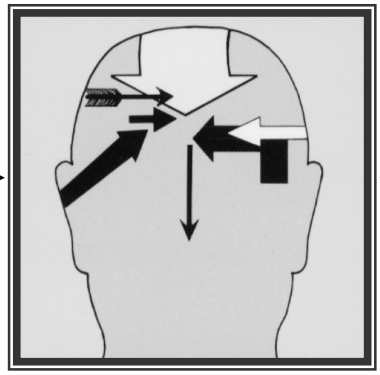


Thirteenth Annual

ACT-R Workshop Proceedings

July 21-23, 2006

Carnegie Mellon University
Department of Psychology
Pittsburgh, PA 15213
http://act-r.psy.cmu.edu/

Supported by Grant N00014-03-1-0115 from the Office of Naval Research.

Friday

7:45 Continental breakfast
8:15 Welcome

8:30 Five talks (20 minutes each)

10:10 Break

10:30 Five talks

12:10 Lunch

1:30-5:30 David Noelle, Leabra tutorial and discussion (with 3:30-4:00 break)

6:30-10:00 Party at the Pittsburgh Centre for the Arts, 6300 Fifth Avenue, Pittsburgh.

Saturday

7:45 Continental breakfast
8:30 Five talks

Sunday

7:45 Continental breakfast
8:30 Five talks

Reasoning/problem solving

Complex tasks

10:10 Break

10:30 Five talks

Emotion

Visual perception and Search

12:10 End

A New Utility Learning Mechanism

John R. Anderson
Carnegie Mellon University

Outline:

- The old mechanism.
- Fu & Anderson: Explorations with Reinforcement Learning.
- The new mechanism.

Conflict Resolution

- When multiple productions match, there is a conflict; ACT-R chooses the production with highest *expected gain.*
- *Example:* It's noon, you're hungry…

EatOnCampus:	EatOffCampus:
When I'm hungry and at CMU	When I'm hungry and at CMU
Then I switch my goal to find food at the University Center	Then I switch my goal to find food at a Squirrel Hill restaurant
Expected gain = 16	*Expected gain* = 9

Expected Gain

- E = expected gain of a production
- P = estimated probability that the current objective will be completed successfully if that production is fired
- G = the value of the current objective
- C = estimated cost of achieving the objective if that production is fired

Production	P	C	PG–C
EatOnCampus	.95	3	16
EatOffCampus	.95	10	9

- P, C (& thus E) are production specific

Chunk Choice Equation

$$\Pr obability(i) = \frac{e^{U_i/t}}{\sum_j e^{U_j/t}}$$

$$\sigma^2 = \frac{\pi^2}{6} t^2$$

Learning Conflict-Resolution Parameters

$$P = \frac{\text{Successes}}{\text{Successes} + \text{Failures}} \quad C = \frac{\text{Efforts}}{\text{Successes} + \text{Failures}}$$

$$Successes = n * PriorP + ExperiencedSuccesses$$

$$Failures = n * (1 - PriorP) + ExperiencedFailures$$

$$Efforts = n * PriorE + ExperiencedEfforts$$

Compiled production rules start with PriorP = 0, PriorE= G

Therefore, E=P*G-C = -G
It gets out of this hole by learning from parent each time it is recreated (subliminal learning):

PriorP = PriorP - α(ParentP - PriorP)
PriorE = PriorE - α(ParentE - PriorE)

Criticisms of the Existing Mechanism

- It treats success as an all-or-none mechanism.
- It does not extend naturally to situations where one is "harvesting" goods of different values.
- While the parameters are few -- G(value of goal), s(noise), n(weighting of priors), α(learning rate),
- The stored quantities are many -- E, P, C, Successes, Failures, Efforts, PriorP, Experienced Successes, Experienced Failures, PriorE, Experienced Efforts.
- It is not in keeping with the trend to reinforcement learning.

Virtues of Reinforcement Learning

- It makes the mechanisms of ACT-R more consistent with the ideas in the field.
- It is supposed to be what the basal ganglia do and that's where ACT-R has its productions
- It does deal nicely with varying valued goals and harvesting.
- It allows us to unite learning for compiled productions with learning from experience.
- However, it does have at least one Achilles heel when applied in a production rule framework.

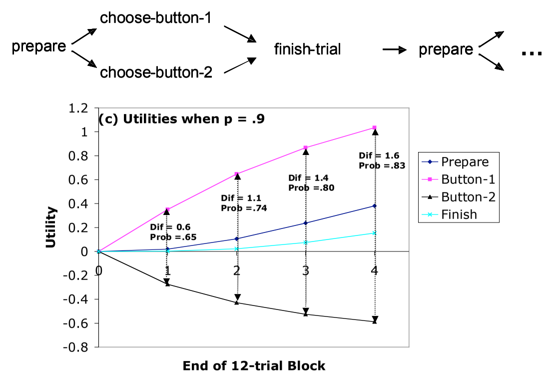
Fu & Anderson: Three Equations

$V_i(n) = V_i(n-1) + \alpha[R_i(n) - V_i(n-1)]$
where $V_i(n)$ is the value of production i after its nth application
$V_i(n-1)$ is the value of production i after its n-1st application
$R_i(n)$ is the reinforcement it receives after its nth application
α like previous system but the ONLY learning.
This is the simpler linear learning model that goes back to Bush and Mosteller (1955) and beyond, also known as simple integrator model

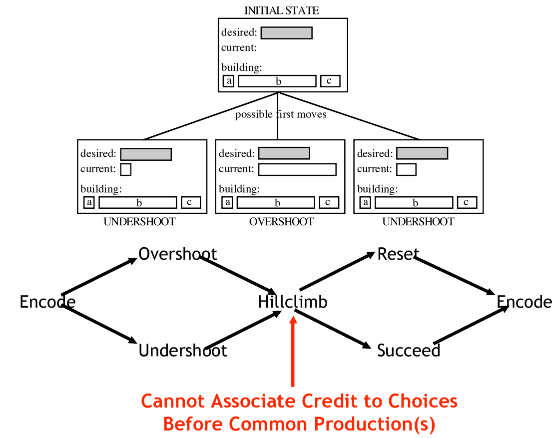
$R_i(n) = r_i(n) + g(t)\, V_{i+1}(n-1)$
where $r_i(n)$ is the actual reinforcement received
t is the time between this production and the next to fire, i+1
$V_{i+1}(n-1)$ is the value of production i+1 before it fires
g(t) is a discount of that value which increases with time

$g(t) = a^t$ normally (an exponential) but psychological research argues $g(t)= 1/(1+kt)$ (a hyperbolic -- a special form of a power function)

Illustration: Prob(button 1) = .9



The Achilles Heel



The Implemented Version

1. Just maintain the simple integrator equation:

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

2. Rewards occur potentially asynchronously of production firing -- can be implemented as separate reward module.
3. All productions from previous reward to current reward are credited with outcome but discounted according to how long it has been:

$$R_i(n) = R(n) - t_i(n)$$

4. By default all productions start at 0 and same rule applies to productions that fire and productions that are compiled.

Comments

- Naturally extends to harvesting situation with variable utilities.
- Reduces utility parameters to 1 -- just current utility.
- Two critical parameters are α that controls learning rate and s that controls noise.
- Implemented in newest version of tutorial and seems to work fine.
- However, it one can still choose old version of production learning.

Subtraction Example

Retrieve-Instruction (Reinforcement 10)
If the goal is to process a column
Then retrieve an operator for that kind of column

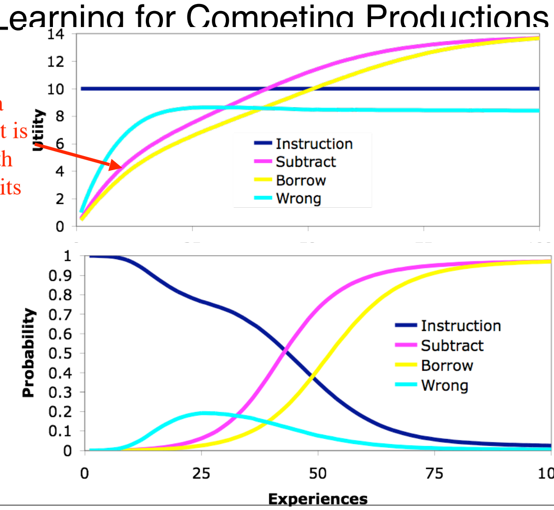
Request-Difference-Subtract (Reinforcement 14)
If the goal is to process a column
and the top digit is not smaller than the bottom digit,
Then subtract the bottom from the top

Request-Difference-Borrow (Reinforcemnt 14)
If the goal is to process a column
and the top digit smaller than the bottom digit,
Then subtract the top from the bottom

Request-Difference-Wrong (Reinforcement 14 or 0)
If the goal is to process a column
Then subtract the larger from the smaller

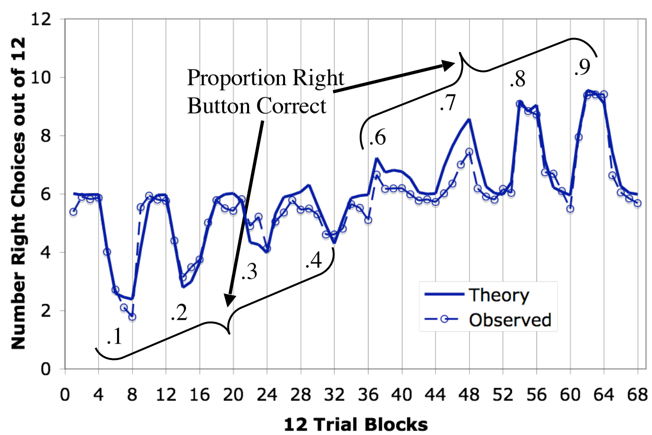
Utility Learning for Competing Productions

Every time a rule created it is rewarded with the utility of its parent

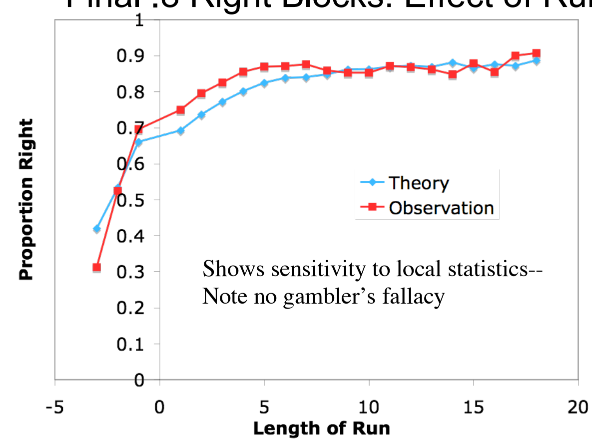


Probability Matching: Illustrate Utility Learning

Friedman, Burke, Cole, Keller, Millward, Estes, (1964)

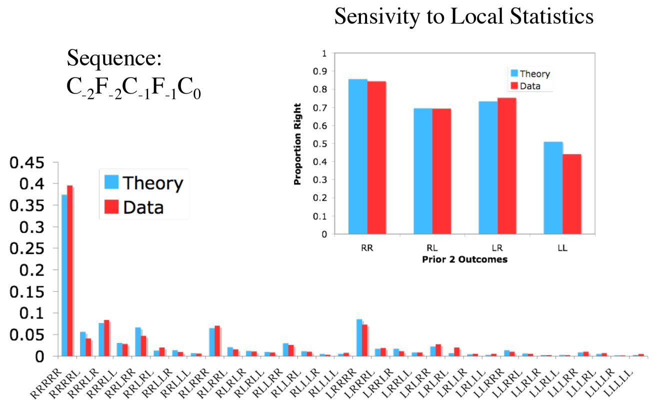


Final .8 Right Blocks: Effect of Run



Final .8 Right Blocks: Sequential Effects

Sequence:
$C_{-2}F_{-2}C_{-1}F_{-1}C_0$



Some Residual Issues

1. Do we want to maintain the simple exponential discounting of the past:

$$U_i(n) = \alpha \sum_{j=1}^{n} (1-\alpha)^{n-j} R_i(j)$$

Or do we want to go to a power function weighting:

$$U_i(n) = \sum_{j=1}^{n} t_j^{-d} R_i(j) \Big/ \sum_{j=1}^{n} t_j^{-d}$$

2. Should a compiled production be credited with the utility of its parent (currently implemented) or the utility actually experienced on that trail (probably better)?

Representing Human Spatial Competence in ACT-R

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

Introduction

Spatial cognition is a topic that has been explored using a variety of methodologies over the course of the last 60 years or more in psychological research. These studies have uncovered many phenomena relating to human (and animal) performance in spatial tasks. What has not emerged, however, is a unified account of the representations and mechanisms that enable human spatial competence across a variety of domains and diverse tasks. This is the goal pursued in this research.

The Theory

This account of human spatial competence is being developed within the context of the ACT-R cognitive architecture (Anderson et al., 2004). The proposal consists of adding a module to the existing architecture to perform spatial transformations, estimations, and computations. In addition, several buffers are proposed to augment the representation of spatial location in vision to be more consistent with the neuropsychological literature and to provide the functionality needed for ACT-R to operate in complex, 3-D spatial environments. Lastly, mechanisms are added to support mental imagery. The functional abilities that are proposed, as well as the anatomical locations in the brain to which they are ascribed, are supported by existing empirical, theoretical, and neuropsychological research.

The proposal as a whole integrates with the existing ACT-R architecture to create a system that inherits the existing benefits of ACT-R, while adding a theoretically and neuropsychologically motivated account of human spatial competence that extends the reach of ACT-R into new areas of research. A new architectural diagram, based upon the ideas described here is illustrated in Figure 1, including references to proposed brain areas as functional locations for each component.

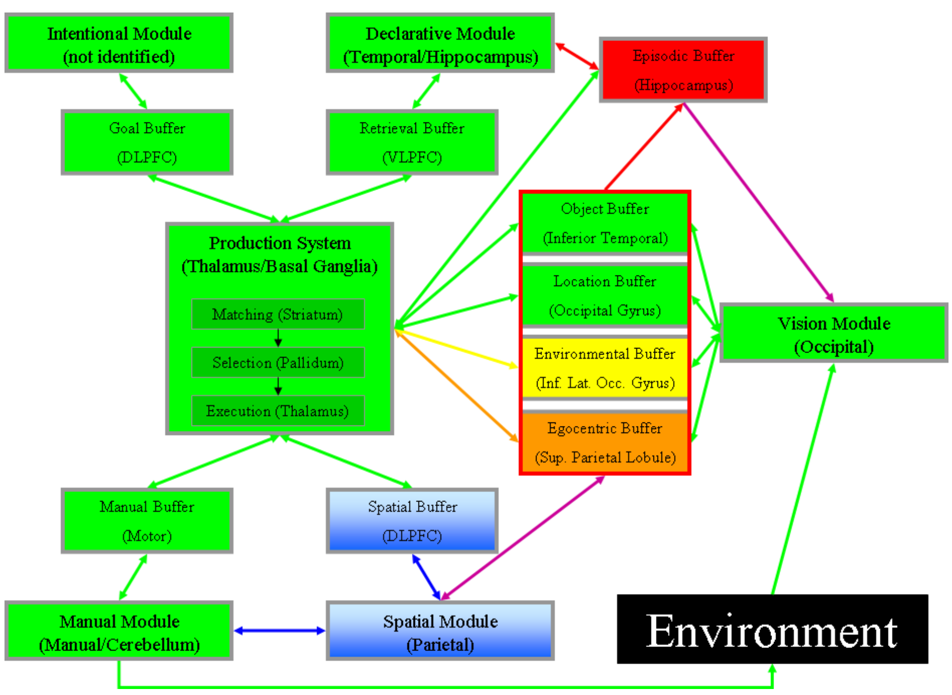


Figure 1. Schematic illustration of the current ACT-R architecture, with proposed additions included. Structures identified in green represent existing components of the architecture. Other colors represent proposed additions. The *environment* is indicated in black.

Conclusion

The account of human spatial competence presented here is broad, but is detailed enough to provide the foundation for computational accounts of a variety of cognitive phenomena involving spatial information processing. As the implementation of these mechanisms progresses, they will be validated against all available sources to ensure that they accurately capture the dynamics of spatial cognition in humans.

Acknowledgments

The research described here was supported by AFOSR Grant #02HE01COR. I would like to thank Don Lyon, Kevin Gluck, Greg Trafton, and Jerry Ball for their helpful comments on this work.

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.

Representing and Reasoning about Space

William G. Kennedy and J. Gregory Trafton
Naval Research Laboratory
2006 ACT-R Workshop

How best to represent and reason about space for a mobile robot is an open question. To extend our previous work in reasoning about space, specifically the hide & seek work (Trafton, et al, 2006), our next step is to develop a mobile robot that can covertly approach another robot or person. To do that, we need a good spatial representation that is useful for the cognition associated with the concepts of hiding and approaching. We also need to model the behavior of the other robot or person to build a form of situation awareness that supports determining when the robot can covertly approach.

