Asymmetrical 'Squishiness'

- Hit responses
  - Disjoint distractors are not often attended (if ever); however, their presence acts to 'shadow' conjunctive distractors relative to the target
  - Causes more accurate target pinpointing when a high number of disjoint distractors are displayed
  - Works alongside serial search effects to separate level curves

26

Discussion: Asymmetrical 'Squishiness'

- Correct rejections
  - Subjects concluding 'target absent' by an analogous memory retrieval failure for the vision system
  - Disjoint distractors again not often attended; however, their presence acts to increase information content for conjunctive distractors
  - Assuming a constant threshold, may cause a higher proportion of conjunctive distractors searched before concluding 'target absent'
  - Works against serial search effects to overlap level curves
    - Overlapping may also be influenced by a strong tendency to search for color

27

Discussion: Future Predictions

- Modified experiment: remove disjoint distractors

- Predictions using previous hypotheses
  - Hit responses
    - Less efficient search overall
    - Level curves closer together (i.e., more overlap)
  - Correct rejections
    - Curious about the interaction between salience threshold and task
    - If threshold unaltered, search time should decrease (more prominently where larger numbers of disjoint distractors resided)

28

Closing Remarks

- Strengths of model
  - Good correlations with participant data
  - Produces asymmetrical results for hit/cr conditions present in data
  - Interpretation of parameters are enlightening and seem plausible for the task

- Weaknesses
  - Search times still a bit long even after decreasing 'visual-attention-latency' to 25ms
  - Areas where longer search times exist in ACT-R model are not exchanged with more accurate responses (i.e., miss rate higher than participant data in these areas)

29

Closing Remarks

- Model predictions
  - Although disjoint distractors are not highly salient, their presence may actually improve search efficiency for the task by causing more accurate target pinpointing when the target is present
  - When the target is absent, disjoint distractors increase the information content of conjunctive distractors, affecting the average time elapsed before terminating the search
  - Next experiment aimed to challenge these predictions

- Code for the salience computations which works with the new vision module is now available @ [insert URL]

30

SYMPOSIUM

Large-Scale Computing Resources and ACT-R Modeling

Organizer: Kevin Gluck
Presenters: Sue Kase (skase@ist.psu.edu)[1]
Glenn Gunzelmann (glenn.gunzelmann@mesa.afmc.af.mil)[2]
Kevin Gluck (kevin.gluck@mesa.afmc.af.mil)[2]
Wayne Gray (grayw@rpi.edu)[3]
Brad Best (bjbest@adcogsys.com)[4]

[1]Penn State; [2]Air Force Research Laboratory; [3]Rensselaer Polytechnic Institute; [4]Adaptive Cognitive Systems

HPC + PGA + ACT-R = perfect fits (lacking validation?)

Sue E. Kase, Frank E. Ritter, and Michael Schoelles

Fitting a cognitive model to human data is a stochastic global optimization problem. The fitness function of a parallel genetic algorithm (PGA) executed a lisp image file of ACT-R 6.0 and a model of a serial subtraction task to 'optimize' the model to multiple levels of human data (overall average, subject groups, individual subjects). Running on a high performance cluster at the National Center for Supercomputing Applications, the PGA searched a 3-D bounded parameter space finding nearly perfect fits in all cases. Each optimization found, not one, but several different nearly perfect fitting ACT-R parameters sets scattered across the ACT-R parameter space. These results raised concerns that this powerful optimization technique may work a little too well. For example, the highest performing subject reported using an entirely different problem-solving strategy than the model, nevertheless, the PGA expediently found six different exceptionally-fitting ACT-R parameters sets to match the human data. One is left to wonder which is worse: using 'manual' optimization and not knowing what fits you missed; or finding all the best fits possible but not knowing which are valid.

Enabling Scientific Progress Through the Use of Large Scale Computing Resources

Glenn Gunzelmann

An important issue to consider in adopting a new methodology is how it will facilitate achieving the research goals. By enabling faster parameter space explorations that are both larger and more detailed, large scale computing resources have the ability to open research opportunities that were previously intractable. For example, we have been able to conduct a detailed analysis of individual differences in sustained attention performance across 88 hrs of total sleep deprivation, which involved variations in both the model's implementation and parameters. Hand tuning multiple model variants to fit each of 1716 experimental sessions of data, or running a sufficiently large parameter space on a small number of machines, would have stretched the timeline to many months. Using large scale computing resources, the evaluation was completed within a couple of weeks, including running the space and fitting the data. The larger goal of the research is to find parameter values to produce fits across multiple tasks for individuals on a session-by-session basis, as they experience several days of sleep deprivation. This includes simultaneously fitting data across tasks, as well as expanding the set of parameters that may be manipulated as the theory is extended to other cognitive capacities. Without large scale computing resources, this goal would be unapproachable.

The MindModeling Meta-Computing Infrastructure

Kevin Gluck, Jack Harris

We will describe our MindModeling system (http://www.mindmodeling.org), which integrates local, volunteer, and high performance computing resources into a meta-computing software infrastructure for computationally demanding cognitive modeling research. The presentation will include a description of some of the challenges and interests that motivated creation of the system, its current implementation, and its ongoing development. The take-away message is that access to large-scale computational resources has been useful for us and may be useful for others. To as great an extent as possible, we would like to make this enabling infrastructure available to our colleagues and collaborators in the cognitive science community, and especially those who are doing ACT-R modeling. We look forward to discussing whether and how the MindModeling system can be useful for you.

MindModeling@Rensselaer

Chris Sims, Wayne Gray

We have used the MindModeling system to perform a parameter search for the reinforcement learning model reported in Gray, Sims, Fu, and Schoelles (2006). The published parameters were selected the usual way: intuition, prior experience, and a limited search of the parameter space. For the new search we tried to fit the model on 400,000 parameter combinations. I will talk briefly about our experience doing this with the MindModeling software, about what we learned about parameter fitting, and about the issues that having 400,000 fits to your data raises.

Adaptive Model Exploration Algorithms

Brad Best

The combinatorics and inherent computational load associated with exploration and understanding of the details and limits of performance spaces in computational cognitive models, such as those implemented in ACT-R, can quickly overwhelm any computing resource, no matter how large. Thus, there is a significant challenge in both selecting an appropriate granularity for exploring parameter spaces, and for managing the combinatorial explosion that results from considering these larger cognitive modeling parameter spaces. The path to more efficient development and validation of cognitive models that are orders of magnitude larger than those developed today must include approaches that reduce sampling load through intelligent search of parameter spaces. At Adaptive Cognitive Systems we are working with AFRL on the development, comparison, and extension of adaptive model exploration algorithms. In particular, we are exploring methods for improving the efficiency of HPC cognitive modeling work through three main paths: 1) the application of Adaptive Mesh Refinement (AMR) techniques as a method for broadly sampling a wide range of model behavior while minimizing resource usage, 2) the generation of mathematical models describing model behavior across parameter variations, including exploration of variations in sampling and in the efficacy of different mathematical representations as a means of understanding model and architecture dynamics (e.g., regression models, cubic splines), and 3) the packaging of interacting parameters into higher-order constructs to simplify both the search of parameter space and the explanations of model behavior in that space as well as to reduce the overall size of the search space. Taken as a whole, these methods together support the carefully focused application of available computational resources.

Disclaimer and Acknowledgments: The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The research conducted at the Air Force Research Laboratory and at Adaptive Cognitive Systems was sponsored by AFRL's Warfighter Readiness Research Division and by grant 07HE01COR from the Air Force Office of Scientific Research.

MindModeling.org

LANGUAGE + INTEGRATION

The development of Lisp bindings for the D-bus interprocess communication system

Marc Destefano (destem@rpi.edu)
Department of Cognitive Science
Rensselaer Polytechnic Institute
Troy, NY 12180 USA

This presentation will discuss the development of Lisp bindings for the D-Bus Interprocess Communication System, as well as the use of D-Bus in interfacing ACT-R with external simulations.