We have a successful history with robots developing evidence grids based on existing sensors (Schultz and Adams 1998; Skubic, et al, 2004; Trafton, et al, 2006). We now want to focus on the more cognitive aspects involved rather than re-opening hardware issues. Therefore, we need to build on the evidence grids generated by our robot.

Our previous work in hide & seek did not really have a cognitive spatial module. It relied on the robot to implement spatial commands, such as “hide behind the box” with the location of “behind the box” being implemented by the robot’s hardware. The focus was modeling the learning of how to play the hiding side of the game and then using that knowledge in seeking. To address covertly approaching another agent, we need to do more detailed spatial reasoning at the cognitive level.

We are exploring how far we can get with a simple, metric-preserving, spatial representation supporting a cognitive model. We will describe the spatial representation and the cognitive functions it provides and our approach to acquiring the cognitive skills to covertly approach another robot or person.

We have a working demo.

References:

Trafton, J.G., Schultz, A.C., Perznowski, D., Bugajska, M.D., Adams, William, Cassimatis, N.L., Brock, D.P. (2006) Children and Robots Learning to Play Hide and Seek. In *Proceedings of the 2006 ACM Conference on Human-Robot Interaction*, Salt Lake City, Utah. ACM Press: New York.

Skubic, M., Perzanowski, D., Blizard, S., Schultz, A., Adams, W., Bugajska, M., and Brock, D. (2004) Spatial Language for Human-Robot Dialogs, IEEE Transactions on Systems, Man, and Cybernetics, 34(2), 154-167.

Schultz, A. and Adams, W. (1998) Continuous localization using evidence grids. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, IEEE Press: Leven, Belgium, 2833-2939.

Representing and Reasoning about Space

StealthBot

ACT-R Workshop 2006
July 2006

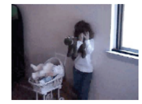
Bill Kennedy and Greg Trafton
Naval Research Lab

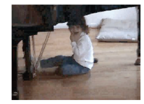
1

Previous Work

Hide & Seek

- Created a mobile robot that can play "hide and seek"
- Vision & navigation handled by robot platform
- Learned the effectiveness of hiding places based on feedback
- Successfully paralleled learning of the game by a 3½ year old child (Trafton, et al, 2006)
- Limited spatial reasoning, e.g., "behind" concept implemented by the robot





2

StealthBot

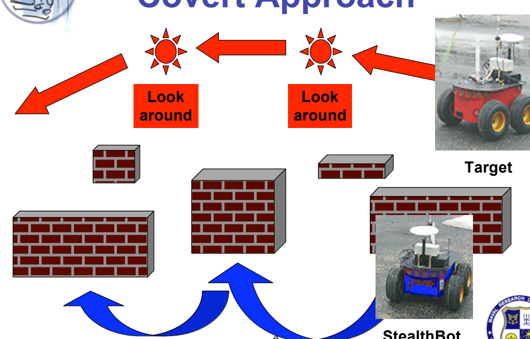
Goal: to create a mobile robot that can covertly approach another agent (person or robot) in a variety of situations (e.g., autonomously, as part of a team, in different environments, etc.)

Extends previous research:

- Builds on NRL work in perspective-taking (Hiatt, 2004; Trafton, et al 2005)
- Adds spatial reasoning (considering Scott Douglass's work)
- Awaiting cognitive vision system
- Adds cognitive modeling of target, specifically its movement and sensor capabilities and use
- Learns effectiveness of hiding places based on spatial reasoning and experience
- Where to hide based on cognitive model of target

3

Covert Approach



4

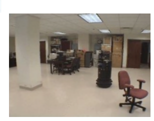
Robot platform provides

- Vision:
  - Moving target recognition
  - Stationary object recognition
  - Scene metrics
- Voice, NL, and gesture recognition
- Generated voice output
- Navigation, collision avoidance

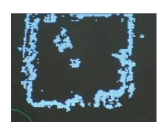
(Schultz and Adams 1998; Skubic ,et al 2004)

5

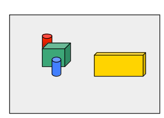
Representations



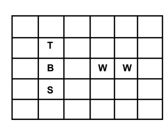
Visual: beautiful, interpretation?



Robot sensors: metrics, but ugly!



Symbolic: pretty, but useful?



Cognitive map: useful

6

Spatial Representation

- Robot provides:
  - Object locations
  - Object characteristics (size, height, etc.)
- StealthBot (ACT-R) software:
  - Builds/updates a cognitive map
  - Builds linear models of target movement
  - Inserts chunks into declarative memory for
    - Updated target location
    - Change in linear model

7

Using the Cognitive Map

- Map provides relation to closest object
  eg: relation: north-of
  ref-object: wall34

- Provides distances (3 ranges) to determine closest object to target & to StealthBot
- Can 'mentally' walk map to evaluate visibility

8

Cognitive Map Plausibility

- Basically working in a 2D world
- Hard constraints from robot's metrics
- Ego-centric referencing by robot difficult
- Ego-centric buffer not in ACT-R (yet)
- Appropriate balance between cognitive plausibility and AI functionality for us

9

Modeling Target Motion

- Kalman Filters not used to allow reasoning
- Target movement modeled in 3 levels:
  tactics, strategies, missions
- Lowest level implemented in Lisp
- Higher levels implement in ACT-R

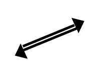
10

Modeling Target Motion

- Tactic: a linear model: 
- Strategy: series of tactics: 
- Mission: purpose of strategy: "transit area"

11

Modeling Target Motion

- Tactic: a linear model: 
- Strategy: series of tactics: 
- Mission: purpose of strategy: "patrol"

12

Spatial Reasoning

- **Productions to "covertly approach" based on:**
  - Target's location
  - Target model (tactic, strategy, mission)
  - Objects' locations & characteristics
  - Hidden? fnc(target location, objects, sensor)
  - Closer to target?
  - Path from here to there hidden?

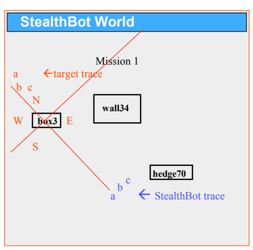
13

Summary

- **Cognitive Level (ACT-R)**
  - Functions: deciding where & when to move
  - Objects: hiding places, target's strategies & mission
  - Tools: ACT-R productions
- **Symbolic Level (Lisp associated with ACT-R)**
  - Functions: translation between cognitive & sensor levels
  - Objects: object characteristics & models of movement/sensors
  - Tools: cognitive map
- **Sensor Level (C++/hardware dependent)**
  - Functions: sensing environment & robot navigation
  - Objects: sonar readings, voice output, scene metrics
  - Tools: Evidence grids, Kalman filters

14

Experimental Display



15

References

Hiatt, L. M., Trafton, J. G., Harrison, A.M., & Schultz, A. C. (2004) A cognitive model for spatial perspective taking. In *Proceedings of the sixth International Conference on Cognitive Modeling* (pp. 354-355). Pittsburgh, PA: Carnegie Mellon University/University of Pittsburgh.

Skubic, M., Perzanowski, D., Blizard, S., Schultz, A., Adams, W., Bugajska, M., and Brock, D. (2004) Spatial Language for Human-Robot Dialogs, IEEE Transactions on Systems, Man, and Cybernetics, 34(2), 154-167.

Schultz, A. and Adams, W. (1998) Continuous localization using evidence grids. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, IEEE Press: Leven, Belgium, 2833-2939.

Trafton, J. G., Cassimatis, N. L., Bugajska, M., Brock, D., Mintz, F., Schultz, A. (2005). Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man and Cybernetics*, pp. 460-470. Volume 25. Issue 4.

Trafton, J.G., Schultz, A.C., Perznowski, D., Bugajska, M.D., Adams, William, Cassimatis, N.L., Brock, D.P. (2006) Children and Robots Learning to Play Hide and Seek. In *Proceedings of the 2006 ACM Conference on Human-Robot Interaction*, Salt Lake City, Utah. ACM Press: New York.

16

A color perceptual process theory: Letting ACT-R see colors

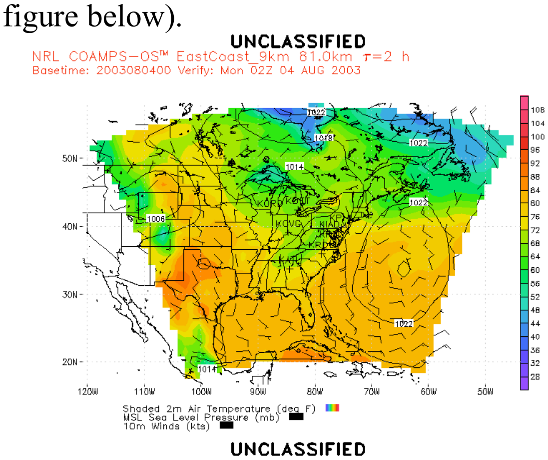
Greg Trafton (NRL)
Raj Ratwani (GMU/NRL)
Len Breslow (NRL)

Color is a core component of our visual system, yet many cognitive theories do not handle colors well, even though good color theories and spaces exist (e.g., the CIE* color spaces). ACT-R is able to see colors, but it 'perceives' a blue R (R) as:

```
TEXT0-0
  ISA TEXT
  SCREEN-POS  LOC1-0
  VALUE  "R"
  STATUS  NIL
  COLOR  BLUE
  HEIGHT  13
  WIDTH  9
```

For ACT-R (and all other cognitive architectures), color is represented propositionally. This approach works fine when all that is needed is the actual color (e.g., you remember seeing the R as blue). However, this approach does not work well when the perception of color is important. For example, ACT-R has problems deciding whether a color is lighter or darker than another, finding patterns within stimuli that are color-coded, etc.

Color perception is also critical for perceiving graphs and visualizations (e.g., a meteorological display). Complex visualizations frequently use color to represent quantitative data (see figure below).



I will present some experimental data and a new color buffer (part of the vision module) that is able to perceptually see colors. The new color buffer is able to determine whether two colors are the same. The new color buffer is also able to determine which of two colors are lighter (or darker).

The color buffer is able to account for several empirical effects with relatively few free parameters.

An ACT-R Timing Module Based on the Attentional Gate Model

Michael D. Byrne
Department of Psychology
Rice University
Houston, TX
byrne@acm.org
http://chil.rice.edu/

THE COMPUTER-HUMAN INTERACTION LABORATORY AT RICE UNIVERSITY

Overview

- Why another temporal module?
  - Review of current temporal module
  - Time perception phenomena
    - Timing multiple intervals
    - Underestimation and load effects
- The Attentional Gate Model
- An ACT-R implementation
- Issues and future work

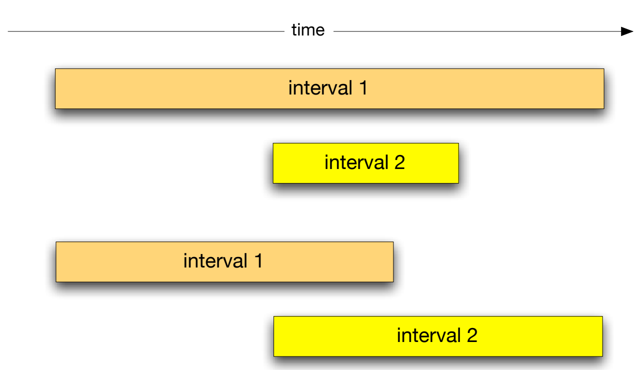
IMTS INNOVATIVE MANAGEMENT & TECHNOLOGY SERVICES

Review of Taatgen Temporal Module (TTM)

- Module contains a counter which starts at zero and “ticks” every so often, with ticks being spaced further apart the longer the interval
  - Timing short intervals more accurate than long intervals
- Productions can read off the current value of the counter, which is noted in a chunk in the temporal buffer
  - Conversion to desired interval typically done via retrieval
    - For example, what was the counter when I last experienced the desired amount of time?
  - Interference possible as time-reading productions compete with other productions when busy
- Counter can be reset when needed

First Issue: Timing Multiple Intervals

- In modeling a complex task with many time-sensitive components, timing of multiple overlapping intervals appears necessary



- Non-uniform tick size makes this very difficult, particularly if SOAs are not constant

Second Issue: Underestimation and Load Effects

- One of the primary findings in the temporal literature is that people systematically *underestimate* intervals under most conditions
  - In fact, proportion of true interval has been taken as the dependent measure of choice for numerous meta-analyses
    - For most experiments, it's < 1.0
- Most powerful independent variable: cognitive load
  - More load during interval yields greater underestimation
  - Suggests timing is attention- or resource-demanding
- Problematic for TTM: only load at the *end* of the interval can interfere and produce underestimation
- Some other smaller issues as well

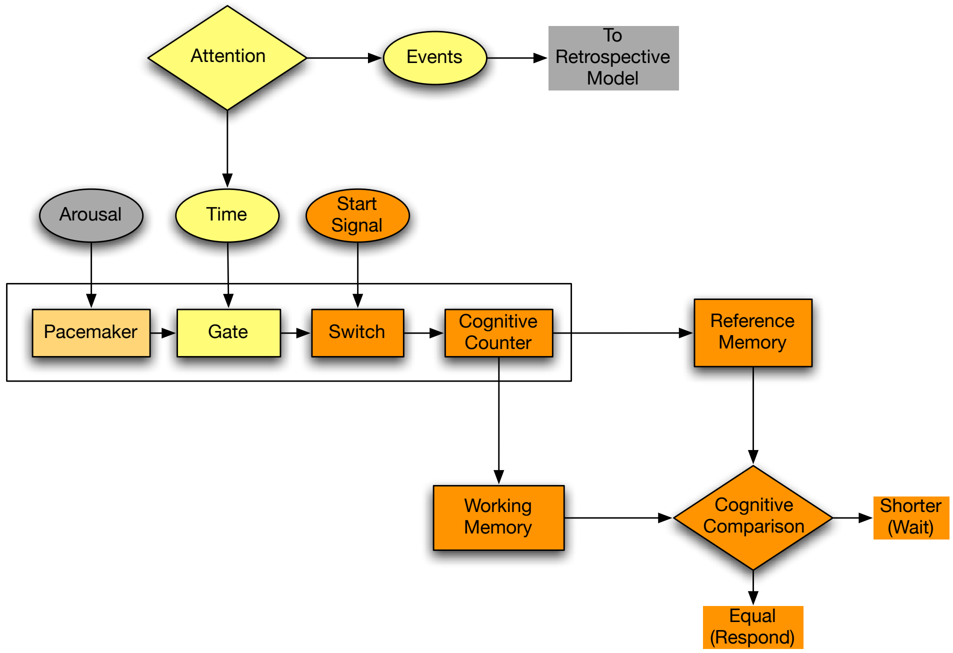
5

Alternate Account

- My conundrum: TTM is great, but not quite what I want, and yet I have no desire to re-invent the wheel here
- My solution? See if there's a nice, ACT-R compatible model out there and "borrow" it (cf. EPIC→ACT-R/PM)
  - Grad student in another lab in my department fortunately doing her dissertation on a related topic, pointed me to her candidate for "best account so far"
- The Attentional Gate Model (AGM) of Block and Zakay (Block & Zakay 1996; Zakay & Block, 1997)
  - Block and Zakay are two of the more well-known names in the temporal domain
  - And the AGM turns out to be somewhat similar to the TTM

6

AGM vs. Taatgen Temporal Module



AGM vs. Taatgen Temporal Module

- Pacemaker
  - AGM pacemaker uses a fixed mean pulse rate (termed Λ) rather than an increasing one
    - Handle overlapping intervals by subtraction (messy)
  - Also noisy (logistic, *s* parameter termed Λ_s)
    - Note: timing of long intervals still less accurate than short (noise summation yields increased variance as time passes)
- 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
  - Means pulses can be missed, producing underestimates

AGM Module Operations

- Start the pacemaker with this on RHS:

```
+temporal>
  isa  start
```

- Inquire about the presence of a new pulse via LHS:

```
?temporal>
   new-pulse  yes
```

- Increment the counter via RHS

```
+temporal>
   count    increment
```

- Read off the counter with LHS:

```
=temporal>
  isa     counter
  count   =c
```


Example Productions

```
(p start
   =goal>
     isa     time-goal
     done    start
==>
  +temporal>
     isa     start
  =goal>
     done    nil
)
```

```
(p count
   =goal>
     isa     time-goal
     done    nil
   ?temporal>
     new-pulse  yes
   =temporal>
     isa     counter
==>
   +temporal>
     count      increment
)
```

```
(p retrieve
   =goal>
     isa    time-goal
     done      nil
   =temporal>
     isa    counter
     count  =c
   ?retrieval>
     state  free
==>
   +retrieval>
     isa    time-mem
<    count  =c
)
```

```
(p stop
   =goal>
     isa     time-goal
     done    nil
   =retrieval>
     isa     time-mem
==>
   +temporal>
     isa     stop
   =goal>
     done t
)
```

Issues and Future Work

- Not yet entirely clear what parameters (pulse rate and noise) are generally most appropriate
  - Need more models which use the system!
- The two temporal modules make slightly different predictions, someone should do the experiments:
  - Heavy vs. light cognitive load during interval, but not at the ends of the interval
    - AGM predicts increased underestimation, TTM predicts no effect
  - Multiple embedded/overlapping time intervals (varying SOA critical)
    - AGM says doable but hard, TTM says nearly impossible (I think)

References

Block, R. A., & Zakay, D. (1996). Models of psychological time revisited. In H. Helfrich (Ed.), *Time and mind* (pp. 171–195). Kirkland, WA: Hogrefe & Huber.

Block, R. A., & Zakay, D. (1997). Prospective and retrospective duration judgments: A meta-analytic review. *Psychonomic Bulletin & Review, 4*(2), 184–197.

Taatgen, N. A., Anderson, J. R., Dicksion, D., & van Rijn, H. (2005). Time interval estimation: Internal clock or attentional mechanism? In *Proceedings of the Twenty-Seventh Annual Meeting of the Cognitive Science Society* (pp. 2122–2127). Mahwah, NJ: Erlbaum.

Taatgen, N. A., van Rijn, H., & Anderson, J. R. (2004). Time perception: Beyond simple interval estimation. In C. D. Schunn, M. C. Lovett, C. Lebiere & P. Munro (Eds.), *Proceedings of the Sixth International Conference on Cognitive Modeling* (pp. 296–301). Mahwah, NJ: Erlbaum.

Zakay, D., & Block, R. A. (1997). Temporal cognition. *Current Directions in Psychological Science, 6*(1), 12–16.

Four levels of communication, errors, and recovery in ACT-R

Mike Matessa
Alion Science & Technology

Moving ACT-R out of the desktop

- ACT-R bot in a complex environment
- Communication is more than just talking
  - Attention direction
  - Object placement
- Communication has internal & external parts
  - Internal levels of processing
  - External feedback from partner for all levels
- Understanding communication allows error prediction and error recovery

ACT-R levels of processing

ACT-R level	Module request	Example
1. Attention	1. Visual location	movement loc.
2. Identification	2. Visual object	moving guide
3. Retrieval	3. Declarative chunk	instr. to follow
4. Action	4. Motor action	move wheels

Levels of processing comparison

ACT-R level	Clark (1996) level
1. Attention	1. Attention
2. Identification	2. Identification
3. Retrieval	3. Understanding
4. Action	4. Negotiation

Clark levels of processing

Clark level	Sender	Receiver
1. Attention	Execute	Attend
2. Identification	Present	Identify
3. Understanding	Signal	Recognize
4. Negotiation	Propose	Consider

Evidence of completion:
Sender needs evidence from Receiver in order to complete level

Downward evidence:
Evidence of higher number is evidence of all lower

Clark levels of processing

Clark level	Example: “I’ll be right there”
1. Attention	1. Attend to voice
2. Identification	2. Identify English expression
3. Understanding	3. Recognize meaning: delay
4. Negotiation	4. Consider accepting delay

Clark errors

Example: "I'll be right there"	Possible error
1. Attend to voice	1. Can't hear at all
2. Identify English expression	2. "Have a white chair"
3. Recognize meaning: delay	3. Right there = 5 sec
4. Consider accepting delay	4. Can't wait

ACT-R errors

ACT-R level	Possible error
1. Attention	1. No visual location returned
2. Identification	2. No visual object returned
3. Retrieval	3. No chunk returned
4. Action	4. Motor module error

ACT-R recovery

ACT-R level	Possible recovery request to partner
1. Attention	1. “I can’t find what I’m looking for”
2. Identification	2. “I don’t know what that is”
3. Retrieval	3. “I forget what I’m supposed to do”
4. Action	4. “I can’t move”

Evidence of completion:
Sender needs evidence from Receiver in order to complete level

Downward evidence:
Evidence of higher number is evidence of all lower

ACT-R recovery

ACT-R level	Possible recovery action by partner
1. Attention	1. Make sure obj. in range of sensor
2. Identification	2. Adjust orientation for easier ID
3. Retrieval	3. Prompt bot with instruction
4. Action	4. Change environment to allow action

Conclusions

- Putting ACT-R in complex environments requires a richer representation of communication than passing symbols
- Levels of processing in ACT-R can be used as a rich representation
- Levels can be used to predict communication errors
- Levels can be used as feedback to partner to recover from errors

Other uses for the four levels

- Understanding task complexity
  - Attention: number of distractors in environment
  - Identification: number of object categories
  - Retrieval: number of task instructions
  - Action: number of obstacles to action

Human cognition as a winning design: Learning algebra by exploration

Angela Brunstein (angelab@cmu.edu)
Carnegie Mellon University

John R. Anderson (ja+@cmu.edu)
Carnegie Mellon University

Learning algebra

A unique human cognitive achievement is to master tasks and situations it was originally not designed for (Anderson, in preparation). One of these artificial tasks concerns learning algebra. Almost all of us master it even though this ability is not directly vitally important to us.

One way to learn algebra is to use the algebra tutor (XXX). For adult learners, the challenge here is more to learn how to interact with the interface than the underlying algebra. First, interacting with the tutor requires performing intermediate steps most experts skip. Second, the tutor demands a rigid order of steps and operations. Third, not all relations in the display can be mapped directly to algebraic relations. In our actual study, in worst case it took participants 177 steps in addition to 21 required steps for evaluating a diagram when exploring the algebra tutor. On average, it took them 14 in addition to 25 required steps to solve the diagrams presented.

Nevertheless, all of our 40 participants were able to solve all 175 linear algebra problems presented by the tutor even given only minimal instruction. Reasons for this cognitive masterpiece could be, first the low degrees of freedom that make it easier to detect the structure behind the reactions of the tutor when interacting. Second, the four plus steps were always the same for all kinds of operations in the tutor. Third, the tutor always provides a feedback after interactions: Either the display changes after successful interaction, an error message pops up, or the tutor doesn't react at all. Forth, participants could use their mathematical pre knowledge for deducting hypotheses on how the tutor works. So participants with minimal instruction needed in best case no steps in addition to required steps at all when exploring how the tutor works.

What humans do

One example of this cognitive achievement is the very first diagram presented corresponding to (5+7)*8. This display consists of a (5+7) box feeding into a (x*8) box which feeds into an empty resulting box. For solving this problem, participants had to select the (5+7) box, to press the Evaluate button, to click the green box that pops up for taking in the result and to enter the result. Thereafter, they had to repeat the same steps for evaluating (12*8). Finally, they had to press the next-problem button for getting the following task.

What participants do in this situation is typically to immediately calculate (5+7) and try to enter the result somehow. For reaching this goal they systematically try to type in the result without or after clicking involved boxes and buttons. More generally, behavior of participants seems to be triggered by eye-catching operations they know how to perform instead of global strategies how to clean up the display in a most efficient manner. Therefore, they seem to be guided by their mathematic expectations and later on by their experiences on how the tutor works when interacting with the interface. For the (5+7) case, participants typically first try to enter 12 without selecting any boxes. The tutor doesn't react to this action. So they try as next to select the (x*8) box and to type in the result. That is because this is the box where the result of evaluation has to be entered. Again the tutor does not react. Alternatively, participants try to select the (x*8) box and to click the Evaluate button. This time the tutor answers 'Evaluate can't be done with the selected box.' Therefore participants select as next either both, the (5+7) and (x*8) boxes, or only the (5+7) box followed by clicking the Evaluate button. Mostly participants now click the popping up green box for entering the result '12'. Otherwise they systematically try out what to enter guided by the 'Your answer is incorrect,' feedback of the tutor.

What the model could do

There are 3 kinds of implications for teaching the algebra model how to perform this task. First, in principle the model would be able to calculate eye-catching operations by cued retrieval. It could also scan the display for promising operations when deciding what to do next.

Second, what the model can't provide in the moment is to order goals and acquired knowledge hierarchically as participants do. They describe that for evaluating, they search for a box to be evaluated… In addition, the actual version of model does not map its algebraic pre knowledge to tutor states when discovering how to perform.

Third and even harder to achieve, is to let the model act by its expectations how the tutor should behave, systematically and not per random try out what to do next, and repeat operations for memorizing order of steps associated. Getting the algebra model there would mean to make it much more similar to that winning human design of mastering situations it was originally not designed for.

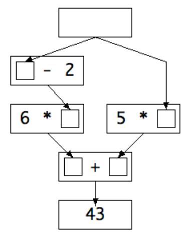
Acknowledgement

This project is partly founded by the Alexander von Humboldt Foundation.

Human cognition as a winning design: Learning algebra by exploration

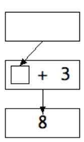
Angela Brunstein & John R. Anderson
Carnegie Mellon University

A winning design



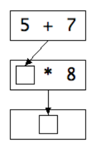
- designed to learn tasks we are not designed for: Learning algebra (Anderson, 2007)
- Algebra: artificial, but almost all can master it
- Algebra tutor: hard & complex - handling the tutor, not the algebra behind

Learning how to handle the tutor



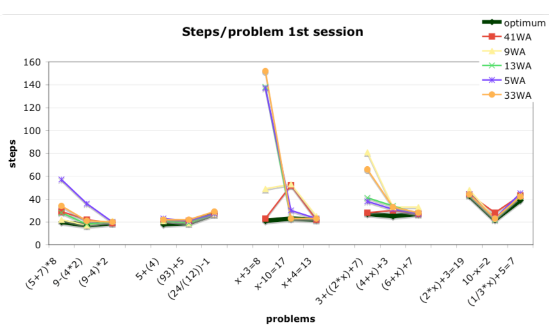
- Why hard to perform?
  - Required steps experts skip
  - Strictly determined order of operations
  - At least 4 steps per operation
  - Not completely matching to doing algebra
  - Worst cases: 21 + 177 steps for x+3 = 8 (*M* = 14)

Learning how to handle the tutor

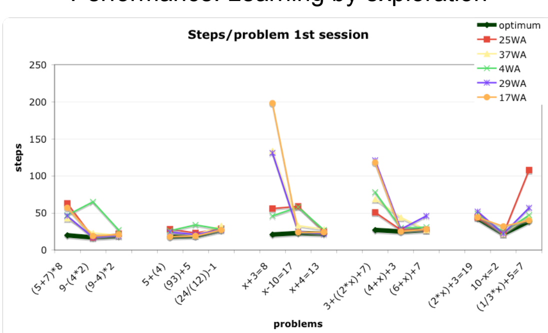


- Why feasible?
  - Strictly determined order of operations
  - Stereotypic order of steps per operation
  - Direct feedback (exception: miscalculating)
  - Math pre knowledge (arithmetic, maybe algebra)

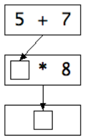
Performance: Learning by exploration



Performance: Learning by exploration

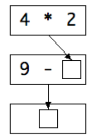


Steps 1st problem: evaluate (13)



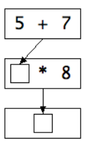

```
   (PROBLEM-START 8026208 (:BOX1 (:BOX5 * (:BOX8 + 5 7) 8)))
 1 (READ-INSTRUCTIONS-DONE 8053441)
 2 (SELECT 8058471 (:BOX5 (:BOX8 * 8) (:SELECTED)))
 3 (SELECT 8063111 (:BOX8 (5 + 7) (:SELECTED)))
 4 (UNSELECT 8064223 (:BOX5 (:BOX8 * 8) NIL))
 5 (UNSELECT 8070950 (:BOX8 (5 + 7) NIL))
 6 (SELECT 8071134 (:BOX8 (5 + 7) (:SELECTED)))
 7 (OPERATION 8074224 EVALUATE SUCCESSFUL CLICKED (:BOX1 (:BOX5 * (:BOX8 + 5 7 :SELECTED) 8)))
 8 (KEYPRESS 8082355 #\1)
 9 (OPEN-DIALOG 8084965 ARG :ARG4)
10 (INPUT-OPEN 8084967)
11 (TYPED-INPUT 8087163 #\1)
12 (TYPED-INPUT 8088346 #\2)
13 (INPUT-TYPE 8090643 SUBMIT)
14 (INPUT 8090645 12 "12" (:BOX5 (FILL * 8) (:ARG1-EDITABLE)))
15 (SUBMIT 8090650 NIL (:BOX1 (:BOX5 * 12 8)))
16 (SELECT 8135920 (:BOX5 (12 * 8) (:SELECTED)))
17 (OPERATION 8139209 EVALUATE SUCCESSFUL CLICKED (:BOX1 (:BOX5 * 12 8 :SELECTED)))
18 (OPEN-DIALOG 8141599 ARG :ARG0)
19 (INPUT-OPEN 8141602)
20 (TYPED-INPUT 8145381 #\9)
21 (TYPED-INPUT 8145925 #\6)
22 (INPUT-TYPE 8154349 SUBMIT)
23 (INPUT 8154351 96 "96" (:BOX1 :ARG0 (:ANSWER-EDITABLE)))
24 (SUBMIT 8154593 NIL (:BOX1 96))
25 (CLICK 8157030 NIL 47 24)
26 (CLICK 8157678 NIL 43 17)
27 (CLICK 8162693 NIL 39 17)
28 (PROBLEM-END 8166374 0 CLICKED)
```

Steps 2nd problem: evaluate (13)



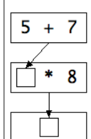

```
   (PROBLEM-START 8166392 (:BOX35 (:BOX37 - 9 (:BOX40 * 4 2))))
 1 (SELECT 8176244 (:BOX40 (4 * 2) (:SELECTED)))
 2 (OPERATION 8179126 EVALUATE SUCCESSFUL CLICKED (:BOX35 (:BOX37 - 9 (:BOX40 * 4 2 :SELECTED))))
 3 (OPEN-DIALOG 8180716 ARG :ARG37)
 4 (INPUT-OPEN 8180718)
 5 (TYPED-INPUT 8183489 #\8)
 6 (INPUT-TYPE 8186570 SUBMIT)
 7 (INPUT 8186572 8 "8" (:BOX37 (9 - FILL) (:ARG2-EDITABLE)))
 8 (SUBMIT 8186576 NIL (:BOX35 (:BOX37 - 9 8)))
 9 (SELECT 8188979 (:BOX37 (9 - 8) (:SELECTED)))
10 (OPERATION 8191645 EVALUATE SUCCESSFUL CLICKED (:BOX35 (:BOX37 - 9 8 :SELECTED)))
11 (KEYPRESS 8193928 #\1)
12 (OPEN-DIALOG 8195378 ARG :ARG34)
13 (INPUT-OPEN 8195380)
14 (TYPED-INPUT 8196328 #\1)
15 (INPUT-TYPE 8197656 SUBMIT)
16 (INPUT 8197659 1 "1" (:BOX35 :ARG34 (:ANSWER-EDITABLE)))
17 (SUBMIT 8197663 NIL (:BOX35 1))
18 (PROBLEM-END 8207666 14 CLICKED)
```

Steps 1st problem: evaluate (29)



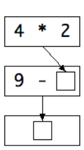

```
   (PROBLEM-START 5395453 (:BOX1 (:BOX5 * (:BOX8 + 5 7) 8)))
 1 (READ-INSTRUCTIONS-DONE 5406656)
 2 (SELECT 5413047 (:BOX5 (:BOX8 * 8) (:SELECTED)))
 3 (KEYPRESS 5417440 #\1)
 4 (KEYPRESS 5417532 #\2)
 5 (UNSELECT 5418726 (:BOX5 (:BOX8 * 8) NIL))
 6 (SELECT 5419726 (:BOX5 (:BOX8 * 8) (:SELECTED)))
 7 (UNSELECT 5420246 (:BOX5 (:BOX8 * 8) NIL))
 8 (SELECT 5422510 (:BOX5 (:BOX8 * 8) (:SELECTED)))
 9 (OPERATION 5425705 EVALUATE UNSUCCESSFUL CLICKED (:BOX1 (:BOX5 * (:BOX8 + 5 7) 8 :SELECTED)))
10 (UNSELECT 5430365 (:BOX5 (:BOX8 * 8) NIL))
11 (SELECT 5430549 (:BOX5 (:BOX8 * 8) (:SELECTED)))
12 (UNSELECT 5431181 (:BOX5 (:BOX8 * 8) NIL))
13 (KEYPRESS 5432010 #\1)
14 (KEYPRESS 5432098 #\2)
15 (SELECT 5432685 (:BOX5 (:BOX8 * 8) (:SELECTED)))
16 (KEYPRESS 5433298 #\1)
17 (KEYPRESS 5433362 #\2)
18 (SELECT 5440676 (:BOX8 (5 + 7) (:SELECTED)))
19 (OPERATION 5443256 EVALUATE UNSUCCESSFUL CLICKED (:BOX1 (:BOX5 * (:BOX8 + 5 7 :SELECTED) 8 :SELECTED)))
20 (OPERATION 5444945 COMBINE UNSUCCESSFUL CLICKED (:BOX1 (:BOX5 * (:BOX8 + 5 7 :SELECTED) 8 :SELECTED)))
21 (CLICK 5447083 NIL 24 -30)
22 (UNSELECT 5447731 (:BOX8 (5 + 7) NIL))
23 (UNSELECT 5448203 (:BOX5 (:BOX8 * 8) NIL))
```