Given the real-time demands of certain task environments, it is important to have an extremely fast way to have ACT-R and external simulations communicate with each other – simulations need to send data representing the current state of the world to ACT-R's visicon (via the device module), and ACT-R needs to send data representing the actions of its manual module to the simulation's event manager. The idea of sending "raw" data over UDP or TCP/IP is undesirable because the platform-agnostic nature of both ACT-R and many simulation engines (e.g., Pygame, Delta3D) demand a standard method of serializing and deserializing data transmitted over a socket, and it would then be necessary to write a custom encoder and decoder for every language incorporated within the system. Our solution is D-Bus, a lightweight and super-fast message bus system specifically designed for inter-process communication (IPC). D-Bus is used extensively within the Linux operating system, and thus has been widely tested for many years, proving its robustness and reliability. D-Bus, following a template provided by the application developer, encodes and decodes data into XML behind the scenes, and this data can be transmitted to any application that attaches itself to the communication bus, whether it is local or over a network.

Unfortunately, despite D-Bus having bindings for almost every programming language in use today, it does not specifically have bindings for Lisp. Efforts are now almost complete to write these necessary bindings, allowing Lisp to join the large collection of programming languages that D-Bus supports. This will be a highly rewarding effort, for once the bindings are complete, ACT-R will be able to communicate rapidly with any application to which the researcher has source code (in essentially any language), and in some cases, when the research only has access to an application's scripting language. By interfacing with ACT-R's device module, this opens a whole new world of external software devices with which cognitive models can interact, including 3D virtual environments.

Modeling Long-Distance Dependencies in Double R Language

Jerry T. Ball (jerry.ball@mesa.afmc.af.mil)
Air Force Research Laboratory

Long-distance dependencies are the *sine qua non* of modern linguistic theorizing (cf. Chomsky, 1981, Culicover & Jackendoff, 2005; O'Grady, 2005). An empirically valid computational model of language comprehension must be capable of modeling long-distance dependencies as a prerequisite to determination of meaning. Common forms of long-distance dependency include:

- Binding of Anaphors, Pronouns and R(eferring)-Expressions (e.g. "John$_i$ kicked himself$_i$", "John$_i$ kicked him$_j$" ($_i$ not = $_j$), "*John$_i$ kicked John$_i$")
- Subject and object control constructions (e.g. "He$_i$ promised me t$_i$ to go", "He persuaded me$_i$ t$_i$ to go")
- 'Raising' constructions (e.g. "He$_i$ seems t$_i$ to be happy", "The ball$_i$ was kicked t$_i$")
- Relative clauses (e.g. "The man$_i$ who$_i$ t$_i$ kicked the ball")
- Wh-questions (e.g. "What$_i$ did he kick t$_i$")

In the presentation, I will discuss the modeling of long-distance dependencies within the context of an ACT-R model of language comprehension—Double R Model (Referential and Relational Model)—which is intended to be at once functional, empirically valid and cognitively plausible (Ball, Heiberg & Silber, 2007).

Besides striving for empirical validity from the perspective of linguistic data, the model adheres to well-established psycholinguistic constraints on language processing (cf. Crocker, 1999; Gibson & Pearlmuttter, 2000; Lewis, 2000), including:

- A "mildly" *deterministic, serial* processing mechanism (selection and integration) operating over a *parallel, probabilistic* spreading-activation substrate (activation)
- *Interactive* and *non-autonomous* processing (no distinctly syntactic representations exist)
- *Incremental* processing with *immediate determination of meaning* – word by word
- *No algorithmic backtracking or lookahead* – a mechanism of context accommodation is used instead (a mechanism of reanalysis when accommodation fails is yet to be implemented)
- *Forward chaining* only (no backward chaining of productions)
- *Declarative and explicit linguistic representations* generated via *implicit execution of productions*
- Model must *operate in real-time* on Marr's algorithmic level (serial and parallel processing are relevant) as implemented in the ACT-R cognitive architecture

I will discuss the processing of linguistic expressions containing long-distance dependencies within the context of the above psycholinguistic constraints.

References

Ball, J., Heiberg, A. & Silber, R. (2007). Toward a Large-Scale Model of Language Comprehension in ACT-R 6. *Proceedings of the 8th International Conference on Cognitive Modeling*, 173-179. Edited by R. Lewis, T. Polk & J. Laird. NY: Psychology Press.
Chomsky, N. (1981). *Lectures on Government and Binding*. Dordrecht: Foris.
Crocker, M. (1999). Mechanisms for Sentence Processing. In Garrod, S. & Pickering, M. (eds), *Language Processing,* London: Psychology Press.
Culicover, P. & Jackendoff, R. (2005). *Simpler Syntax*. NY: Oxford.
Gibson, E. & Pearlmutter, N. (2000). Distinguishing Serial and Parallel Parsing. *Journal of Psycholinguistic Research*, 29, 231-240.
Lewis, R. (2000). Falsifying serial and parallel parsing models: Empirical conundrums and an overlooked paradigm. *Journal of Psycholinguistic Research*, 29, 241-248.
O'Grady, W. (2005). *Syntactic Carpentry, An Emergentist Approach to Syntax*. Mahway, NJ: LEA

Adapting the use of attributes to the task environment in joint action: results and a model

Markus Guhe
University of Edinburgh
Linguistics and English Language & HCRC
40 George Square
Edinburgh, EH8 9LL
m.guhe@ed.ac.uk

Ellen Gurman Bard
University of Edinburgh
Linguistics and English Language & HCRC
40 George Square
Edinburgh, EH8 9LL
ellen@ling.ed.ac.uk

Abstract

Speakers use referring expressions to identify an object in the environment. To generate a referring expression, features of the intended referent have to be selected that distinguish the object from the other potential referents. Current accounts of referring expressions consider a number of factors that influence the choice of features but ignore the influences of the task environment. In particular, they do not address how these influences change the generation of referring expressions over an extended period of time. We present results of how colour terms are used to describe landmarks in a task oriented dialogue (a route communication task) and describe a computational cognitive model of the observed adaptations over time.

1 Introduction

Much attention in recent computational as well as psychological research on language has been given to the linguistic problem of the use and generation of referring expressions. Referring expressions are linguistic expressions that identify either a referent entity in the real world or a discourse entity in the form of an antecedent. Referring expressions serve the purpose of distinguishing the target or referent from the set of other possible referents in the given context, called the distractor set. For example, in the set of objects in Figure 1, *the black cup* and *the small, black cup* would both succeed in distinguishing the cup at the lower left (the referent) from the other two objects (the distractor set).

A speaker wanting to pick out that small, black, cup at the lower left of the array could use any of the attributes in the expressions just given. Computational approaches to generating referring expressions often produce expressions that, if possible, uniquely and minimally select the target object. But such algorithms are computationally costly and may not be helpful in modelling human behaviour: People (1) produce non-minimal expressions, which contain redundant information (e.g., Pechmann 1989) and (2) interpret such expressions more easily (e.g., Paraboni, van Deemter and Masthoff 2007).

Figure 1: A simple domain of reference: for each object, the other are distractors

A prominent account of how human-like, non-minimal referring expression can be generated is the algorithm by Dale and Reiter (1995), which by now has many extensions (see van der Sluis (2005) for a recent overview). This algorithm incrementally tests whether using an attribute in a referring expression will rule out distractor objects. The attributes are tested according to a preference list that is fixed beforehand. For the domain used in Figure 1, for example, this preference list could be <type, colour, size>. Identifying the object to the right would then produce the non-minimal expression *large, white cup* by first adding the type attribute (which has a special status and is always added), then by adding *white* (because it removes the object in the lower left from the distractor set), finally by adding *large* (because it removes the object in the top left). Non-minimal expressions arise simply because a selected attribute is never de-selected, even if a subsequently selected attribute makes it redundant.

While these approaches deal with which of the available possibilities to describe the target object is chosen, they do not account for the adaptations that a speaker makes over time to the demands of the current task environment. The computational as well as the psycholinguistic paradigms typically lack history: On each trial a participant (or algorithm) is presented with a picture like Figure 1 and instructed to produce a suitably distinguishing expression. The trial terminates without feedback and is followed by others, presenting different objects and distinguishing features. How the fourth target is distinguished from its distractors might actually owe something to the participant's experience with the first three, and our work attempts to discover and model such effects of experience.

We examine referring expressions in an unrestricted, task-oriented dialogue in which the interlocutors get natural feedback on failures of reference and refer to many different objects. We use a variant of the HCRC Map Task (Anderson et al. 1991) in which a player who can see the route on a schematic map describes it to a fellow player who must reproduce it. Each map is populated with cartoon landmarks, distinguished by several different features. We have shown that the use of features changes across first mentions as players pursue their task (Guhe and Bard 2008). In the present paper we ask how and why the changes take place. Colour is a perceptually salient property, usually one of the first tested in the incremental Dale and Reiter type algorithms. In our experiment, however, we set unreliability against salience: Colour is an unreliable distinguisher. In contrast, each map allows for use of a reliable attribute, too, (shape, number, kind or pattern). Thus, our participants need to use the adaptive attributes but waste time and can cause misunderstandings using the unreliable one.