Steps 1st problem: evaluate (29)



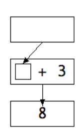

```
24 (SELECT 5448667 (:BOX8 (5 + 7) (:SELECTED)))
25 (OPERATION 5451229 EVALUATE SUCCESSFUL CLICKED (:BOX1 (:BOX5 * (:BOX8 + 5 7 :SELECTED) 8)))
26 (OPEN-DIALOG 5457954 ARG :ARG4)
27 (INPUT-OPEN 5457956)
28 (OPN-DURING-SUBMIT 5459548 EVALUATE CLICKED)
29 (TYPED-INPUT 5466472 #\1)
30 (TYPED-INPUT 5466528 #\2)
31 (INPUT-TYPE 5468296 SUBMIT)
32 (INPUT 5468298 12 "12" (:BOX5 (FILL * 8) (:ARG1-EDITABLE)))
33 (SUBMIT 5468303 NIL (:BOX1 (:BOX5 * 12 8)))
34 (SELECT 5469905 (:BOX5 (12 * 8) (:SELECTED)))
35 (OPERATION 5473171 EVALUATE SUCCESSFUL CLICKED (:BOX1 (:BOX5 * 12 8 :SELECTED)))
36 (OPEN-DIALOG 5475001 ARG :ARG0)
37 (INPUT-OPEN 5475003)
38 (TYPED-INPUT 5481806 #\9)
39 (TYPED-INPUT 5482182 #\6)
40 (INPUT-TYPE 5482767 SUBMIT)
41 (INPUT 5482769 96 "96" (:BOX1 :ARG0 (:ANSWER-EDITABLE)))
42 (SUBMIT 5482772 NIL (:BOX1 96))
43 (CLICK 5484936 NIL 59 11)
44 (CLICK 5486312 NIL 64 28)
45 (CLICK 5486472 NIL 64 28)
46 (PROBLEM-END 5489193 0 CLICKED)
```

Steps 2nd problem: evaluate (29)

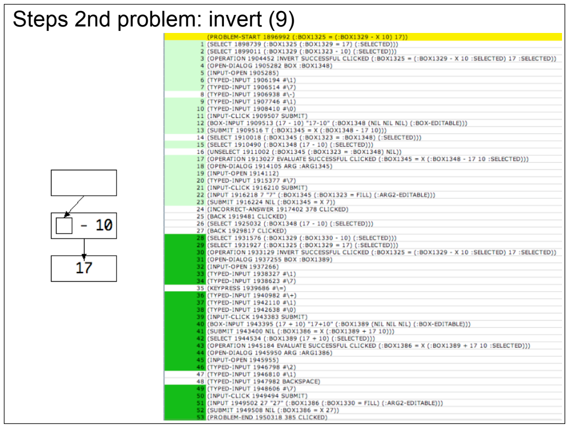


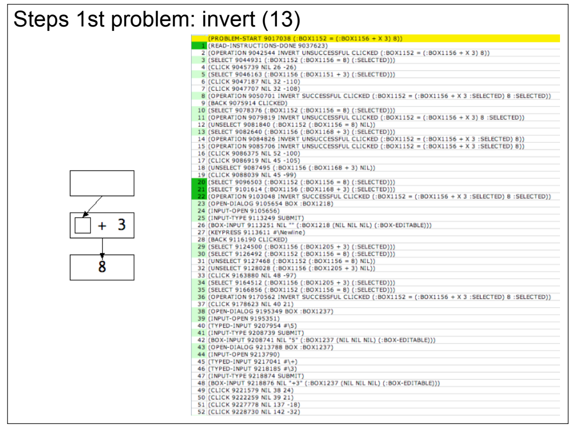

```
   (PROBLEM-START 5489211 (:BOX56 (:BOX58 - 9 (:BOX61 * 4 2))))
 1 (SELECT 5492575 (:BOX61 (4 * 2) (:SELECTED)))
 2 (OPERATION 5494761 EVALUATE SUCCESSFUL CLICKED (:BOX56 (:BOX58 - 9 (:BOX61 * 4 2 :SELECTED))))
 3 (OPEN-DIALOG 5495807 ARG :ARG58)
 4 (INPUT-OPEN 5495809)
 5 (TYPED-INPUT 5496765 #\8)
 6 (INPUT-TYPE 5497293 SUBMIT)
 7 (INPUT 5497295 8 "8" (:BOX58 (9 - FILL) (:ARG2-EDITABLE)))
 8 (SUBMIT 5497299 NIL (:BOX56 (:BOX58 - 9 8)))
 9 (SELECT 5497999 (:BOX58 (9 - 8) (:SELECTED)))
10 (OPERATION 5499472 EVALUATE SUCCESSFUL CLICKED (:BOX56 (:BOX58 - 9 8 :SELECTED)))
11 (OPEN-DIALOG 5500878 ARG :ARG55)
12 (INPUT-OPEN 5500880)
13 (TYPED-INPUT 5501532 #\1)
14 (INPUT-TYPE 5501805 SUBMIT)
15 (INPUT 5501807 1 "1" (:BOX56 :ARG55 (:ANSWER-EDITABLE)))
16 (SUBMIT 5501810 NIL (:BOX56 1))
17 (PROBLEM-END 5503912 14 CLICKED)
```

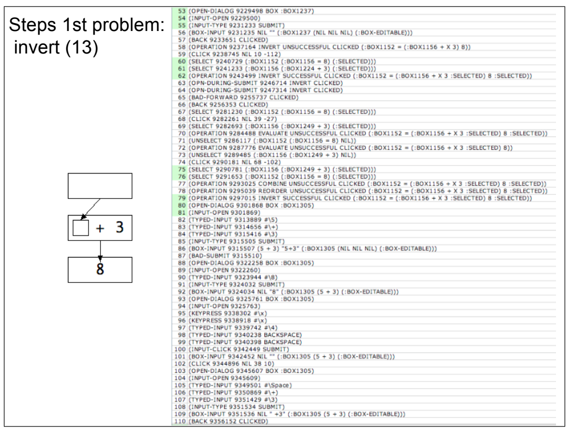
Steps 1st problem: invert (9)

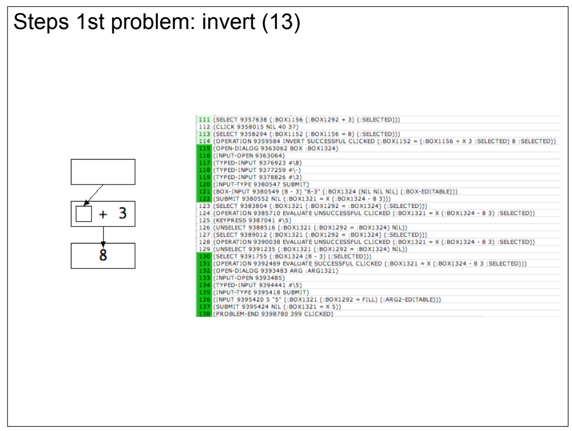


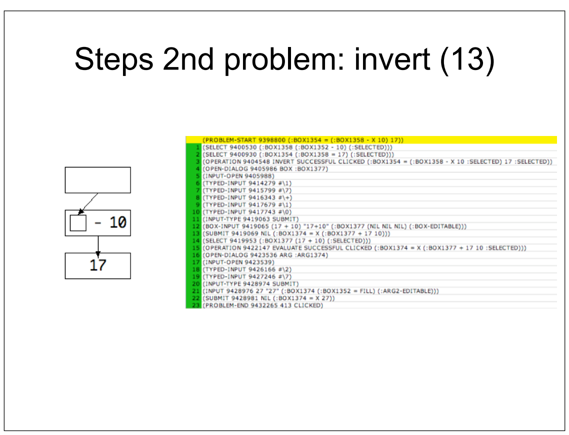

```
   (PROBLEM-START 1766360 (:BOX1253 = (:BOX1257 + X 3) 8))
 1 (READ-INSTRUCTIONS-DONE 1778121)
 2 (CLICK 1779040 NIL 14 -99)
 3 (CLICK 1780232 NIL 38 -109)
 4 (SELECT 1781528 (:BOX1257 (:BOX1252 + 3) (:SELECTED)))
 5 (SELECT 1782192 (:BOX1253 (:BOX1257 = 8) (:SELECTED)))
 6 (CLICK 1782744 NIL 41 -109)
 7 (OPERATION 1784250 INVERT SUCCESSFUL CLICKED (:BOX1253 = (:BOX1257 + X 3 :SELECTED) 8 :SELECTED))
 8 (OPEN-DIALOG 1800990 BOX :BOX1276)
 9 (INPUT-OPEN 1801237)
10 (INPUT-CLICK 1815509 SUBMIT)
11 (BOX-INPUT 1816003 NIL "" (:BOX1276 (NIL NIL NIL) (:BOX-EDITABLE)))
12 (CLICK 1816764 NIL 169 47)
13 (CLICK 1820556 NIL 191 -89)
14 (BACK 1822180 CLICKED)
15 (SELECT 1835922 (:BOX1253 (:BOX1257 = 8) (:SELECTED)))
16 (SELECT 1837762 (:BOX1257 (:BOX1263 + 3) (:SELECTED)))
17 (CLICK 1838226 NIL 56 -101)
18 (CLICK 1838522 NIL 56 -101)
19 (CLICK 1838714 NIL 56 -101)
20 (UNSELECT 1839618 (:BOX1257 (:BOX1263 + 3) NIL))
21 (SELECT 1841873 (:BOX1257 (:BOX1263 + 3) (:SELECTED)))
22 (OPERATION 1843277 COMBINE UNSUCCESSFUL CLICKED (:BOX1253 = (:BOX1257 + X 3 :SELECTED) 8 :SELECTED))
23 (OPERATION 1847579 REORDER UNSUCCESSFUL CLICKED (:BOX1253 = (:BOX1257 + X 3 :SELECTED) 8 :SELECTED))
24 (OPERATION 1851171 INVERT SUCCESSFUL CLICKED (:BOX1253 = (:BOX1257 + X 3 :SELECTED) 8 :SELECTED))
25 (OPEN-DIALOG 1876230 BOX :BOX1307)
26 (INPUT-OPEN 1876494)
27 (TYPED-INPUT 1877750 #\8)
28 (TYPED-INPUT 1878222 #\-)
29 (TYPED-INPUT 1879101 #\3)
30 (INPUT-CLICK 1879878 SUBMIT)
31 (BOX-INPUT 1879886 (8 - 3) "8-3" (:BOX1307 (NIL NIL NIL) (:BOX-EDITABLE)))
32 (SUBMIT 1879890 NIL (:BOX1304 = X (:BOX1307 - 8 3)))
33 (INCORRECT-ANSWER 1882526 378 CLICKED)
34 (CLICK 1884749 NIL 163 11)
35 (SELECT 1889308 (:BOX1307 (8 - 3) (:SELECTED)))
36 (CLICK 1889972 NIL 48 -52)
37 (CLICK 1890452 NIL 48 -1)
38 (SELECT 1890820 (:BOX1304 (:BOX1263 = :BOX1307) (:SELECTED)))
39 (UNSELECT 1891556 (:BOX1307 (8 - 3) NIL))
40 (UNSELECT 1892188 (:BOX1304 (:BOX1263 = :BOX1307) NIL))
41 (SELECT 1892524 (:BOX1307 (8 - 3) (:SELECTED)))
42 (OPERATION 1893253 EVALUATE SUCCESSFUL CLICKED (:BOX1304 = X (:BOX1307 - 8 3 :SELECTED)))
43 (OPEN-DIALOG 1893899 ARG :ARG1304)
44 (INPUT-OPEN 1893902)
45 (TYPED-INPUT 1894764 #\5)
46 (INPUT-CLICK 1895381 SUBMIT)
47 (INPUT 1895392 5 "5" (:BOX1304 (:BOX1263 = FILL) (:ARG2-EDITABLE)))
48 (SUBMIT 1895398 NIL (:BOX1304 = X 5))
49 (PROBLEM-END 1896964 378 CLICKED)
```

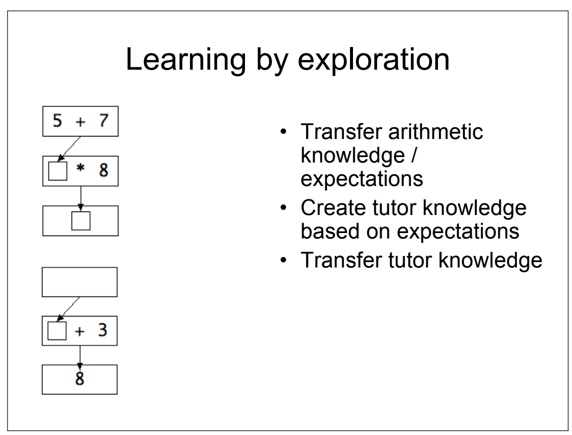

Steps 2nd problem: invert (9)
- 10
17


Steps 1st problem: invert (13)
+ 3
8


Steps 1st problem: invert (13)
+ 3
8


Steps 1st problem: invert (13)
+ 3
8


Steps 2nd problem: invert (13)
- 10
17


Learning by exploration
5 + 7
* 8
+ 3
8
• Transfer arithmetic knowledge / expectations
• Create tutor knowledge based on expectations
• Transfer tutor knowledge

…and the model?

- What it could do…
  - – Cued retrieval: “Stroop” like calculating
  - – Searching the display for finding what to do next
- What it cannot in the moment…
  - – Hierarchical structure of goals
  - – Mapping math knowledge to tutor functions

…and the model?

- Out of reach in the moment…
  - – Acting by expectations
  - – Systematically try what to do next
  - – Repeat for learning

Memory Structures as User Models

Leendert van Maanen (leendert@ai.rug.nl)
Jelmer Borst (jpborst@ai.rug.nl)
Chris Janssen (C.Janssen@ai.rug.nl)
Hedderik van Rijn (D.H.Van.Rijn@rug.nl)
Artificial Intelligence, University of Groningen
Grote Kruisstraat 2/1
9712 TS Groningen, The Netherlands

Introduction

The role of information increases. Both for individuals as for society as a whole, handling information has become a tremendously important aspect of daily live. Simultaneously, the amount of available information increases as well. Given this current information overload (Brusilovsky & Tasso, 2004), research into personalization and recommender systems seems necessary. Applications that limit the amount of information presented to a user by selecting only relevant information would be extremely useful.

Relevant information could be filtered by creating a personal profile of a user, and subsequently selecting information that fits the constraints of that profile. We refer to such a profile as a user model (Brusilovsky & Tasso, 2004). The user model could be explicitly created by presenting a user with a questionnaire on her interests, and using the answers to the questionnaire as a model of that user's interests. A drawback of this approach is that it takes time for a questionnaire to be completed, and the user is thus presented with even more information than before. In addition, in many situations users find it hard to explicate their interests, or their interests may change over time, making it hard to infer their interest using a questionnaire. Therefore, implicit inference of user interests should be applied, for instance using eye movements (Van Maanen *et al.*, 2006) or mouse clicks (Claypool *et al.*, 2001).

ACT-R's declarative memory structure might prove useful for maintaining these personal profiles. ACT-R proposes that chunks in declarative memory are characterized by activation, a quantity that reflects how likely it is that a chunk will be needed in the immediate future (Anderson & Schooler, 1991). The level of activation depends on the history of usage of a chunk (base-level activation), and a component reflecting the influence of the current context on a chunk's activation (spreading activation, Anderson & Lebiere, 1998). The spreading activation component is a weighed sum of the activation of associated chunks, with the weights being the strengths of association. The chunks in ACT-R's declarative memory module form a semantic network structure, in which the edges represent spreading activation between chunks.

The strengths of association can be determined by looking at the frequency of co-occurrences of chunks. If two words frequently co-occur, the presence of one word can be regarded as a predictor for the presence of the other word. However, if a word co-occurs with many different words (such as for instance determiners), than the predictive value of that word is less (Posterior Strength Equation, Anderson & Lebiere, 1998).

These strengths of association may also reflect individual interests. As an example, consider the case of a sports fan reading the newspaper: For her, reading a newspaper will usually involve reading the sports section. Therefore, chunks representing sports related notions and chunks representing the newspaper co-occur more frequently for a sports fan than for a non-sports fan. In ACT-R, a higher strength of association would thus be created between newspaper chunks and sports related chunks for sports fans than for non-sports fans.

Image Recommender System

This feature of ACT-R's associative strength learning mechanism can be exploited to create personalized applications. Searching images on the internet is a typical domain in which personalization is useful, because image search based on one key word generally results in very diverse search results. For instance, searching for the key word *apple* results in images of fruit and images of computers, and searching for the key word *mouse* results in images of rodents or images of computer equipment. Using ACT-R's declarative memory structure, we have developed a recommender system that expands search queries for image search. The Image Recommender System functions as follows.

The user can issue a query to an online image search engine (we used Yahoo! Search SDK), which returns a series of images. By clicking on an image, the user can indicate interest in that particular image. Each time the user indicates interest in an image, the website that contains the image is parsed, and the words are harvested. The assumption is that the words on the websites visited by a user represent not only the content of the websites, but are also indicative of the content of the images on these websites.

Because the user only (or at least generally) visits websites that are of interest to her, the words on these websites also reflect her interests. Spreading activation between these words is calculated using the Posterior Strength Equation. To reduce the computational load, high-frequent words in the semantic network are excluded. These words will probably not influence the recommendations that the system will give, because they likely co-occur with many other words, resulting in low spreading activation. The words that are excluded are for instance determiners or pronouns. Also to reduce the computational load, only the ten most frequent words on a webpage plus the search query are used to calculate strengths of activation, because, low frequent words on a webpage are less indicative of the contents of a webpage than higher frequent words. Again, these words would have low spreading activation. It should be noted that there is no principled reason for these implementation choices, but are only intended to make the size of the semantic network incorporated in the Image Recommender System feasible.

Every new query triggers a retrieval from declarative memory and provides an opportunity to train the strengths of association. The query is stored as a goal chunk, which, being in the focus of attention, spreads activation towards all associated chunks. Given the individualized strengths of association, different chunks might be retrieved for individual users: The chunk with the highest activation will be retrieved, which differs for individual users. The retrieved chunk is the chunk that is the most associated with the goal chunk (search query). That is, the word represented by that chunk occurred most frequently in the context of the query key word. Since the frequency of co-occurrence is determined by the mouse clicking behavior of the user, the retrieved chunk also represents the most likely notion of interest for the current user in the current context. The retrieved chunk is used to expand the search query. In another application, it could be involved in some other personalized task component

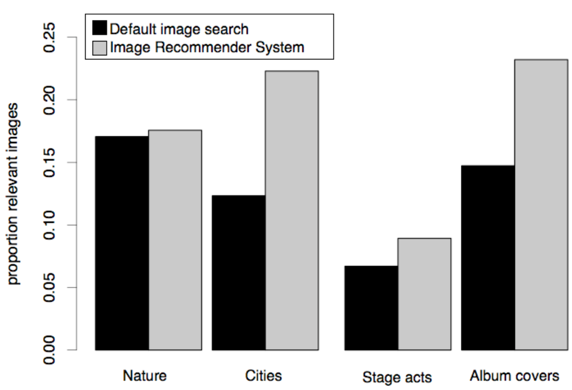


Figure 1. Proportion of relevant images returned by the standard search engine (Without Association) and returned by the Image Recommender System (With Association), for four different image categories and two different query sets.

The Image Recommender System was tested in two experiments. In the first experiment, we performed a series of searches and counted the number of relevant hits, with and without expanding the query. We performed searches for images of 38 European countries, and selected images from a specific category. In one condition, we only selected images that depicted natural scenes, whereas in a second condition, we only selected images that depicted cities. Semantic networks were formed based on these selections, and afterwards we searched for the same 38 European country names, but this time with expanding the query using the Image Recommender System. Searches were performed with queries that were expanded with one of the two most associated items. We did the same experiment with image searches for 14 pop band names, and selected images representing stage acts of these bands and album covers, respectively. We found that in all categories recommending a related key word based on the declarative memory user model increased the number of relevant images, as is depicted in Figure 1.

In the second experiment, we searched for the same word using two different semantic networks. We used the *Nature* and *Cities* networks for this test. Figure 2 shows the image results for the search query *picture*. As can be seen, the recommender system based on the *Nature* semantic network gives different results than the recommender system based on the *Cities* network. The *Nature* recommender system suggested the terms *Lofoten*, an archipelago near the Norwegian coast, and *Reine*, a small fishing village on one of the Lofoten islands. The *Cities* recommender system suggested *Nicosia* and *Nuernberg*, two European cities. Similar results were obtained for the key words *view*, *photo*, *country*, and *time*.

Because during the training of the semantic networks European countries were used as queries, it is not surprising that all recommended terms relate to Europe. However, because of the specific choices made when training the *Nature* and *Cities* semantic networks, the Image Recommender System, expands new queries differently for the users modeled by these declarative memory structures.

Discussion

An issue in our tests is the relatively small size of the declarative memories. Because the initial period in which the network of associations was trained was relatively short, the network size never exceeded 8,000 unique entries and no more than 30,000 words were parsed. Therefore, the system has not reached a stable configuration in which always appropriate recommendations can be made. It could be that some words are strongly associated, because at the web sites visited these words co-occur, although these web sites are not representative of the normal contexts of these

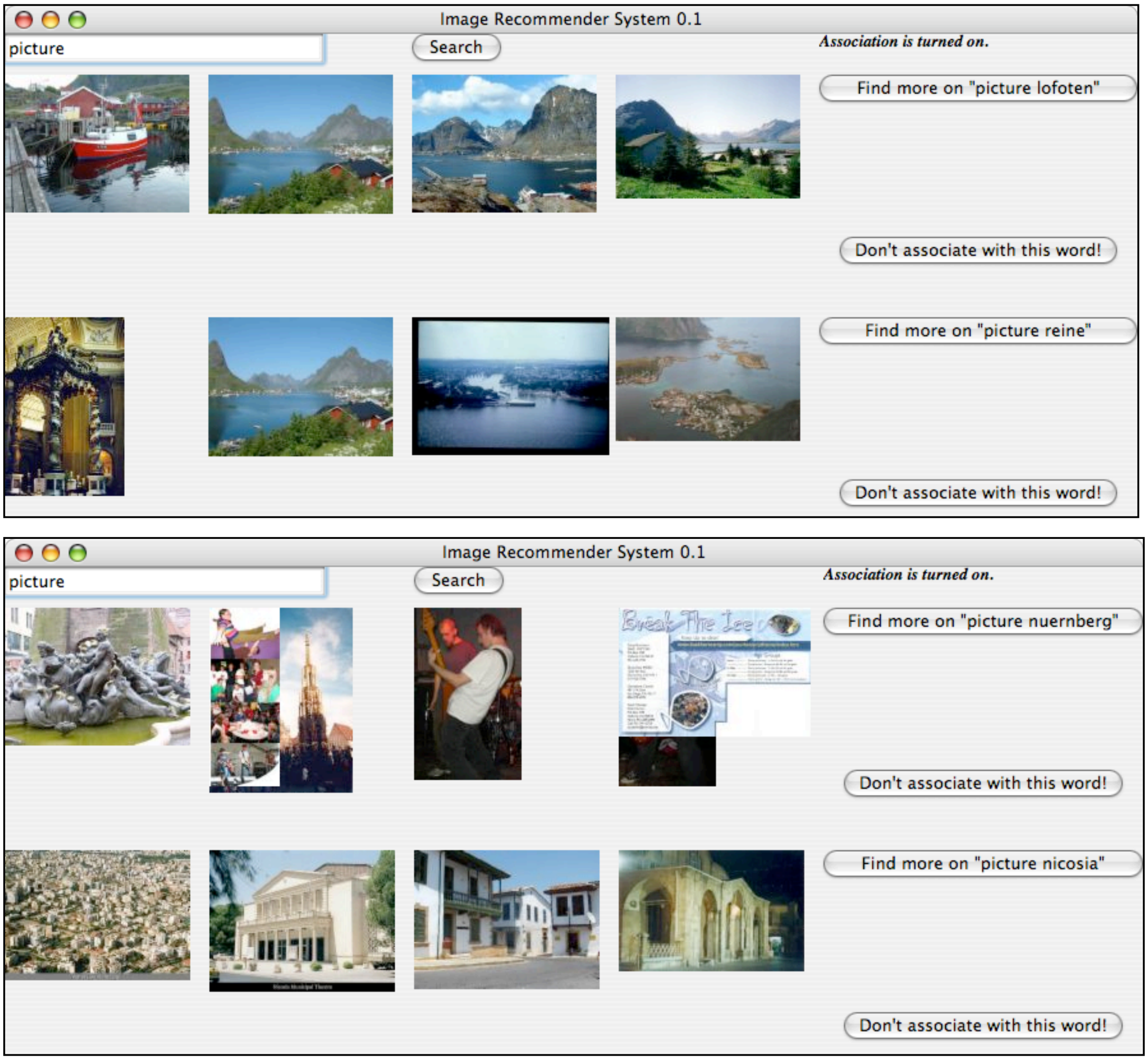


Figure 2. Image search for the key word *picture* using the *Nature* semantic network (top) and the *Cities* semantic network (bottom). The recommender system that uses the *Nature* network expands the query with the key words *Lofoten* and *Reine*, and mainly finds images with natural scenes. The recommender system that uses the *Cities* network expands the query with the key words *Nuernberg* and *Nicosia*, finding images of buildings.

words. In those cases, inappropriate recommendations will be made.

In addition, because of the limited network size, some words that are highly frequent will not be eliminated, but instead will be used for expanding the query. We expect that these issues will resolve if a larger training period is allowed.

In the Image Recommender System we developed, we only relied on strength of association for recommending possibly interesting chunks. The strengths of association can be regarded as reflecting the user's long-term interests, because the strengths of association only change slowly. The short-term interests of a user might be incorporated by including the base-level activation into the equation. If a chunk is recently attended, for instance because the word represented by that chunk has recently been used in a search query, the base-level activation of that chunk has been increased. An increased base-level activation means that the likelihood of being retrieved has also been increased. In this enhanced Image Recommender System, retrieval of the chunk will depend on the strengths of association – based on the long-term interests of the user – and on the base-level activation – reflecting the short-term interests of the user.

Conclusion

A dynamically updated declarative memory structure, consisting of a semantic network of chunks connected by strengths of association, might serve as a model of interest of an individual user. This model subsequently can be used to limit the amount of information presented to a user to a relevant subset. A typical domain of application is (of course) web search, but all situations that involve high information load (Brusilovsky & Tasso, 2004) might benefit from applying ACT-R's declarative memory principles to personalization research.

References

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

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

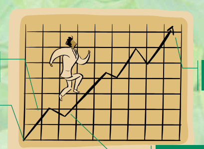
Brusilovsky, P., & Tasso, C. (2004). Special issue on user modeling for web information retrieval - preface. *User Modeling and User-Adapted Interaction, 14*(2-3), 147-157.

Claypool, M., Brown, D., Le, P., & Waseda, M. (2001). Inferring user interest. *IEEE Internet Computing, 5*(6), 32-39.

Van Maanen, L., Janssen, C., & Van Rijn, H. (2006). *Personalization of a virtual museum guide using eye-gaze.* Paper presented at the CogSci 2006, Vancouver, BC.

Yahoo! Search SDK. [software]: Yahoo! Inc.

Simulation & Training Study: Learning and Forgetting in ACT-R



Thesis topic proposed

2005 ACT-R Summer School

Model of learning & forgetting

Building the model

Jong W. Kim, Frank E. Ritter & Richard J. Koubek
The Pennsylvania State University
University Park, PA

Training Issues

- In Advanced Manufacturing Environments
  - Humans still remain as an important element (Mital, 1995).
  - Workers need to acquire various levels of knowledge and skills (Mital, 1997).
  - Task complexity affects the variance of individual learning and forgetting rates (Nembhard & Osothsilp, 2002).
  - A worker's knowledge structure of plastic intrusion machine operations
    - Training can manipulate the knowledge structure and performance (Koubek, Clarkston, & Calvez, 1994).
- In Homeland Security: Counter-terrorism Training Drill
  - In the Washington Post report (Johnson, 2003)
  - Tested the readiness of the first responders under simulated explosions of dirty bombs in Seattle and the release of B/A in Chicago
  - The first large scale training drill in USA for 5-day
  - 8,500 people included from federal, state, and local agencies
  - Expenditure: 16 million dollars
  - **How long will these lessons be retained? How can they be retained?**

Knowledge and Skills are Degraded by "Disuse" or "Infrequent Use"

- Performance can be degraded by
  - Infrequent use
  - Disuse
  - Psychological factors: stress, time-critical situations
- Example: Cardiopulmonary Resuscitation (CPR) Task
  - Non-medical trainees may need to do the task in a space flight mission (Ramos et al., 1995).
  - CPR skill retention study (McKenna & Glendon, 1985)
    - 120 occupational first responders
    - Six months after training, less than a quarter of them were skillful.
- Skill decay will be the cause of human performance decrements !

Issues

- Role of Training
  - Preparedness & Readiness
- We need qualified human performance with expeditious responses, minimized cost of training, making skills more robust against decay
- Using Knowledge & Skills
  - Disuse or infrequent use of knowledge and skills can aggravate human performance.
  - There is a great interest in mission-critical domains to minimize the decay of expert skills after a period of disuse (Chong, 2004).
  - Forgetting: a general human characteristic.
- Different Decay Performance of Knowledge and Skills in a Task
  - Perceptual-motor task: riding a bicycle - generally rarely forgotten
  - Procedural task: CPR task - susceptible to decay
    - An ordered sequence of steps to achieve a goal

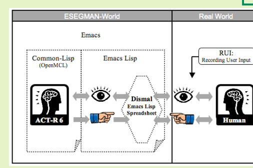
Research Objectives and Approaches

- The Objectives
  - Investigate training performance and proficiency based on the cognitive modeling of procedural skill decay
  - Explore a training paradigm that might mitigate skill decay
- Approach
  - Implement a computational cognitive model of learning and forgetting in the ACT-R cognitive architecture
  - Propose the mechanism of skill decay for modeling
  - Explore mitigating factors against skill decay based on the model
  - Instantiate strategic training research program to mitigate skill decay

07/21/06 2006 ACT-R Workshop 5

The ESEGMAN

- Limitation of the Architecture
  - A cognitive model's restriction to access real task environments
  - How to embody a model to interact with tasks?



- ESEGMAN?
  - Emacs SubstratE: Gate toward MAN-made world
  - A substrate interfacing an ACT-R model with a man-made task environment to explore office work
- The ESEGMAN Roles
  - It functions as a hand and eye of the ACT-R model
  - Eye: to get information on the Dismal window and to pass it back to the model
  - Hand: to pass decisions and actions from the model to the Dismal (a set of key presses or mouse clicks) window

07/21/06 2006 ACT-R Workshop 6

Dismal: Spreadsheet Tasks

- Dismal - gather and analyze behavioral data in spreadsheet based on Emacs editor (Ritter & Wood, 2005)
- In this study
  - Two spreadsheet tasks were generated for the target tasks that the model and the human will perform
- Two Reasons of Using Dismal Spreadsheet
  - Spreadsheet tasks
    - The attributes of procedural and cognition demanding skills
    - Measure learning and forgetting effects on procedural skills
    - Novel spreadsheet, so can measure learning and forgetting
  - Running under Emacs editor
    - ACT-R model with ESEGMAN runs under Emacs.
    - Humans can directly perform the same tasks under Emacs.
    - Human behaviors will be recorded by RUI, Recording User Input (Kukreja et al., in press)

07/21/06 2006 ACT-R Workshop 7

ACT-R Forgetting Model

- Two basic parts
  - A set of declarative representations:
    - Using menu bar (File>Open)
    - Using command keys (C-x C-f)
  - A set of domain-general production rules
    - Attend the Dismal spreadsheet screen
    - Press sequences of keys, Move mouse, and Click mouse button
    - Change goals
    - Repeat certain actions
- Learning: production compilation
- Forgetting mechanism need to be newly proposed and added to extend the current ACT-R architecture

07/21/06 2006 ACT-R Workshop 8

Can ACT-R Forget?

- ACT-R forgetting model of Japanese-English paired-associate tasks (Pavlik & Anderson, 2005; Pavlik, in press)
- Based on the activation mechanism
  - Chunk retrieval performance from DM
  - An item in DM receives an increment of strength when it is practiced.
  - The increment decays as a power function of time.
- Limitation
  - This model only focused on the retrieval performance of declarative knowledge
  - It does not address procedural skill decay.

Skills Decay in ACT-R

- Chong (2004) argued that…
  - The existing set of mechanisms from several architectures (EPIC, ACT-R, and Soar) cannot afford modeling of procedural skill decay.
  - Thus, it is necessary to extend the current architecture to address forgetting phenomena.

Architecture	Capability
EPIC	It doesn't provide a rule learning mechanism. This indicates that the architecture is not able to model procedural skill learning.
ACT-R	The architecture's performance is limited to declarative knowledge learning and forgetting.
Soar	As a rule learning mechanism, chunking is used to model learning phenomena but not the decay of skill.

SOURCE: Chong, R. S. (2004). *Architectural explorations for modeling procedural skill decay.* Paper presented at the The Sixth International Conference on Cognitive Modeling, Pittsburg, PA., USA.

Proposal: Mechanism of Skill Decay

- Learning New Productions: "Production compilation"
  - Collapsing two productions into a single one
  - A process of putting declaratively stored knowledge into a procedural form
- "Production degradation?"
  - After the process of production compilation, what should happen to the model to represent skill decay
  - Need to model procedural skill degradation

Toward Modeling Procedural Skill Decay

- Chong (2004): "skill decay" in ACT-R architecture can be realized by the consideration of knowledge availability in production rules.
- This study will propose procedural skill decay mechanism for ACT-R:
  - Cue unavailability can be the architectural representation of forgetting phenomena that is observed in human behavior.