In this paper, we report how the use of colour terms changes over the course of the experiment and present a simple computational cognitive model of this change. More precisely, we describe how the utility of the colour feature influences the Instruction Giver's choice of whether to use colour in introductory referring expressions. The model offers an explanation of this change in terms of Anderson's rational analysis (Anderson 1990; Anderson and Schooler 1991). Rational analysis is the core mechanism in ACT-R's utility-based production selection (Anderson 2007) and is a variant of utility learning mechanisms found in reinforcement learning or the delta rule (Sutton and Barto 1998). In brief, rational analysis says that human memory reflects the frequency of events in the environment, making more frequent experiences easier to retrieve and corresponding behaviours more likely to be used. By using rational analysis our model goes beyond existing accounts of use and generation of referring expressions in that it reveals the environmental influences on these processes.

2 Comparison to existing research

The problem of whether the use of features changes with the demands of the task environment has scarcely been addressed in the literature. Although Brennan and Clark's (1996) conceptual pacts address changes in referring expressions, these changes are about how speakers refer to objects after they have been introduced. However, our questions here address the overall use of features in referring expressions over the course of many interactions. To exclude effects of conceptual pacts we are only analysing the use of introductory (first) mentions of landmarks.

Garrod and Doherty (1994) describe how a community of speakers establishes a sublanguage in referring to entities. We are concerned with the internal structure of the referring expressions themselves and propose a utility-based explanation instead of one based on precedence and salience.

There is some evidence that extra-linguistic factors play a role in generating referring expressions. For example, Arnold and Griffin (2007) show that the presence of a second character influences the choice of whether to use a pronoun or the character's name for references following the introductory mention. This is true even if the characters differ in gender, so that the name does not disambiguate any more than the pronoun. Arnold and Griffin argue that the reasons for this behaviour lie in the speakers' cognitive load when they generate the referring expression.

This is part of another strand of findings in which the cooperative view on dialogue (e.g. Clark 1996) is changed towards a speaker-oriented view (e.g. Bard et al. 2000). In this view, the speaker makes the general assumption that what he/she knows is shared knowledge. Only if problems arise in the dialogue, e.g. by explicit feedback from the listener, might the speaker adapt to the listener's needs. In fact, even if overspecified referring expressions (Dale and Reiter 1995; Paraboni, van Deemter and Masthoff 2007; Pechmann 1989) help the listener to identify the target object, the speaker also profits in terms of a generation process of greatly reduced complexity. Since both – speaker and listener – benefit from using such referring expressions, the communicative strategy cannot be attributed uniquely to concerns for the listener's needs. In our task, however, the colour feature is counterproductive in the majority of cases, because it does not match between the two maps. So the speaker's assumption about the usefulness of the salient feature colour are mistaken.

Another related line of research is the use of machine learning techniques to extract the way attributes are selected for modified versions of the Dale and Reiter algorithm (Jordan and Walker 2005). Although these algorithms already incorporate psychological findings, e.g., conceptual pacts, they only provide global adaptations to properties of linguistic corpora and do not account for changes over time and for adaptations to the properties of the task environment.

3 Experiment

3.1 Task

The experiment is a modified Map Task (Anderson et al. 1991). The Map Task is an unscripted route-communication task in which an Instruction Giver and an Instruction Follower each have a map of the same fictional location. The Giver's map contains a route that is missing on the Follower's map. The dyad's goal is to recreate the Giver's route on the Follower's map.

The dialogue partners use the landmarks on the maps to navigate from START (shared) to FINISH (only on the Instruction Giver's map).

3.2 Materials, procedure, data collection

Materials. Some landmarks differ between the two maps. In our experiment they can differ by:

1. Being absent on one of the maps or present on both;
2. Mismatching in a feature between the two maps (most notably colour);
3. Being affected by 'ink damage' that obscures the colour of some landmarks on the Instruction Follower's map.

There are four attributes which also distinguish landmarks. Each serves for two different kinds of landmarks:

1. *Number* (bugs, trees),
2. *Pattern* (fish, cars),
3. *Kind* (birds, houses/buildings),
4. *Shape* (aliens, traffic signs).

Three crossed independent variables determine the nature of Giver–Follower map pairs:

1. *Homogeneity*: whether the landmarks on a map are of just one kind (single) or of different kinds (mixed).
2. *Orderliness*: whether the ink blot on the Instruction Follower's map obscures a contiguous stretch of the route (orderly) or a non-contiguous stretch (disorderly). The number of obscured landmarks is constant.
3. *Animacy*: whether the landmarks on a map are animate or inanimate (thus, on the mixed maps there are only landmarks from the 4 inanimate or the 4 animate kinds of landmarks).

The maps in Figure 2 are a pair of Giver and Follower maps for the disorderly, mixed tree condition. Thus, the maps contain mainly trees but also other inanimate objects (mixed), and the Follower's map shows multiple, non-contiguous ink blots (disorderly).

Procedure. Participants are told that the maps are 'of the same location but drawn by different explorers'. They thus know that the maps can differ but not where or how. They are instructed to recreate the route on the Follower's map as accurately as possible.

Each dyad did 2 simple training maps and then completed a set of 8 maps, one for each kind of landmark. The maps were counterbalanced with respect to the experimental conditions. After the fourth map, the role of Instruction Giver and Instruction Follower were exchanged.

To reduce the variability of words and concepts used in the unrestricted dialogues, each participant was prompted textually to provide standard type names for a few landmarks that would occur on the following map.

Setup and data collection. Participants sat in front of individual computers, facing each other, but separated by a visual barrier.

This research is part of a larger multimodal project. The communication was recorded using 5 camcorders. The Giver was eye tracked using a remote eye tracker. Speech was recorded using a

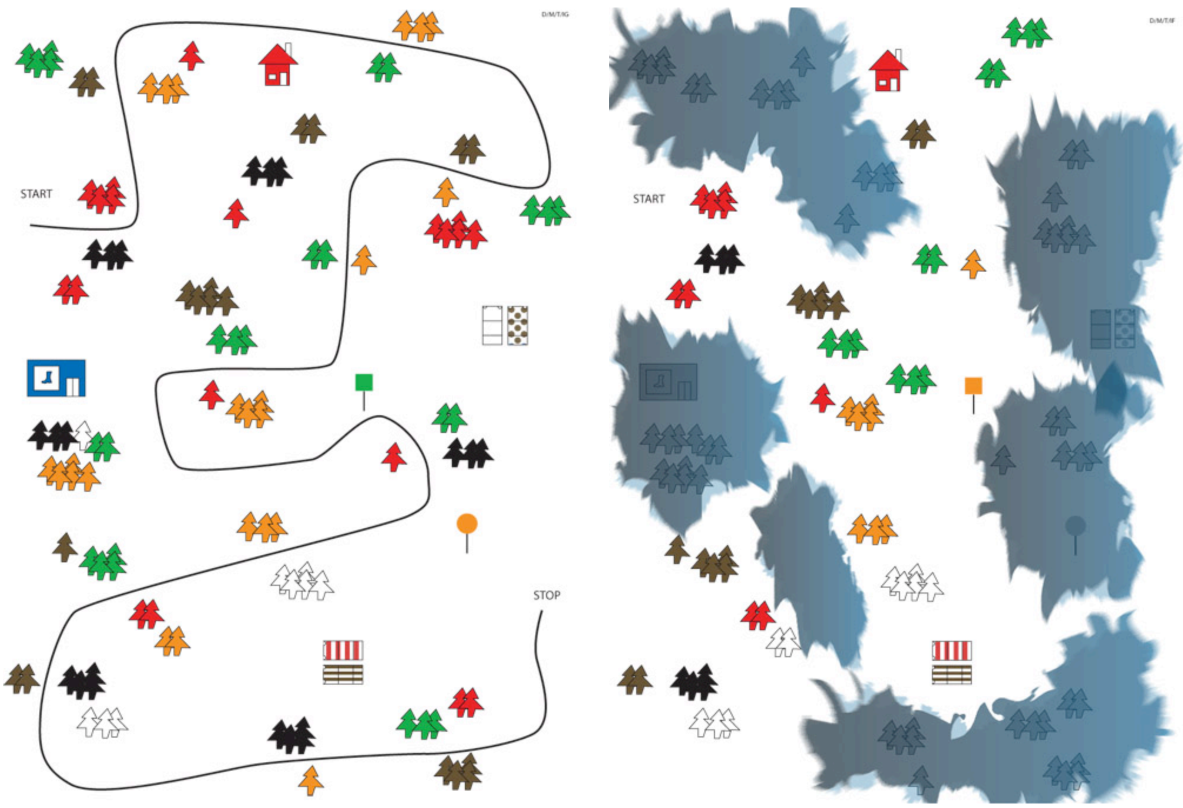


Figure 2: A pair of example maps; Instruction Giver left, Instruction Follower right

Marantz PMD670 recorder whereby Giver and Follower were recorded on two separate channels using two AKG C420 headset microphones. The speech was transcribed manually. The routes drawn by the Follower were recorded by the computer.