If the condition of a rule is not satisfied by the declarative elements, → Then, the rule will not match and fire.

A cue

∴ Skill decay can be represented by the inability to retrieve or match a rule cue due to insufficient declarative cues in the context of architectural mechanisms.

Directions

- Complete the ESEGMAN system
  - View Dismal buffer
  - Grab and deliver visual information in Dismal buffer to the model
  - Deliver a string to Dismal from the model
- ACT-R performance
  - Modeling of skill decay
  - Type numbers and string
    - "press-key" command appropriate to model two key presses in Emacs (e.g., C-x C-f)
- Run experiments and gather human performance data

References

Chong, R. S. (2004). *Architectural explorations for modeling procedural skill decay.* Paper presented at the The Sixth International Conference on Cognitive Modeling, Pittsburg, PA., USA.

Johnson, G. (May 13, 2003). Mock Explosion Launches Bioterror Drill: Five-day exercise to test readiness, and determine strengths and weaknesses. In the Washington Post, Retrieved from http://www.washingtonpost.com/wp-dyn/articles/A47092-2003May12.html

Koubek, R. J., Clarkston, T. P., & Calvez, V. (1994a). The training of knowledge structures for manufacturing tasks: an empirical study. *Ergonomics, 37*(4), 765-780.

Kukreja, U., Stevenson, W. E., & Ritter, F. E. (in press). RUI: Recording user input from interfaces under Window and Mac OS X. *Behavior and Research Methods, Instruments, & Computers*.

McKenna, S., & Glendon, A. (1985). Occupational first aid training: Decay in cardiopulmonary resuscitation (CPR) skills. Journal of Occupational Psychology, 58, 109-117.

Mital, A. (1995). The role of ergonomics in designing for manufacturability and humans in general in advanced manufacturing technology: Preparing the American workforce for global competition beyond the year 2000. International Journal of Industrial Ergonomics, 15, 129-135.

Mital, A. (1997). What role for humans in computer integrated manufacturing? International Journal of Computer Integrated Manufacturing, 10(1-4), 190-198.

Nembhard, D. A., & Osothsilp, N. (2002). Task complexity effects on between-individual learning/forgetting variability. International Journal of Industrial Ergonomics, 29, 297-306.

Pavlik, P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. Cognitive Science, 29, 559-586.

Pavlik, P. I. (in press). Incremental learning and spacing effects. In F. E. Ritter, J. Nerb, T. O'Shea & E. Lehtinen (Eds.), In Order to Learn: How the order of materials influences learning. New York: Oxford University Press.

Ramos, M. A. G., & Chen, J. J. G. (1994). On the integration of learning and forgetting curves for the control of knowledge and skill acquisition for non-repetitive task training and retraining. International Journal *of Industrial Engineering, 1*(3), 233-242.

Ritter, F. E., & Wood, A. B. (2005). Dismal: A spreadsheet for sequential data analysis and HCI experimentation. *Behavior Research Method, 37*(1), 71-81.

Towards a Constraint Analysis of Human Multitasking

Duncan P. Brumby (Brumby@cs.drexel.edu)
Dario D. Salvucci (Salvucci@cs.drexel.edu)

Department of Computer Science, Drexel University
3141 Chestnut St., Philadelphia, PA 19104 USA

Introduction

When people conduct multiple tasks in tandem, they often interleave the various operators of each task. Just how these basic cognitive, perceptual and motor processes are ordered generally affords a range of possible multitasking strategies. We briefly outline how a cognitive constraint approach can potentially be used to explicitly explore a range of multitasking strategies, within the theorized constraints that operate on the human cognitive architecture. The power of this approach lies in the task description language, which allows higher-level task performance to be constrained by information requirements and resource demands of lower-level tasks. In general, this approach could provide an *a priori* method for identifying possible multitasking strategies.

Consider while you are driving in your car, it is sometimes not too difficult to direct your attention away from the road, in order to complete a secondary task, such as dialing a number on a cell phone. In this example, there are obvious tensions between the two tasks; suspending attention from the primary task of driving for too long a time period might result in a collision, but completing the secondary task in a rapid and timely manner is probably also important. We briefly outline how an approach called Cognitive Constraint Modeling (CCM: Howes et al., 2004), can be used in a multitasking context to identify the optimal points at which to interleave a primary task, such as driving, in order to complete a secondary task, such as dialing a number on a cell phone.

One of the aims of the cognitive modeling community is to provide an account of human performance on complex real-world tasks. Cognitive architectures (e.g., ACT-R: Anderson et al., 2004) allow models to be developed within a unified framework that integrate assumptions about the time course and information processing constraints that operate on the human system.

For multitasking scenarios, like that described above, most previous models have tended to rely on a *customized executive*, which strategically controls the interleaving of the various task operators (see Salvucci, 2005, pp. 458-460). In response, Salvucci (2005) has proposed a *general executive* for controlling multitasking in the ACT-R cognitive architecture. The general executive assumes that control between two or more primary tasks is passed through a queuing mechanism. The queuing mechanism allows for the interleaving of the various operators that make up each primary tasks. In other words, the multitasking general executive provides a domain independent mechanism for integrating separate ACT-R task models.

Salvucci (2005) has applied the multitasking general executive to the problem of integrating the control and monitoring required for driving, with the completion of secondary in-car tasks, such as dialing a cell phone number. The model was able to account for the increase in dialing time required while driving compared to baseline, and also the degraded steering that resulted from the introduction of the secondary dialing task. The multitasking general executive accounted for these results by assuming that a central cognitive bottleneck operates to limit performance, and that cognitive control must be sequentially ceded between the two tasks.

However, a limitation of this approach is that the modeler has to make additional assumptions regarding the possible range of points in a task that control could be ceded. In other words, the precise operators in a task, at which control can be temporarily given up to a secondary task, must be specified by the modeler. This is a problem because performing one or more complex tasks in tandem affords the cognitive architecture a range of possible strategies with which to order the basic cognitive, perceptual and motor processes required for each task. Here, we briefly outline how an alternative approach, called CCM (Howes et al., 2004), might be used explicitly explore a range of possible strategies for multitasking.

Cognitive Constraint Modeling

The CCM (Howes et al., 2004) approach provides a framework for directly reasoning about the optimal bounds on skilled behavior, given the constraints imposed by the task environment, by strategic knowledge, and by the cognitive architecture. The CCM approach relies on a task description language, called Information-Requirements Grammar (IRG). IRG is motivated by the theory that higher-level task performance is constrained by the information requirements and resource demands that operate on lower-level task processes (see, Howes et al., 2005). Predictions in CCM are then made using a Prolog-based tool, called CORE, which expands the task description specified in the IRG to determine an optimal schedule of the start times for each low-level process. Previously, this approach has been used to account for dual task performance limitations in the psychological refractory period (PRP) paradigm (Howes et al., 2004), and more recently has been scaled up to account for more complex tasks (Eng et al., 2006; Howes et al., 2005).

In a multitasking context, this approach allows parallelism between task operators to be easily defined. This is because IRG does not limit the task description to a sequence of operators, but instead allows resource constraints on lower-level cognitive, perceptual and motor

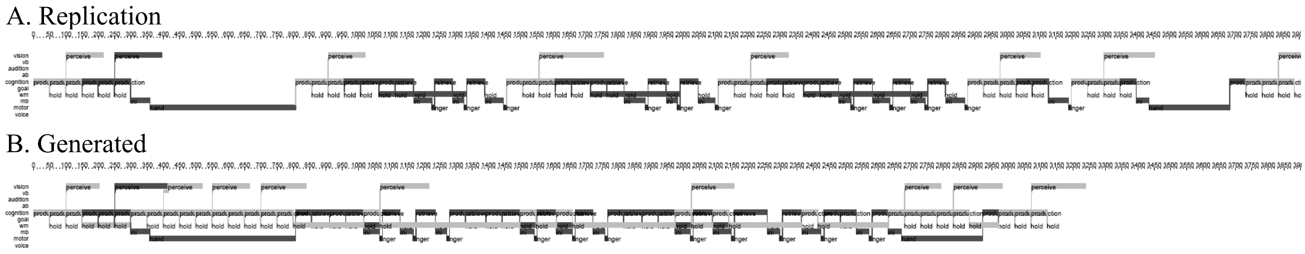


Figure 1. Behavior graphs for dialing a cell phone (dark grey bars) while monitoring a driving task (light grey bars), which (a) replicates Salvucci's (2005) task switching schedule and (b) a greedy schedule generated by CORE that was also consistent with the constraints imposed by the ACT-R cognitive architecture.

processes to determine the sequential orderings of operators. Our explorations of human multitasking performance within a CCM framework is still very much in the early stages of development. Here we present a brief description of some preliminary findings.

Preliminary Results

As a starting point, we reimplemented a model trace from Salvucci's (2005) ACT-R model of driver distraction. As summarized above, this model used a general executive to switch between a primary task (driving) and secondary task (dialing). Figure 1a shows a behavior graph from an IRG description that replicated the original model. In particular, the points at which the ACT-R model could switch between tasks was explicitly represented in the IRG task description. Therefore, this behavior graph is identical to that produced from an ACT-R simulation.

In contrast, Figure 1b removed the explicit task switching points in the IRG and allowed CORE to find a strategy that was consistent with the constraints imposed by the ACT-R cognitive architecture. A greedy scheduling algorithm was used. Comparison of the two outputs suggest that a multitasking strategy could be specified that 1) did more road checks while dialing a cell phone number (7 vs. 5), and 2) could complete the dialing task in less time (3 s vs. 4 s). This difference was partly because the CORE generated schedule exploited slack in the cognitive processor (i.e., the delay between production rule firing) to initiate a new road check, while the dialing task was waiting on the motor processor to execute a key press.

Discussion

We have shown that a CCM approach can potentially be used to directly reason about the space of multitasking strategies afforded within the theorized constraints that operate on the human cognitive architecture. We were able to replicate a previous multitasking model (Salvucci, 2005) by explicitly representing the hypothesized points that control between tasks could be ceded within an IRG task description (Howes et al., 2005). We were also able to use CORE to find a minimal schedule (using a greedy algorithm) that was consistent with the constraints imposed by the ACT-R cognitive architecture and task description. Moreover, this work demonstrates the power of IRG as a task language for describing how the constraints on lower-level cognitive, perceptual and motor processes can determine the sequential orderings of operators, even in the complex case of human multitasking.

Our eventual goal is to identify sets of possible optimal and/or satisficing multitasking schedules. In particular, given the process constraints specified in the ACT-R cognitive architecture, we are interested in identifying a task switching strategy that optimizes the *payoff* between time taken to complete the dialing task and the quality of driver control. In order to specify this payoff function we need to be able to more precisely formalize the quality of driver monitoring, and also the down stream effects of moving attention to a secondary task.

Acknowledgments

Thanks to Andrew Howes for providing access to CORE, and many helpful and insightful suggestions with using the IRG task description language. This research was supported by 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 mind. *Psychological Review, 111,* 1036-1060.

Eng, K., Lewis, R.L., Tollinger, I., Chu, A., Howes, A., & Vera, A. (2006). Generating automated predictions of behavior strategically adapted to specific performance objectives. In the *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'06,* Montreal, Canada.

Howes, A., Vera, A., Lewis, R.L., & McCurdy, M. (2004). Cognitive Constraint Modeling: A formal approach to supporting reasoning about behavior. In the *Proceedings of the 26th Annual Meeting of the Cognitive Science Society,* Chicago.

Howes, A., Lewis, R.L, Vera, A.H., & Richardson, J. (2005). Information-requirements grammar: A theory of the structure of competence for interaction. In the In the *Proceedings of the 27th Annual Meeting of the Cognitive Science Society,* Stresa, Italy.

Salvucci, D.D. (2005). A multitasking general executive for compound continuous tasks. *Cognitive Science, 29,* 457-492.

Exploring Human Multitasking Strategies from a Cognitive Constraint Approach

Duncan Brumby and Dario Salvucci
Department of Computer Science
Drexel University

Thanks to Andrew Howes for support with using CORE

central claims

- multitasking requires the integration of task models
- multitask performance is limited by :-

I. the assumed resource constraints of the human cognitive architecture

e.g., the theoretical commitment to a serial cognitive processor

II. the constraints imposed by the task description

e.g., a cell phone number can only be entered after it has been recalled

the problem of doing more than one thing at once

- people frequently balance performance between two or more continuous tasks

  e.g., they can use a cell phone while driving

- operators for each task can be determined and modeled in a cognitive architecture
- difficulties arise when we model the combined task

the difficulty with modeling human multitasking

- ACT-R, as a theory of the human cognitive architecture, imposes constraints that limit task parallelism

  e.g., the theoretical commitment to a serial cognitive processor

- as a consequence task operators must be interleaved
- but, how do we integrate independent task models?

integrating cognitive models

- integration can be achieved through a process queue (e.g., Halverson, 2006, Liu, Feyen, & Tsimhoni, in press; Salvucci, 2005)

  assume control between two or more tasks is passed through a queuing mechanism, which allows for sequential interleaving between the operators of each task

- the aim of this work was to move away from this concept and instead directly consider the role of resource and task constraints on model integration

towards a constraint analysis of human multitasking

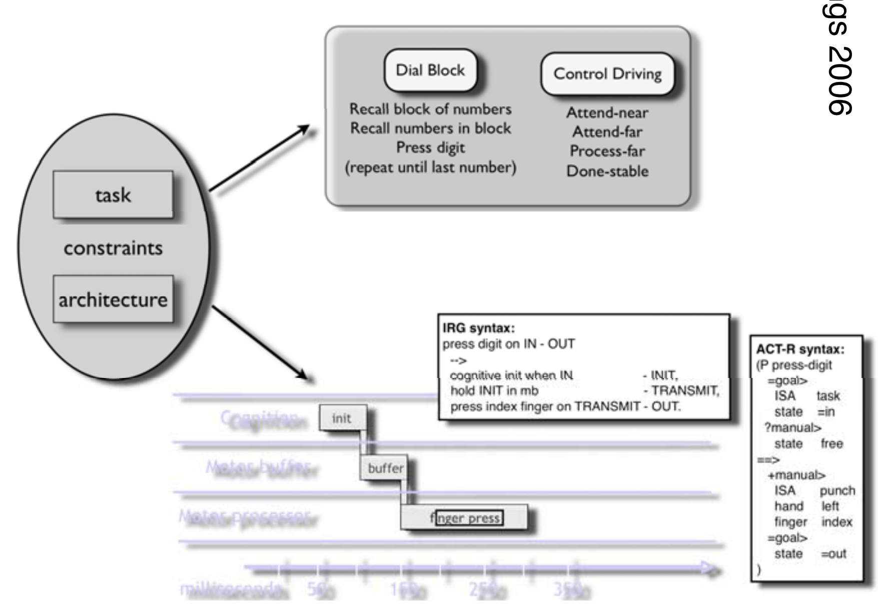
we adopted a *cognitive constraint* approach (CCM) (Howes, Vera, Lewis, & McCurdy, 2004, *CogSci 2004*)

the approach can be used to explicitly explore the space of multitasking strategies afforded by the constraints imposed by the architecture and the task descriptions

overview of CCM

- CCM relies on a hierarchical task description language, called Information-Requirements Grammar (IRG)
- IRG is motivated by the theory that higher-level task performance is constrained by the information requirements and resource demands that operate on lower-level task processes (see, Howes et al., 2005, *CogSci 2005*)
- predictions are made using a Prolog-based tool, called CORE, which expands the task description specified in the IRG to determine an optimal schedule of the start times for each low-level process

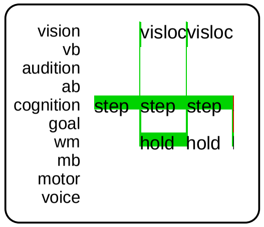
approach: cognitive constraint modeling



simplified driving update

Control Driving

Attend-near
Attend-far
Process-far
Done-stable



Request near visual location

```
(P attend-near
  =goal>
    isa      control
==>
  +visual-location>
    isa      visual-location
    kind     near
)
```

Harvest near visual location
Request far visual location

```
(P process-near-&-attend-far
  =goal>
    isa      control
  =visual-location>
    isa      visual-location
    kind     near
==>
  =goal>
    nloc     =visual-location
  +visual-location>
    isa      visual-location
    kind     far
)
```

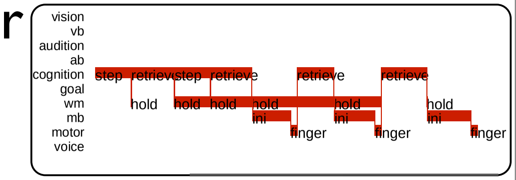
Harvest far visual location

```
(P process-far
  =goal>
    isa      control
    nloc     =near-loc
  =visual-location>
    isa      visual-location
    kind     far
==>
!eval! (do-steer =near-loc =visual-location)
  =goal>
    nloc     nil
)
```