As the participants were in the same room, they could hear each other's speech. They could also see each other in the left half of their monitor, which showed the dialogue partner's upper torso video stream. The right half of the monitor showed the map.

Participants. In exchange for course credit, 64 undergraduates of the University of Memphis participated in pairs. In 4 dyads the participants knew each other previously.

3.3 Analysis and results

The recorded dialogues were coded for referring expressions. We present results for the first mentions of landmarks by the Instruction Giver. Introductory mentions should be both maximally independent of one another (as repeated mentions reflect precedence in naming a given object) and maximally detailed (as reductions in form characterise anaphora). Mentions of colour in landmark introductions were calculated as a proportion of opportunities

1. Over the course of single dialogues (by quartiles),
2. Across successive maps (1–8) and
3. Between those where the Instruction Giver lacked or already had experience as Instruction Follower.

The changes in the ratio of colour term use is depicted in Figure 3.

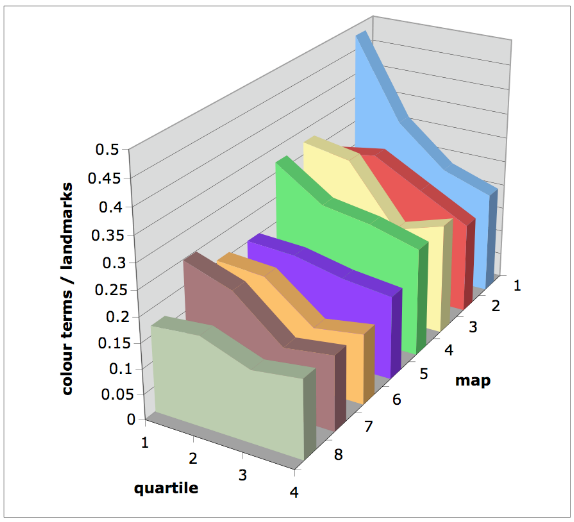


Figure 3: Change of the use of colour terms over quartiles of the eight maps

The use of colour terms significantly decreased over an average dialogue (effect of quartile within experience (2) x map encountered as Instruction Giver (4) x quartile (4) ANOVA on the arcsine transformed proportion of colour terms: $F_1(2, 54.8) = 15.57$, $p < 0.001$). Although there was no significant reduction across dialogues with the same Instruction Giver, the Givers used significantly fewer colour terms when they had served earlier as Follower (0.267 colour terms on average in the first four maps vs. 0.175 in the second four). This is a significant effect of experience ($F_1(1, 28) = 7.90$, $p < 0.01$).

Note that the orderliness of the ink blots on the Instruction Follower's maps did not have a significant effect. In contrast to colour, distinguishing features (number, kind, shape, pattern) are significantly more common in the maps where they are critical (used in more than 80%) and significantly increase within a dialogue. Thus, the decrease and low overall use of colour terms is not due to a general decrease in use of feature terms. There is also no effect of prior experience as Giver for useful features. The detailed results are presented in Guhe and Bard (2008).

3.4 Discussion

The participants adapted their use of colour to its low utility in the given task environment. The adaptation was distributed between speaker and listener. The use of colour terms does not fall significantly over the 4 dialogues a participant has the role of Instruction Giver, but there is a significant drop when the participants exchange roles: experience trying to match colour terms to grey-scale objects as Instruction Follower discourages to mention colour as Instruction Giver. Any listener-centric effect is outweighed or fuelled by a speaker-centric appreciation of utility.

4 Utility and task environment

4.1 Utility and selection probability

This is not the place to delve into the depths of the ACT-R theory, see Anderson (2007) for the most recent account. For the model described below it is only relevant that in ACT-R procedural knowledge (such as to decide whether to use colour or not) is encoded as production rules, or productions for short. A production is basically an if–then rule: *if* a certain set of conditions are given *then* execute a specified action.

In ACT-R, each production has a utility value. The utility is an estimate of how likely the use of the production results in achieving the current goal (here: successfully describing the landmark to the interlocutor).

Productions' utilities are important in the cases in which more than one production is applicable for a given set of conditions. Then, the utilities serve to compute the probabilities with which a production is selected. This selection probability is computed as:

$$P_i = \frac{e^{U_i/s}}{\sum_j e^{Uj/s}}$$

with:
P_i: selection probability for production i
U_i: utility of production i
s: noise in the utilities (defaults: $s = 1$)
j: set of all applicable productions (including i)

Utility values are learnt over time. After a production has been used, its utility is updated depending on whether it was successful according to the following equation:

$$U_i(n) = U_i(n-1) + \alpha[R_i(n) - U_i(n-1)]$$

with:
U_i: utility of production i
n: number of applications of the production
α: learning rate
R: reward

If the production is applied successfully, the utility is updated with a positive reward, if it is unsuccessful, it receives a negative reward.

Anderson (2007, p 161) points out that this is basically the Rescorla-Wagner learning rule (Rescorla and Wagner 1972) or the delta rule by Widrow and Hoff (1960). So there is nothing special 'ACT-R-ish' about this rule; it is a general learning rule.

4.2 Structure of the task environment

In the maps about half of the landmarks on the Instruction Follower's map are obscured by ink blots, and, therefore, don't have colour. Additionally, some of the route critical landmarks mismatch in colour. Overall this means that using colour to describe a landmark is successful in only about 40% of cases. By comparison, using the distinguishing feature of a map is successful in about 92% of cases.

5 Model

5.1 Introduction

The following analyses compare the model's performance to the introduction of the first 33 landmarks of each map by the Instruction Giver. The 33rd landmark is still mentioned in 206 of the possible 256 cases (32 dyads with 8 maps each). The 34th landmark is introduced only 186 times.

There are three main patterns in the data. Firstly, map 1 behaves differently than the other maps in that the number of colour terms shows a pronounced drop from 0.6 to 0.25 (taken from the means of the first and last three values). Secondly, maps 2 to 4 each show a decrease of colour rate from 0.3 to 0.2. Thirdly, in maps 5 to 8 – after the role change – the colour rate drops in each map from 0.2 to 0.15. (This lower colour rate is the basis for the effect of role change.)

Thus, between maps the colour rate is going up again. Explanations may be that the longer-term utility of colour (learnt over a lifetime) or the textual prompting between dialogues exert some influence. The fact that the colour rate in maps 5 to 8 starts at the same rate as it ends in maps 2 to 4 may be due to the utility learning during the time as Instruction Follower. But a more detailed model is needed to explain this.

5.2 The model

The model is not a fully implemented ACT-R model, but just uses the two equations for updating production utility and probability of production selection introduced above. The model contains two competing 'productions' one for using colour, one for not using colour. Because the Instruction Giver always has colour available to describe a landmark, the model assumes that both productions are applicable for each landmark. Thus, the model is similar to the ACT-R model for an experiment by Friedman et al. (1964), described by Anderson (2007, p. 165–169; in this experiment participants have to predict which one of two lights will be lit.) Using the other features would be modelled as analogous sets of productions.

For each decision, the model selects one of the productions according to their utilities and corresponding selection probabilities at that time. After the decision has been made, the usefulness of colour is determined according to the structure of the task environment (thus, using colour is successful in 40% of cases) and the utility of the selected production is updated accordingly. For a successful application the production receives a reward of $R = 14$; if it is unsuccessful it receives a reward of $R = 0$ (cf. Anderson 2007, p. 162).

The results reported in the remainder of this section were obtained by 500 runs of the model. However, just 32 runs – matching the number of dyads in the experiment – suffice to get significant results; more runs of the model just produce a smoother curve.

5.3 Map 1

For the first map the model starts with the following estimated utilities:

$U_{\text{colour}}(1) = 5.5$
$U_{\text{no-colour}}(1) = 5$

These values mean that the colour-production has a probability of being selected of 0.622, which is close enough to the mean of the first three values of 0.594. (Using $U_{\text{colour}}(1) = 5.4$ would give an initial probability of 0.599, but one can be too fussy.)

The final average utilities are:

$U_{\text{colour}}(33) = 4.6$
$U_{\text{no-colour}}(33) = 7.7$

Choosing these initial utilities gives an excellent fit to the data, see Figure 4. A regression using the model as predictor for the data shows a significant correlation ($\beta_1 = 0.90$, $p < 0.001$) that accounts for 72% of the variance ($R^2 = 72\%$, $F(1, 31) = 79.5, p < 0.001$).

However, the initial values are not that important, and the model matches the data significantly for a wide range of start values, as long as $U_{\text{colour}}(1) > U_{\text{no-colour}}(1)$ and the values are not close to the extremes of 0 and 14. The same holds for the following simulations.

5.4 Maps 2 to 4

For maps 2 to 4 (see Figure 5) the initial utilities were set to:

$U_{\text{colour}}(1) = 5.5$
$U_{\text{no-colour}}(1) = 6.5$

resulting in final average utilities of:

$U_{\text{colour}}(33) = 4.5$
$U_{\text{no-colour}}(33) = 7.5$

The regression shows that the model accounts for 66% of the variance (R^2 = 66.3%, $F(1, 31) = 61.0$, $p < 0.001$) with $\beta_1 = 2.44$ ($p < 0.001$).

5.5 Maps 5 to 8

Finally, for maps 5 to 8 (see Figure 6) the initial utilities were set to:

$U_{colour}(1) = 3$
$U_{no\text{-}colour}(1) = 4$

resulting in the final average utilities

$U_{colour}(33) = 3.3$
$U_{no\text{-}colour}(33) = 7.7$

The model accounts for 52.7% of the variance (R^2 = 52.7%, $F(1, 31) = 34.6$, $p < 0.001$) with $\beta_1 = 0.84$ ($p < 0.001$).

6 Conclusions

There are two main conclusions from the research presented here. Firstly, the dialogue partners indeed adapt their naming behaviour to the task environment. More specifically, they adapt to the fact that colour is an unreliable distinguisher for the landmarks on the maps. (This is amplified by the fact that the participants do not make a substantial effort to identify the parts of the maps that are obscured by ink, which shows in the absence of an orderliness effect.)

Secondly, the simple computational cognitive model accounts for this change. In particular, the model shows that the change in behaviour is indeed an adaptation to the structure of the task environment, because the rate of the probabilities and the changes in the probabilities with which

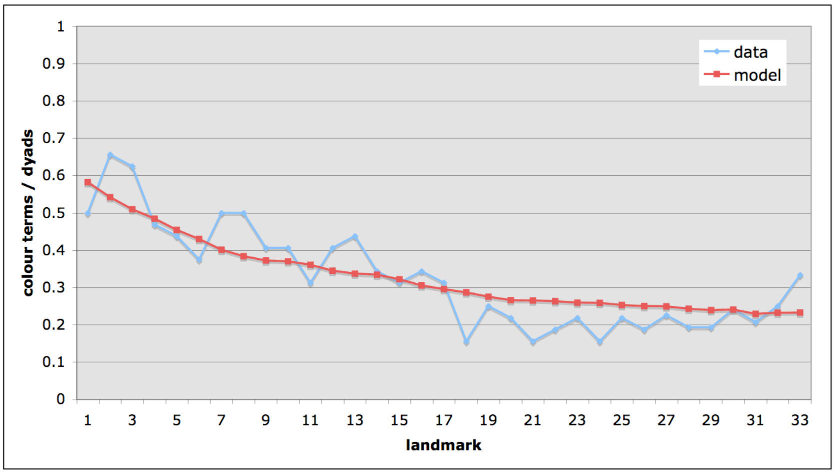


Figure 4: Comparison of data and model for the first 33 landmarks in map 1.

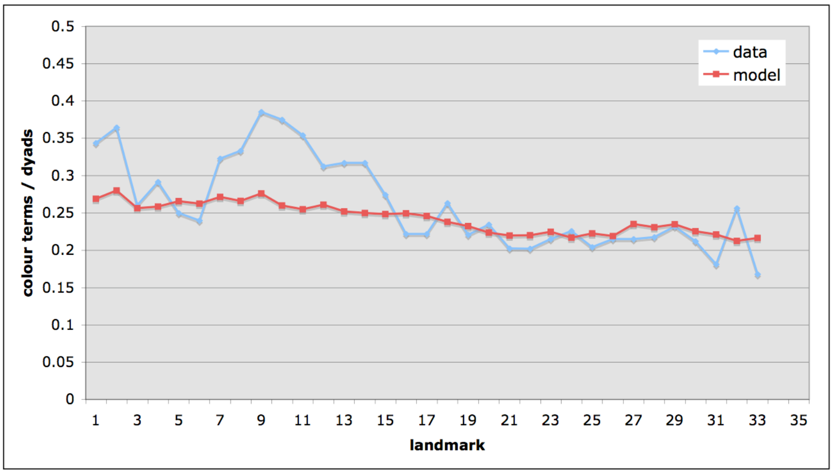


Figure 5: Data and model for maps 2 to 4.

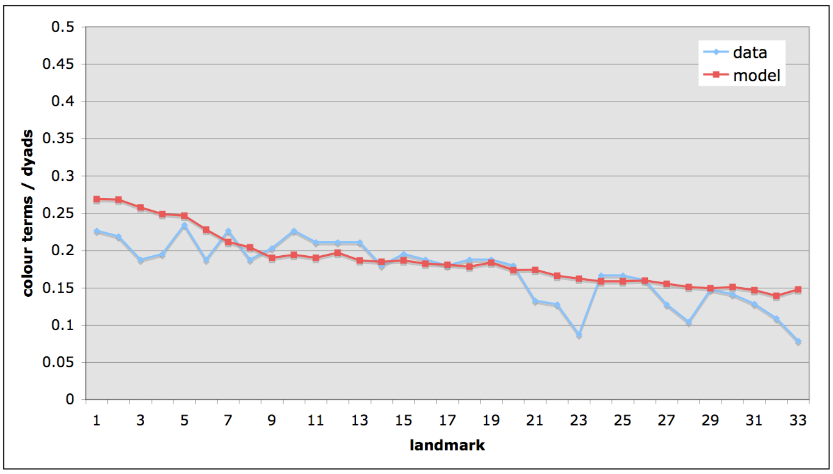


Figure 6: Data and model for maps 5 to 8.

colour is used as a descriptor is a direct result of the fact that colour can be successfully used for about 40% of the landmarks on the maps. Thus, rational analysis (the fact that memory reflects the probabilities encountered in the environment) explains the observed phenomenon.

Although – after the fact – it may not be too surprising that rational analysis explains the observed phenomenon, the result is more far-reaching, because the influences of the task environment on naming behaviour (the generation of referring expressions) has not yet been reported.

7 Future work

Our future research will address a number of direct follow-up issues. Firstly, the model will be extended to account for the changes in the mentions of the distinguishing features (number, pattern, kind, shape). Secondly, after a more detailed analysis of the data we will extend the model to account for individual adaptation patterns in the sense that the model can account for groups of dyads showing similar dialogue histories. For this, we will model the landmark introductions made by the Instruction Follower as well. This model serves as starting point for a comprehensive ACT-R model of how referring expressions (including repeated mentioned of landmarks) are generated in the given task.

Acknowledgements

This research was supported by grant NSF-IIS-0416128 to Max Louwerse, Art Graesser, Mark Steedman, and Ellen Gurman Bard. Thanks to Antje van Oosten and Jonathan Kilgour for help with the data coding and extraction and to Max Louwerse, Nick Benesh, Gwyneth Lewis and Megan Zirnstein for providing the transcriptions.

References

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H. S., and Weinert, R. (1991). The HCRC Map Task corpus. *Language and Speech*, 34(4), 351–366.