Request near visual location

```
(P done-stable
  =goal>
    isa      control
==>
  +visual-location>
    isa      visual-location
    kind     near
)
```

dialing cell number block



Dial Block

Recall block of numbers
Recall numbers in block
Press digit
(repeat until last number)

Request block

```
(p recall-block
  =goal>
    isa      dial
    person   =name
==>
  +retrieval>
    isa      block
    name     =name
    block    one
)
```

Harvest block
Request first number in block

```
(p recall-first-number
  =goal>
    isa      dial-block
    block    nil
  =retrieval>
    isa      block
==>
  =goal>
    block    =retrieval
    blockpos 1
  +retrieval>
    isa      number
    block    =retrieval
    pos      1
)
```

Harvest number
Press digit
& Request next number in block

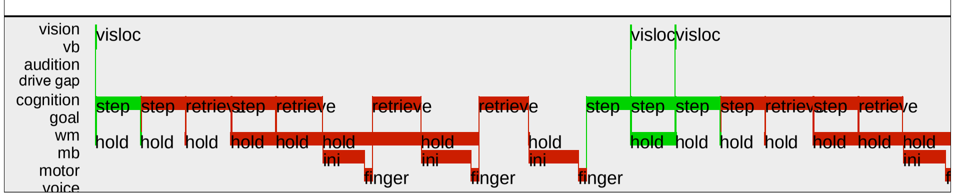
```
(p press-digit-&-recall-next
  =goal>
    isa      dial-block
    block    =block
    blockpos =blockpos
  =retrieval>
    isa      number
    value    =value
  =manual-state>
    isa      module-state
    modality free
!bind! =next (1+ =blockpos)
==>
  =goal>
    blockpos =next
  +manual>
    isa      peck-recoil
    hand     right
    finger   =finger
  +retrieval>
    isa      number
    block    =block
    pos      =next
)
```

dialing while driving model:
I. replication of Salvucci (2005)

the model switched between driving and dialing at particular points using a queue-based scheduler

the assumed switch points were represented in the IRG task description

CORE generated a strategy that replicated the output produced by ACT-R simulation

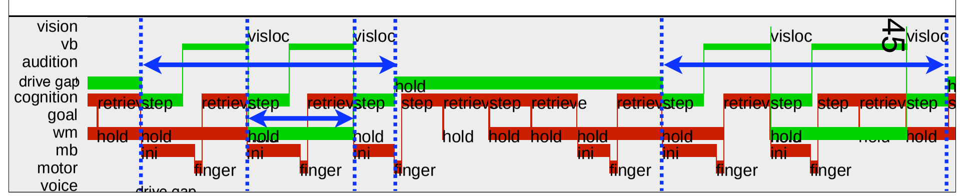


dialing while driving model:
II. greedy scheduler

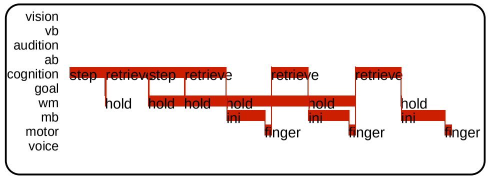
remove the explicit task switching points in the IRG

CORE used to determine strategy (greedy schedule)

rely on task and architectural constraints to limit multitask performance



results



- task parallelism was limited by serial cognitive processor
- the generated schedule exploited slack in the cognitive processor to initiate a new road check, while the dialing task was waiting on the motor processor to execute a key press

compared to a queue-based approach, the CORE model

performed more road checks while dialing (8 vs. 5)

could complete the dialing task in less time while driving (3.10 s vs. 3.75 s)

assumptions, assumptions, assumptions

- is reasonable to assume that driving updates can be separated?
- independence of driving updates

if there is a long delay between updates, then shouldn't the duration of subsequent updates increase?

- alternatively the delay between driving updates can be used to evaluate the performance of a strategy

performance objectives

CORE can be used to systematically explore the full range of strategies

evaluate each strategy by performance objectives

dial fast minimize the total time to dial number (requires an upper limit between driving updates)

drive safe minimize the duration of driving update and delay between driving updates

maximize payoff - find the strategy that maximizes the mutual performance objectives of the tasks

summary

- multitasking requires the integration of task models
- multi-task performance is limited by resource and task constraints
- a cognitive constraint approach can be used to explore the space of strategies afforded by these constraints
- an advantage of this approach is that each strategy does not have to be hand coded

An Integrated Approach to Multitasking in ACT-R

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

Niels A. Taatgen (taatgen@cmu.edu)
Department of Psychology, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213, USA

Human multitasking arises in many real-world situations, from mundane everyday tasks to the most complex, demanding work environments. Cognitive models developed within the framework of cognitive architectures have accounted for multitasking in small-scale (e.g., PRP) tasks and also, to some extent, in complex real-world tasks. However, these models have generally utilized specific multitasking mechanisms to manage component subtasks in their particular domains; as such, these models have "customized executives" (Kieras et al., 2000) that are fine-tuned for the particular task. Other modeling efforts have focused on more general characteristics of domain-independent multitasking for integration of smaller task models into larger multitasking models (see Kieras et al., 2000, for a discussion). For example, Salvucci (2005) has described a general executive for the ACT-R architecture (Anderson et al., 2004), and Taatgen (2005) has explored a general way in which this architecture can account for multiple concurrent tasks.

Our current efforts aim to explore how to integrate the variety of multitasking models and modeling approaches in ACT-R under a single integrated framework. To this end, we have been studying how a simple mechanism can generalize across domains from typical laboratory tasks (e.g., PRP and the "Wickens" tracking task) to complex real-world domains. In particular, we have been exploring a novel approach that allows for constraint-bounded cognition along with additional constraints through use of ACT-R's non-cognitive (e.g., perceptual and motor) modules. In our talk we will describe our new approach and provide a brief overview of the models, their integration with a new temporal module (Taatgen et al., 2005), and their fits to empirical studies of driver behavior (Salvucci, 2001, 2006; Salvucci, Taatgen, & Kushleyeva, 2006).

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

Kieras, D.E., Meyer, D.E., Ballas, J.A., & Lauber, E.J. (2000). Modern computational perspectives on executive mental processes and cognitive control: Where to from here?. In S. Monsell & J. Driver (Eds.), *Control of Cognitive Processes* (pp. 681-712). Cambridge, MA: MIT Press.

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. (2005). A multitasking general executive for compound continuous tasks. *Cognitive Science*, *29*, 457-492.

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

Salvucci, D. D., Taatgen, N. A., & Kushleyeva, Y. (2006). Learning when to switch tasks in a dynamic multitasking environment. In *Proceedings of the Seventh International Conference on Cognitive Modeling* (pp. 268-273). Trieste, Italy: Edizioni Goliardiche.

Taatgen, N.A. (2005). Modeling parallelization and flexibility improvements in skill acquisition: From dual tasks to complex dynamic skills. *Cognitive Science*, *29*, 421-455.

Taatgen, N.A., Anderson, J.R., Dickison, D. & van Rijn, H. (2005). Time interval estimation: Internal clock or attentional mechanism? In *Proceedings of the Twenty-Seventh Annual Meeting of the Cognitive Science Society* (pp. 2122-2127). Mahwah, NJ: Erlbaum.

The Neural Correlates of Control States in Algebra Problem Solving

Andrea Stocco & John R. Anderson
Carnegie Mellon University

Algebra is a complex human activity that requires coordination of several cognitive abilities, including visual processing (for parsing the equation), declarative memory (for storing and retrieving arithmetic knowledge), and visual imagery (for updating and manipulating intermediate and partial representations of the equation). It is also a convenient experimental task, since the solution path can be perfectly characterized, and participants are extensively trained in solving algebraic problems with the same algorithm, repeating the same sequence of problem-solving steps.

We took advantage of this paradigm, as well as previous results with algebraic tasks (Anderson, 2005; Qin et al., 2004), to look for the neural correlates of *control states* in ACT-R. Control states are those slot values in the goal chunk that hold a distinctive hallmark for the current state. They allow us to distinguish the current state from similar buffer configurations, allowing the correct sequence of productions to fire. Together with procedural knowledge, they constitute the main components of top-down control in ACT-R.

In our experiment, participants were required to solve a set of 128 equations. In each of them, the unknown could be unwound in two steps, which consisted of first adding (or subtracting) the same quantity to both sides, and then multiplying (or dividing) both sides by the proper factor. Participants had to correctly indicate these two steps by pressing the corresponding finger in a data glove, and eventually choosing the result from a list of four alternatives. The equations were divided into four categories, obtained by varying two dimensions: whether the equations were *Updated* or not, and whether they contained *Numbers* or *Parameters*. In the Update condition, the software computed the intermediate state and displayed it on the screen. Under these conditions, participants did not have to perform mental manipulations of the equation, and the amount of control was limited to the basic choice of the computational steps to carry on. On the contrary, in the No Update condition, the application did not update the equation on the screen, and participants had to mentally calculate the intermediate states. This increased the number of intermediate problem states participants needed to hold, and, when the equations also contained numbers, it also required the retrieval of arithmetic facts. Crucially, the engagement of each new module should result in the requirement of new control states. Activity due to control states should steadily increase from the two No Update conditions to the Update Parameters, and finally reach a peak with the Update Numbers equations.

The experiment was performed within a 3T MRI scanner, with a relaxation time set to 1.5 seconds (FOV = 20cm, Flip Angle = 73º). A confirmatory analysis was conducted on our preliminary results, using eight predefined regions of interest (ROI) that have been previously mapped onto ACT-R buffers (e.g., Anderson, 2005). Five ROIs seemed to be differentially affected in the four task conditions. The Posterior Parietal area showed an activation pattern that was consistent with the demands of the imaginal buffer, being significantly affected by the number of problem states required to solve an equation, but not by the amount of the retrievals. A region in the left Prefrontal Cortex, on the contrary, seemed to be affected by retrieval of arithmetic facts alone, as predicted by its previously assigned interpretation as the neural correlate of the retrieval buffer. Crucially, three regions (anterior cingulated cortex, fusiform gyrus, and caudate nucleus), exhibited activation patterns compatible with increased control requirements. One of these regions (the left fusiform gyrus) is a visual recognition area, and its activation probably reflects increased visual scanning of equations. The remaining two areas confirm their current interpretation as the goal buffer and the procedural module, and their involvement in top-down control.

References

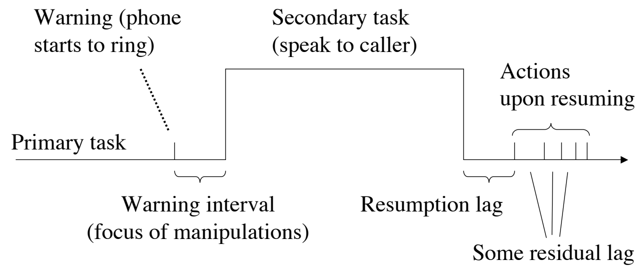
Anderson, J. R. (2005) Human symbol manipulation within an integrated cognitive architecture. Cognitive Science, 29(3), 313-341.

Qin, Y., Carter, C. S., Silk, E., Stenger, V. A., Fissell, K., Goode, A. & Anderson, J.R. (2004). The change of the brain activation patterns as children learn algebra equation solving. *Proceedings of National Academy of Sciences*,*101*, 5686-5691.

Modeling the timecourse of recovery from interruption

Erik M. Altmann
Michigan State University
Greg Trafton
Naval Research Laboratory

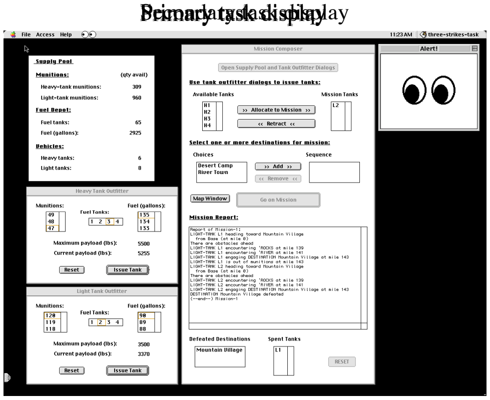
One kind of interruption



Task environment

(Brock & Trafton)

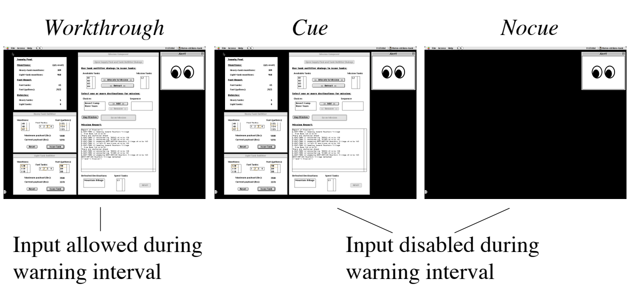
Primary task display



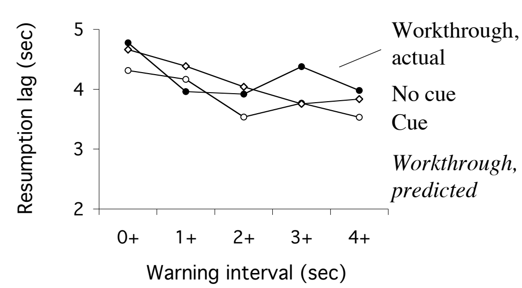
Experiment

- 20 min training, 1 hour of performance
  - 36 interruptions, 35-45 sec each
- Design:
  - 10 (*serial position* after interruption) x
    5 (*warning interval* before interruption, BS) x
    3 (*perception-action* mode of warning interval, BS)
  - 375 subjects, 25 per BS cell

Conditions



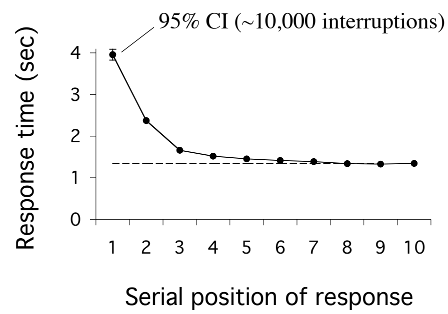
Resumption lag



Resumption lag: Why the floor?

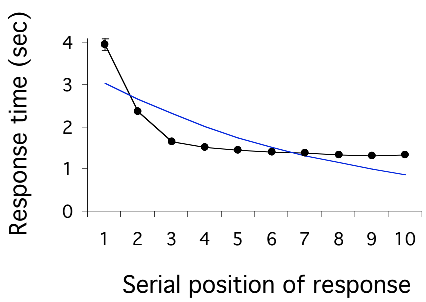
- Interventions
  - Modest effects: Cue availability, warning interval, red-arrow pointer, learning
  - No effects: Notepads (structured on paper, freeform online), cursor as pointer
- A ~2 sec lag over baseline remains
  - Why this floor?
  - Examine *timecourse* of recovery as source of constraint

Timecourse of recovery



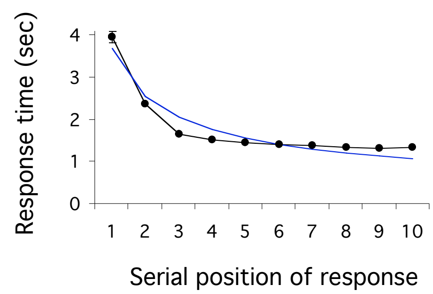
Base-level learning model

Goal loses activation during interruption, then gain it back

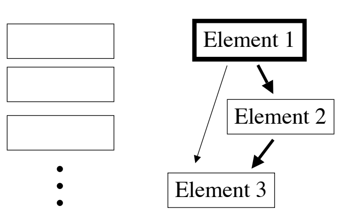


Power-law model

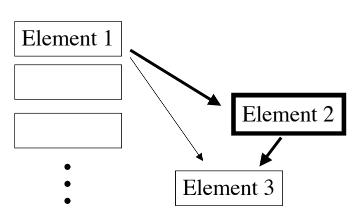
Elements activated during the interruption decay, *or*
Something gets better with practice after an interruption



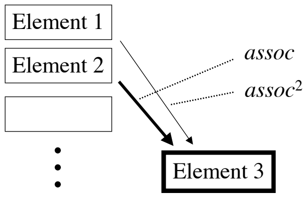
Retrieving elements to the focus



Retrieving elements to the focus



Retrieving elements to the focus

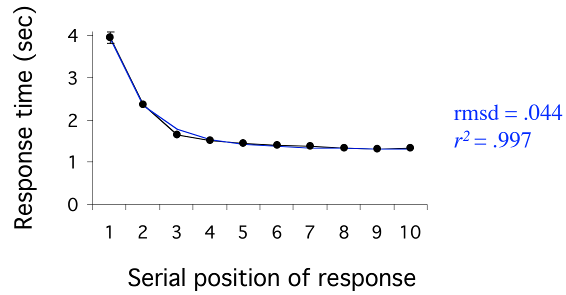


Priming delivered to element $p = -1 + \sum_{i=1}^{p} assoc^{i-1}$

Use this as activation in ACT-R latency model

Primed-retrieval model: $RT(p) = F \exp\left[1 - \sum_{i=1}^{p} assoc^{i-1}\right]$

Primed-retrieval model



Conclusions

- Role for expertise
  - Affects *assoc* parameter
  - Cross-task associations should speed recovery
- A mechanism of *flow*, *situational awareness*
  - A full mental focus that primes the next action
  - Implicit in *priming constraint* (Altmann & Trafton, 2002)

The Roles of Prefrontal and Posterior Parietal Cortices in Algebra Problem Solving: A Case of Using Cognitive Modeling to Inform Neuroimaging Data

Jared Danker

Based on the assumptions of a unified cognitive architecture (ACT-R), we predicted that increasing the retrieval demands of algebra problems would lead to increased activity in prefrontal cortex and increasing the transformational requirements of algebra problems would result in increased activity in posterior parietal cortex. We designed an algebra task that separated the normally correlated processes of transformation and retrieval and manipulated them independently. We found that manipulating either process lead to differential activity in both prefrontal and posterior parietal cortices, as well as several other regions. We propose two explanations for these results. The first is that these two regions do not subserve separate functions as is assumed by ACT-R. The second is that we did not successfully isolate the processes of transformation and retrieval. We rely on cognitive modeling to investigate these two options.