Anderson, J. R. (1990). *The Adaptive Character of Thought*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Anderson, J. R. and Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2(6), 396–408.

Arnold, J. E. and Griffin, Z. M. (2007). The effect of additional characters on choice of referring expression: Everyone counts. *Journal of Memory and Language*, 56(4), 521–536.

Bard, E. G., Anderson, A. H., Sotillo, C., Aylett, M. P., Doherty-Sneddon, G., and Newlands, A. (2000). Controlling the intelligibility of referring expressions in dialogue. *Journal of Memory and Language*, 42(1), 1–22.

Brennan, S. E. and Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(6), 1482–1493.

Clark, H. H. (1996). *Using Language*. Cambridge, MA: Cambridge University Press.

Dale, R. and Reiter, E. (1995). Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2), 233–263.

Friedman, M. P., Burke, C. J., Cole, M., Keller, L., Millward, R. B., and Estes, W. K. (1964). Two-choice behavior under extended training with shifting probabilities of reinforcement. In R. C. Atkinson (ed.), *Human Brain Function* (2nd ed.) San Diego, CA: Academic Press.

Garrod, S. and Doherty, G. (1994). Conversation, coordination and convention: An empirical investigation of how groups establish linguistic conventions. *Cognition*, 53, 181–215.

Guhe, M. and Bard, E. G. (2008). Adapting referring expressions to the task environment. In: *Proceedings of CogSci 2008*.

Jordan, P. W. and Walker, M. A. (2005). Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, 24:157–194.

Kronmüller, E. and Barr, D. J. (2007). Perspective-free pragmatics: Broken precedents and the recovery-from-preemption hypothesis. *Journal of Memory and Language*, 56(3), 436–455.

Paraboni, I., van Deemter, K., and Masthoff, J. (2007). Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, 33(2), 229–254.

Pechmann, T. (1989). Incremental speech production and referential overspecification. *Linguistics* 27, 98–110.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press.

van der Sluis, I. (2005). *Multimodal Reference: Studies in Automatic Generation of Multimodal Referring Expressions*. PhD thesis, Tilburg University, The Netherlands.

Structural priming in language production emerging from learning in an ACT-R model

David Reitter[1,2] , Frank Keller[2] and Johanna D. Moore[2]

[1] Department of Psychology, Carnegie Mellon University

[2] School of Informatics, University of Edinburgh

The psycholinguistic literature has identified two syntactic adaptation effects in language production: rapidly decaying short-term priming and long-lasting adaptation (see Ferreira & Bock 2006 for an overview). Evidence for both effects comes not only from experimental data but also from naturally occurring speech in dialogue corpora (e.g., Reitter et al. 2006). To explain the two types of adaptation, we present an incremental language production model in ACT-R that uses a wide-coverage, lexicalized syntactic theory (Combinatory Categorial Grammar (CCG), Steedman 2000) and models priming as a facilitation of lexical access.

Our production model explains structural priming on the basis of two well-established ACT-R mechanisms: base-level learning and spreading activation. Base-level learning applies to syntactic rules and explains long-term adaptation, while spreading activation from lexemes to syntactic types explains short-term priming through the retention of semantic information in a buffer after a sentence has been processed.

For each clause, the model initially selects a head lexeme with a lexical form and a syntactic frame. It then realizes each argument. Lexical forms spread activation to syntactic categorial types, e.g., the verb form "offered" will spread activation to "ditransitive (NP NP)" and "ditransitive (NP PP)". Here, spreading activation results in short-term priming, while base-level learning causes long-term adaptation. In the fully incremental (and most efficient) case, the processor can integrate each word with the preceding context, i.e. the full structure of the clause produced so far. This context is not retained in detail, but only described by a categorial type, reflecting the subcategorization frame of the current phrase. The current implementation of the model can generate sentences like "The policeman gave a flower to the girl", or "The cop gave the woman a rose". (Since a lexical form is retrieved before the syntactic node, the model predicts that syntactic priming cannot affect semantic choice.)

Simulations show that the model exhibits both priming and long-term adaptation in the form of increased relative accessibility, which leads to faster retrieval and a preference for the primed structure. We account for the following phenomena:

- Long-term adaptation as learning explains cumulativity of priming (prepositional-object with "give", effect of number of primes on repetition log-odds: $b = 0.095$, $p < 0.001$, long-term adaptation) (cf. Jaeger & Snider 2007).
- Rare constructions prime more. Simulations show that after the presentation of Zipf-distributed rules, ACT-R's base-level learning produces an activation pattern and an interaction with frequency equivalent to those found for short-term priming in corpora (effect of the rule's log-frequency on activation decay, $b = 0.0026$, $p < 0.001$).
- Lexical boost is due to spreading activation from retained lexemes to syntactic nodes in memory. (Not accounted for by Chang et al. 2006.)
- Short-livedness of lexical boost (Hartsuiker et al., in press): lexemes can only spread activation while still in a buffer.
- The lexical / semantic cause of short-term priming can potentially explain why short-term priming appears to be stronger in task-oriented dialogue, where semantic processing is needed, compared to spontaneous conversation (Reitter et al. 2006).

References

Chang, F., Dell, G., and Bock, K. (2006). Becoming syntactic. Psychological Review, 113(2):234-272.

Ferreira, V. and Bock, K. (2006). The functions of structural priming. Language and Cognitive Processes, 21(7-8).

Hartsuiker, R. J., Bernolet, S., Schoonbaert, S., Speybroeck, S., and Vanderelst, D. (in press). Syntactic priming persists while thelexical boost decays: Evidence from written and spoken dialogue. Journal of Memory and Language.

aeger, T.F. and Snider, N. 2007. Implicit learning and syntactic persistence: Surprisal and Cumulativity. Rochester Working Papers in the Language Sciences, Spring 2007, 3(1).

Reitter, D., Moore, J. D., and Keller, F. (2006). Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In Proc. 28th CogSci: 685-690, Vancouver, Canada. Steedman, M. (2000). The Syntactic Process. MIT Press.

HBA: Integrating Task Network Modeling and ACT-R

Walter Warwick
Rick Archer
Andrew Hamilton
Mike Matessa
Amy Santamaria
Alion Science & Technology

Ronald S. Chong
Cognitive Composites

Laurel Allender
Troy Kelley
Army Research Lab

Overview

- Motivation for the Human Behavior Architecture (HBA)
- The component technologies and approaches
- Current state of the software integration
  - The new integrated development environment
- Outstanding issues
- What we stand to gain

2

Motivation for the HBA project

- Human performance modeling comes in many shapes and sizes
  - Different questions, different needs, different theories, different resolutions, different tools, different M&S communities etc.
- Different needn't mean exclusive
- In fact, differences highlight opportunities for complementary approaches

3

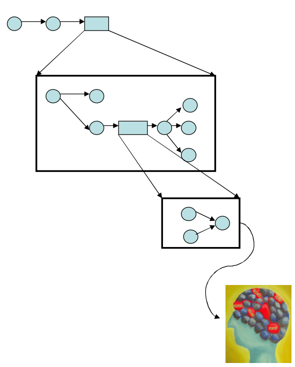
A Case in Point

- Task network models (e.g., Imprint, Micro Saint)
  - A proven and intuitive approach to understanding human performance through hierarchical decomposition of tasks
  - Widely regarded as lacking cognitive fidelity
- Hybrid productions systems as cognitive models (e.g., ACT-R)
  - Provide principled and detailed predictions of human cognitive performance
  - Seen by many as hard to build and understand

4

A Case in Point

- A good fit:
  - Take advantage of the flexible abstractions and the natural representation of a task network model to ease the development and inspection of a production system
  - Add cognitive fidelity to the task network model ("another layer of decomposition")
- Hence, the HBA



Component Technologies—Task Network Modeling Tools

- Intuitive, user friendly development environment
- Direct support for hierarchical task/function decomposition
  - But *not* limited to such representations!
  - Modeler fixes the level of abstraction, not the tool
- Standard practice reduces human performance to SME specifications and fixed decision types

6

Component Technologies—The ACT-R Cognitive Architecture

- A theory of cognition implemented as a hybrid production system
  - Symbolic knowledge representation
  - Underlying sub-symbolic calculi to represent statistical nature of cognition
- Principled representation of cognition
  - Not everything goes
- Ever growing body of validation studies across a wide range of tasks and phenomena
- It's rocket science

7

Common Aspects of the Two Approaches

- Intuitions common to both task network models and ACT-R (and other production systems)
  - Finite states (tasks or buffer contents)
  - Discrete transitions (between tasks and serially executing productions)
  - It's all human performance

8

HBA: A Unified C# Integration

- Human Behavior Architecture
  - Task Network: Micro Saint Sharp
  - Cognitive Architecture: ACT-R Sharp
- Unified Integration
  - reduces development time
  - removes need for communication software
  - instead forcing agreement of representation, allows multiple levels of abstraction in same environment

9

Constraints on the Integration

- ACT-R is an evolving architecture, time and money are always limited
- Chose to implement aspects of ACT-R according to the following criteria:
  - *Stability*: mechanisms that have been in the system since its inception and have stood the test of time;
  - *Demonstrated utility*: mechanisms that are regularly used by the community to account for the data; and
  - *Anticipated utility*: mechanisms that we anticipated would be needed for the kinds of tasks that might be modeled in HBA
- What's missing: Production learning, partial matching and associative learning

10

ACT-R Capabilities in ACT-R Sharp

- Manual Module
- Auditory Module
- Visual Module
  - Visual finsts, buffer stuffing
- Declarative Module
  - Chunk learning, base-level learning, spreading activation
- Procedural Module
  - Reinforcement-based learning of utility

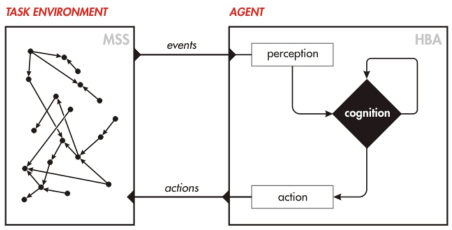
11

ACT-R Sharp Validation

- Procedural Module
  - Unit 1: count.lisp, addition.lisp, semantic.lisp, tutor-model-solution.lisp
- Visual & Motor Modules
  - Unit 2: demo2.lisp
- Visual & Auditory Modules
  - Unit 3: sperling.lisp
- Declarative Module
  - Unit 4: paired.lisp (chunk activation, base-level learning)

12

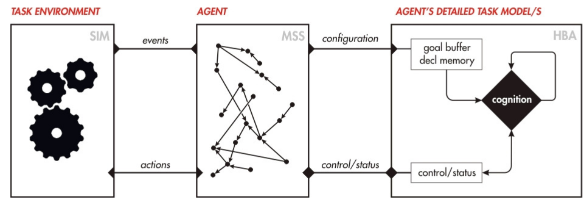
Modes of Interaction



- Task network as task environment; ACT-R Sharp as agent

13

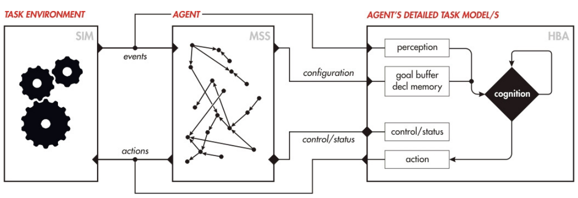
Modes of Interaction



- Task network as agent; ACT-R Sharp as a non-interactive model of an agent's task/s

14

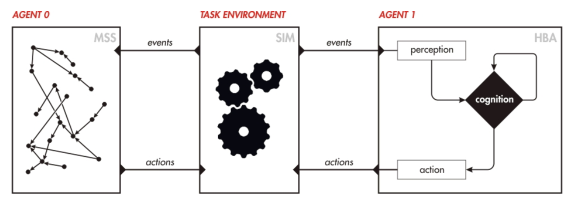
Modes of Interaction



- Task network as agent; ACT-R Sharp as an interactive model of an agent's task/s

15

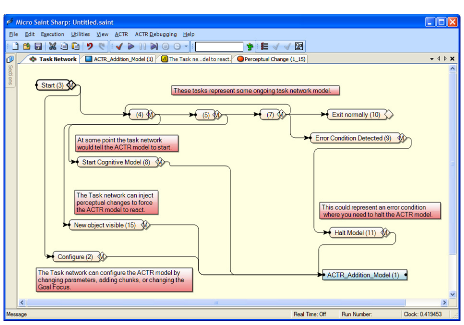
Modes of Interaction



- Task network and ACT-R Sharp as independent, interacting agents

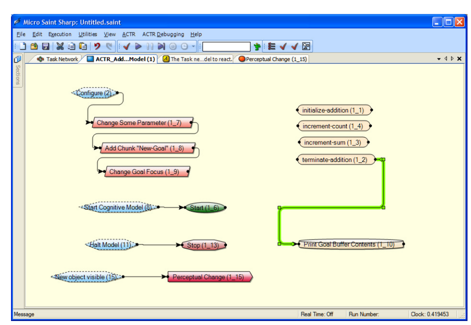
16

The Integrated Development Environment



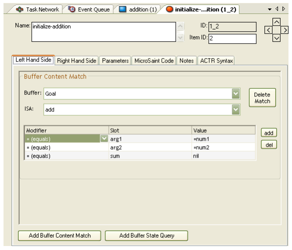
17

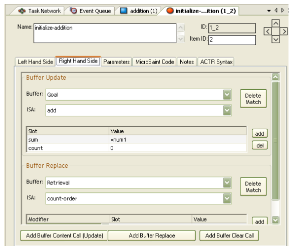
The Integrated Development Environment



18

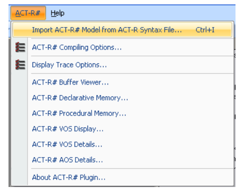
Building the Cognitive Network

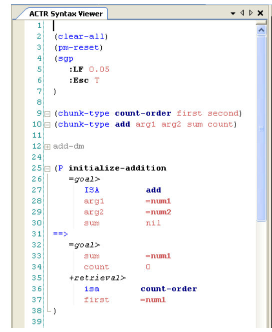




19

Building the Cognitive Network





20

Outstanding Issues

- Running code, regression testing in-house usability assessments…all good
- Still, a “culture gap” to bridge between TN and cognitive modelers
- How far can we push mixed representations?
  – Unanalyzed cognitive tasks
  – Cognitive control at the task level

21

What We Stand to Gain

- Modeling is hard, expensive and time consuming
  – Better tools are needed
- Modeling is modeling
  – Differences in degree need not be taken as difference in kind
- A concrete framework in which to push principle into practice

22

ROBOTICS + THEORY **OF** MIND

ACT-R/E: E for Embodied

J. Gregory Trafton, Magdalena Bugajska, William Kennedy, Anthony Harrison, Benjamin Fransen, and Raj Ratwani
Corresponding Author: {greg.trafton@nrl.navy.mil}

ACT-R/E (for Embodied) (Kennedy, Bugajska, Adams, Schultz, & Trafton, 2008; Trafton, Bugajska, Fransen, & Ratwani, 2008) ventures beyond traditional computer displays and mouse/keyboard manipulation to establish embodied presence on a mobile robot by first and foremost extending the representation of the visual and aural modules to enable 3D object and sound localization. We also extended ACT-R's capabilities to incorporate a locomotion faculty (the "moval" module) and a spatial reasoning capability (the "spatial" module).

Our robot, an iRobot B21r, is a human-scale, zero-turn-radius robotic platform best suited for use in indoor environments. The robot is equipped with an array of sensors and effectors including an animated face (Parke & Waters, 1996; Simmons et al., 2003) displayed on a robot-mounted LCD , which allow it to perceive and interact with the environment. The raw sensors' input are processed by the low-level robotic software and translated into feature-based representation used by ACT-R/E modules as it becomes available. Our visual module is interfaced with the person-tracking (Fransen, et al., 2007) and color-blob detection software (Bruce, Balch, & Veloso, 2000) based on an omni-directional camera. Our auditory module interacts with sound localization software (Martinson & Brock, 2007) based on a 4-microphone array. Our spatial module has access to a 2D "cognitive map" (Kennedy, et al., 2007) or an egocentric version of spatial cognition (Harrison & Schunn, 2003). Requests to moval module in the form of relative or absolute motion-commands are passed onto our motion control subsystem which is integrated into our mobility system (Schultz, Adams, & Yamauchi, 1999). Similarly, speech module requests are forwarded to speech generation system. In addition, the change in the visual attention is indicated by turning to face the desired direction.

We describe a bottom-up model that uses ACT-R/E to integrate visual and auditory information to perform conversation tracking in a dynamic environment (Trafton, Bugajska, Fransen, & Ratwani, 2008). In our system, multiple conversationalists talk to each other and our model (embodied on the robot) "follows" (no understanding) the conversation, looking from person to person as they speak. The model takes the aural spatial information of the speaker (e.g., where the person is heard) and correlates it with the visual spatial information (e.g., where the person is seen) based on the stimuli proximity, and directs its visual attention (and its gaze) to the speaker. If a different person attempts to interrupt or backchannels, the model "ignores" the other speaker. However, once the original speaker is quiet for approximately 500ms (Bull & Aylett, 1998), the model switches its visual attention to the new speaker. This simple model was tested on a previously collected data set of conversation between 4 speakers (Vertegaal, et al., 2001) and matches the empirical data reasonably well without even addressing such important issues of visual participant monitoring or contextual responses.

Simulation within ACT-R as a Theory of Mind

William G. Kennedy, Magdalena D. Bugajska, Anthony M. Harrison, and J. Gregory Trafton
[bill.kennedy; magda.bugajska; anthony.harrison; greg.trafton]@nrl.navy.mil)
Naval Research Laboratory

From imitation behavior to interpersonal communication, successful strategies in humans clearly require consideration of others' knowledge, abilities, goals, and even feelings. The ability to infer that information and use it to simulate the behavior of others is referred to as having a Theory of Mind. Among several explanations of this capability is a simulation of the other based on the host, i.e., a "like-me" simulation (Meltzoff, 2005).

A general "like-me" mental simulation capability is already available within the basic ACT-R system. It is the running of a separate cognitive model (the "simulation") with a specified subset of the originating or "host" cognitive model. The simulation starts with a specific subset of the declarative and procedural memories and an initial goal state. The simulation runs this model and provides a new declarative fact via the "imaginal" buffer accessible by the host model. The specification of which declarative memory and productions of the host model to use allows the system to consider hypothetical and counterfactual situations.

We have developed cognitive models using "like-me" simulations of perspective taking (left-right handedness determinations), teamwork (predicting the teammate's decision), and social behavior (dominant-submissive behavior of chimpanzees). We will discuss the teamwork model, the simulation, the results, and matching of simulations and human/chimpanzee data.

References

Meltzoff, A. N. (2005). Imitation and Other Minds: The "Like Me" Hypothesis. In S. Hurley & N. Chater (Eds.), *Perspectives on Imitation: From Neuroscience to Social Science* (Vol. 2, pp. 55-77). Cambridge, MA: MIT Press.

Exploring Theory-of-Mind Components within Embodied Robotics

Anthony M. Harrison, William G. Kennedy, Benjamin Fransen, and J. Gregory Trafton ({anthony.harrison;bill.kennedy;ben.fransen;greg.trafton}@nrl.navy.mil)

Naval Research Laboratory

Examining the role of embodiment on performance has provided many insights into the underlying cognition across various low-level tasks (e.g. Fu & Gray, 2004; Salvucci & Gray, 2004). We have been looking at the embodied affordances of higher-level cognition, specifically theory-of-mind (Premack & Woodruff, 1978), within a common task as implemented on our robotics platforms (ACT-R/E).

The task examined was a chimpanzee food monopolization scenario introduced by Hare, et al (2000) and further refined by Brauer, Call & Tomasello (2007). The task pit dominant and subordinate chimpanzees against each other in the retrieval of one or two pieces of food in a shared space containing two buckets. The pieces of food were placed either on top of a bucket (visible) or behind the bucket such that only the subordinate could see it (hidden, from the dominant). The positions of the buckets were varied between two experiments. The key finding by Brauer, et al. (2007) was that in low risk situations, the subordinates frequently retrieved the food with no preference for the hidden pieces. However, when the risk was greater (i.e. food was closer to the dominant), subordinates approach the food less frequently but show a clear preference for the hidden pieces, suggesting that they were aware of what the dominant could see.

The use of primate data might seem odd, but it had a number of features that were quite appealing. First, this task has produced significant debate within the animal social cognition community regarding the ability of chimpanzees to understand what conspecifics can see (e.g. Hare et al., 2000; Karin-Darcy & Povinelli, 2002) and recent results suggest that it is the structure of the environment that influences the chimp's strategy use (Brauer, et al., 2007). Second, the task is fundamentally embodied requiring us to push the boundaries of our robot systems and ACT-R itself. Finally, the task lends itself to multiple solution strategies that are all intimately related to the theory-of-mind construct; specifically mental simulation (Meltzoff, 2005), perspective-taking (Flavell, et al., 1981), and gaze-following (Butterworth, 1991).

Three different models of the subordinate chimp were developed and refined in simulation before being deployed and evaluated on the robot. The first used ACT-R's model-within-model functionality to allow the subordinate to simulate where the dominant would search for food. The second used egocentric transformations of the perceived scene (Harrison, in prep) to adopt the perspective of the dominant and used that information to guess dominant's intent. The final model used perceptual knowledge of the dominant's head orientation and a modified visual search to perform gaze-following with approximate occlusion detection to infer the dominant's attentional focus.

The models highlight weaknesses in the current empirical methodology and point towards simple improvements that could help isolate what skills the chimpanzees are actually able to employ. In depth examination of the models' behaviors illustrate the different affordances of the strategies and provide insight into how and when the skills might be used in traditional theory-of-mind tasks.

ACT-R on a Robot: Considerations and Extensions

Eric Avery
Troy Kelley (tkelley@arl.army.mil)
Army Research Laboratory

At the Army Research Laboratory (ARL), we have developed a robotic system based on ACT-R, which includes both symbolic and sub-symbolic representation of knowledge. The system is called the Symbolic and Sub-symbolic Robotic Intelligence Control System (SS-RICS).

In recent years there has been a growing interest in using cognitive architectures for the control of robots (Avery, Kelley and Davani, 2006). While this seems to be a useful approach for robotic control, several considerations need to be taken into account before researchers attempt to use a cognitive architecture for robotic control.

The problem space used by ACT-R is primarily focused on working memory (WM) elements. Declarative memories, which are developed by the modeler, interact with procedural knowledge, in order to solve a specific problem. This is useful for researching and studying human decision making for a specific problem space, but not for robotic control, where many other aspects of memory need to be represented (i.e. Spatial Memory (SM), Iconic Memory (IM), Short Term Memory (STM) and Long Term Memory (LTM)). Instead of using one memory decay rate and a single retrieval threshold as defined by ACT-R for memories, we have found that within SS-RICS we needed to use different decay rates for SM, IM, STM, WM and LTM.

Additionally, it is unrealistic to assume that a modeler can develop all of declarative memory, so we have used ConceptNet (Liu and Singh, 2004) within SS-RICS to help alleviate the burden of memory development. This gives SS-RICS declarative memories to start with and use without the need to be developed by a modeler.

Also, it is unrealistic to assume the modeler will develop every production needed by a robotic system, so within SS-RICS we use a production system syntax (Verb, Noun, Adverb) which helps with production generation. Once this is used, other productions can be generated from this base production using machine learning techniques of substitution.

In order to use a symbolic cognitive architecture on a robotic system a developer must realize that cognitive architectures are intended to simulate human behavior not control a robot. We have used ACT-R as inspiration for working memory but in order to develop SS-RICS we have made additional changes for production generation, declarative memory development and memory decay rates.

References

Avery, E., Kelley, T. D., Davani, D. (2006). Using Cognitive Architectures to Improve Robot Control: Integrating Production Systems, Semantic Networks, and Sub-Symbolic Processing. *In Proceedings of 15th Annual Conference on Behavioral Representation in Modeling and Simulation (BRIMS).* May 15-19, Baltimore, MD.

Liu, H and Singh, P (2004). ConceptNet – a practical commonsense reasoning tool-kit. Kluwer Academic, BT Technology Journal, Vol. 22, Issue 4, pp. 211-226

ACT-R RESEARCH GROUP

CARNEGIE MELLON UNIVERSITY
DEPARTMENT OF PSYCHOLOGY
PITTSBURGH, PA 15213

P: 412**.268.**9883
F: 412**.268.**2844

HTTP://**ACT-R.PSY.CMU.EDU**

SUPPORTED BY GRANT N00014-03-1-0115 FROM THE OFFICE OF NAVAL RESEARCH

THANK YOU FOR JOINING US...

PLEASE TRAVEL SAFELY