Individual differences in multi-tasking and Control
A Hybrid model of Attentional Blink

Niels Taatgen, Ion Juvina, Seth Herd & David Jilk
Carnegie Mellon University, University of Colorado & eCortex

The hypothesis: individuals differ in their ability to structure control

The central hypothesis in this project is that individual differences in multi tasking can be explained by the way individuals construct their control structure of the task. More specifically, high-proficient individuals construct more elaborate control structures than low-proficient individuals. Finding the right control structure for a task is a matter of striking the right balance. One side of the balance has already been widely recognized: too little control leads to suboptimal task performance, basically corresponding to not properly carrying out the task. What is less well recognized is that on the other side of the balance too much control leads to inflexible and brittle behavior, which I have summarized with the term “Minimal Control Principle”.

Dual-task timing and Abstract Decision Making

The basis for this hypothesis was our work in early 2005 in which we compared individual performance on a multi-tasking paradigm that involved time estimation and responding to multiple visual stimuli (the dual-task timing task, DTT), and the Abstract Decision Task (ADM) developed by Joslyn and Hunt. In this experiment we found a very high correlation between the dual-tasking aspects of the DTT task and the score on the ADM task. Performance of individuals who performed best on the DTT task could be explained best by a model with a four-state control structure, while individuals that performed more poorly on the task were best explained by a three-state control structure. The extra control state enabled the high performers to do the time estimation aspect of the task without being interrupted by other aspects of the task. Although we developed some models that could in principle explain similar differences in the ADM task, these models could not be validated because the experimental software that we obtained from Susan Joslyn didn’t register sufficient details of task performance.

New experiment with Attentional Blink and N-Back

For a new experiment we conducted this year, we reimplemented the ADM task to enable more insight in the choices participants make. In addition to the new ADM task and the DTT task, we gave participants two additional tasks: the N-Back task, which is a working memory task with a high level of cognitive control, and a task to measure Attentional Blink. In the Attentional Blink task participants are presented with a rapid sequence of 20 characters consisting of 18 digits and 2 letters. The task is to pick out the two letters and report them back. When the interval between the two letters is one or two digits, the second letter is often not perceived (more often than when the two letters are consecutive or when they are far apart). This is called the Attentional Blink effect. The reason to include Attentional Blink was that Martens has found that under distraction the blink effect disappears, indicating that more control leads to poorer performance.
The experiment confirmed our expectations: ADM, DTT and N-Back correlated positively with each other, while Attentional Blink had a negative correlation with the

other three tasks. In other words, the participants that performed well on Attentional Blink performed poorly at all the other tasks.

Modeling the individual differences

For a better understanding of the individual differences we are currently finalizing the construction of cognitive models of all four tasks, one low-control model and one high-control model. There are two models for the DTT task: high-control (4 states) where time estimation is "protected", and low-control (3 states), where time estimation can be interrupted.

For the ADM task we constructed a low-control (1 state) model that collects information and tests hypotheses, but without clearly structuring those two aspects of the task. This model turns out to fit the low-performing participants very well. We are currently envisioning two high-control versions, one is a two-state model that strategically switches between gathering information and testing hypotheses, and one model with potentially many control states that implements a full decision tree.

For the N-Back task we have implemented a low-control, one-state model that on every stimulus tries to retrieve an earlier occurrence of that stimulus, and if it finds one tries to judge how long ago that was, and whether this corresponds to the current value of N. This strategy is low-effort, but quite inaccurate. A high-control strategy, which requires two control states, is to retrieve all symbols that were presented between the earlier occurrence and the current symbol and count them.

For the Attentional Blink task the high-control model has two states, one for detecting targets among the distractors, and one for storing them. This means that the model can miss the second target while it stores the first target. The low-control model only has a single state, and will therefore not miss the second target, and exhibits no attentional blink.

	blink	NBack	DTT	ADM
S11	1	1	1	1
S12	1	1	1	1
S28	1	1	1	1
S35	1	1	1	1
S37	1	1	1	1
S6	1	1	1	1
S26	1	1	1	1
S27	1	1	1	1
S32	1	1	1	1
S41	1	1	1	0
S5	1	1	1	0
S7	1	1	1	0
S31	1	1	1	0
S25	1	1	0	1
S38	1	1	0	1
S13	1	0	1	1
S21	1	0	1	1
S23	1	0	1	1
S4	1	0	1	1
S18	1	0	1	1
S24	1	1	0	0
S33	1	1	0	0
S16	1	0	1	0
S9	1	0	1	0
S2	1	0	0	1
S39	0	1	0	1
S17	1	0	0	0
S29	1	0	0	0
S30	1	0	0	0
S34	1	0	0	0
S19	0	0	0	1
S1	0	0	0	0
S10	0	0	0	0
S14	0	0	0	0
S15	0	0	0	0
S22	0	0	0	0
S3	0	0	0	0

The table above shows all the participants in the experiment with a preliminary classification for each of the four tasks. If their behavior had a better fit with the high-control model, it is classified as a 1 (grey), when it fits the low-control model better it is classified as a 0 (white). 15 out of 37 participants have either all 1's or all 0's, or 40.5%, which is much more than chance (which is 12.5%).

A Hybrid model of Attentional Blink

Individual differences in multi-tasking and Control

Niels Taatgen, Ion Juvina, Seth Herd & David Jilk

Carnegie Mellon University, University of Colorado & eCortex

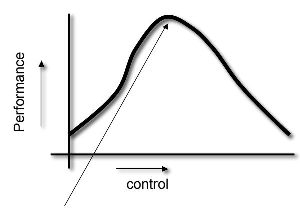
Cognitive Control

- Allows us to keep track of what we are doing
- Allows us to handle multi-tasking
- Allows us to deal with switches in the task
- Allows us to deal with interruptions

Finding the proper control structure is a balancing act

- Not enough control
  - Suboptimal task performance
- Too much control
  - Brittle behavior
  - Cannot handle unexpected events
  - Cannot cope with missing knowledge

Individual differences in Control

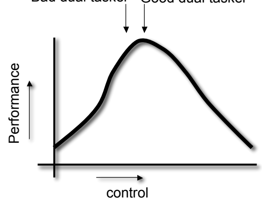


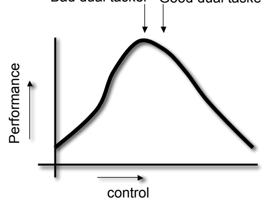
Too little control: Not able to fully perform the task

Too much control: Brittleness

Hypothesis: individuals differ in their ability to find the optimal amount of control

Individual differences in Control



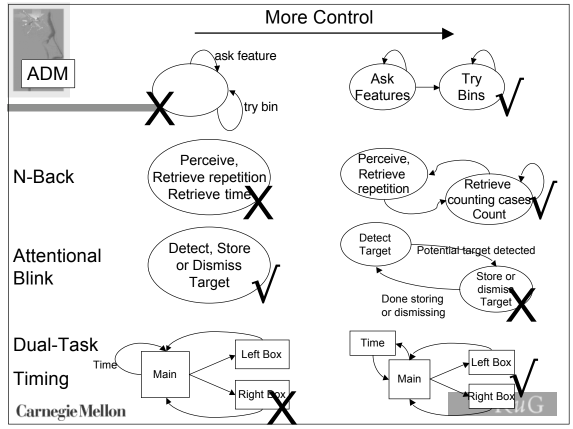


Experiment with four tasks

- For each task we will have a low and a high control version
- Participants' performance will be matched against each of these models
- Expectation: if an individual's behavior on a task corresponds to the high control model, it will also do so for other tasks

The Tasks

- Dual-task Timing (we will only show results)
- Attentional Blink (this talk)
- Abstract Decision Making (Daniel)
- N-Back (Ion)



Attentional Blink

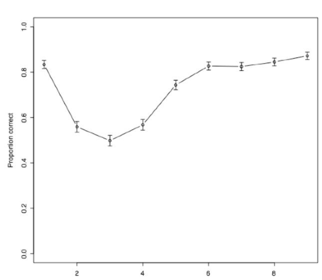
- Stream of 20 characters, task it to spot targets (letters) in between distracters (digits).
- There are either 0, 1, or 2 targets
- If there are 2 targets, there can be 0-8 distracters in between them
- Characters are presented 100 ms apart

Lag

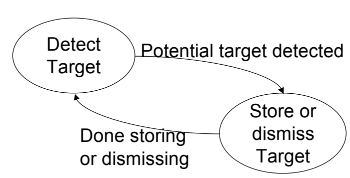
- Lag 1:
  - 49204039GF343432923
- Lag 3:
  - 0230349023Y94D324294
- Lag 9:
  - 9430R32305129K235209

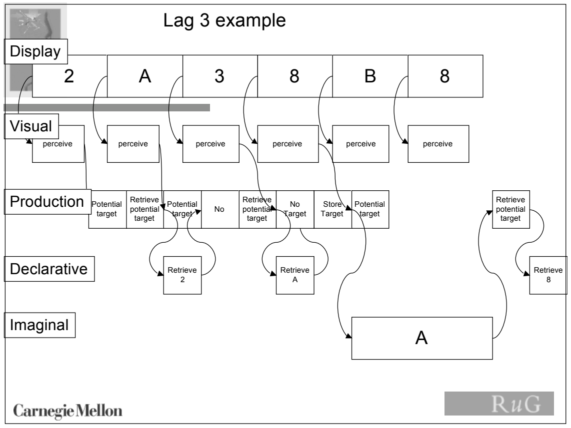
Typical result

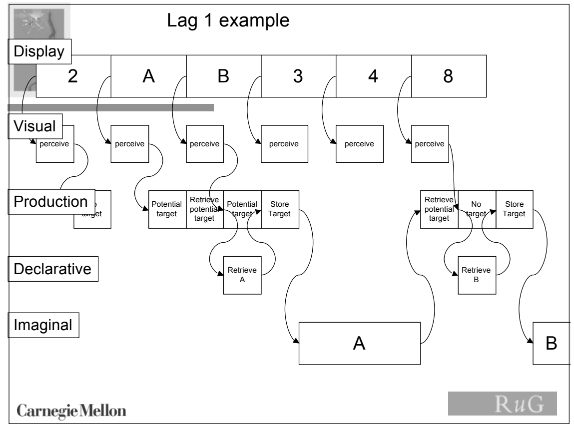
- Proportion where *second* target is reported correctly

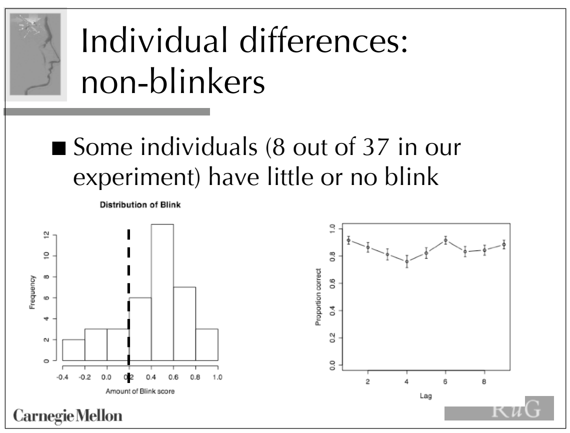


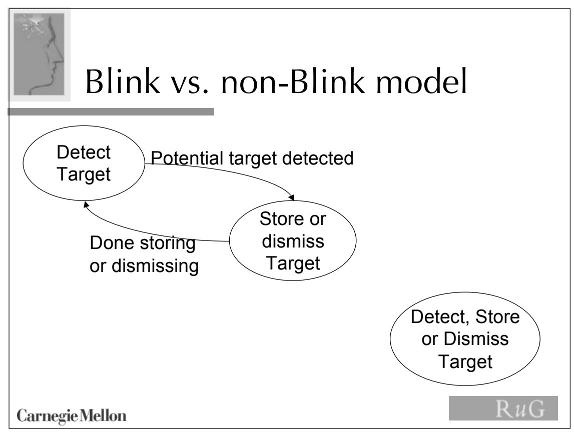
ACT-R model: two states

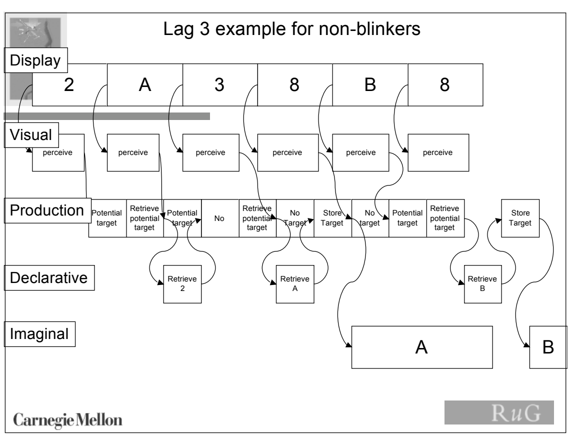


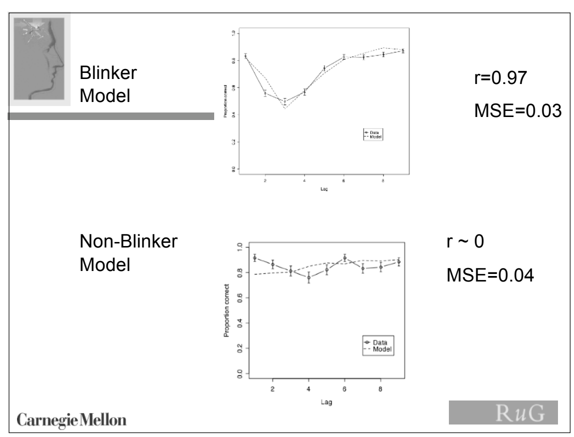

Lag 3 example
Display
2
A
3
8
B
8
Visual
perceive
Production
Potential target
Retrieve potential target
No
No Target
Store Target
Declarative
Retrieve 2
Retrieve A
Retrieve 8
Imaginal
A
Carnegie Mellon
RuG


Lag 1 example
Display
2
A
B
3
4
8
Visual
perceive
Production
Potential target
Retrieve potential target
Store Target
No target
Declarative
Retrieve A
Retrieve B
Imaginal
A
B
Carnegie Mellon
RuG


Individual differences: non-blinkers
■ Some individuals (8 out of 37 in our experiment) have little or no blink
Distribution of Blink
Frequency
Amount of Blink score
Proportion correct
Lag
Carnegie Mellon
RuG

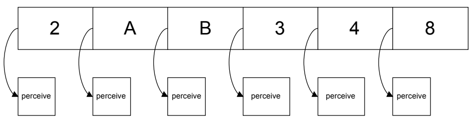

Blink vs. non-Blink model
Detect Target
Potential target detected
Store or dismiss Target
Done storing or dismissing
Detect, Store or Dismiss Target
Carnegie Mellon
RuG


Lag 3 example for non-blinkers
Display
2
A
3
8
B
8
Visual
perceive
Production
Potential target
Retrieve potential target
No
No Target
Store Target
No target
Declarative
Retrieve 2
Retrieve A
Retrieve B
Imaginal
A
B
Carnegie Mellon
RuG


Blinker Model
r=0.97
MSE=0.03
Non-Blinker Model
r ~ 0
MSE=0.04
Proportion correct
Lag
Data
Model
Carnegie Mellon
RuG

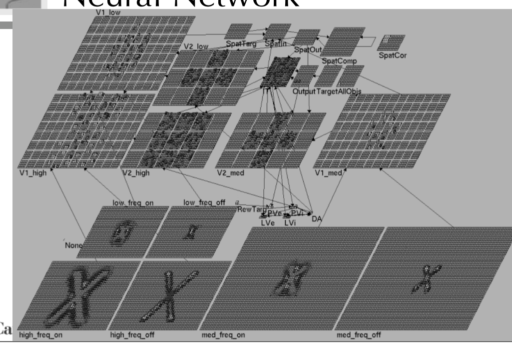
A Hybrid (ACT-R/Leabra) Model

- In the Lag 1 case, participants report the two targets *in the wrong order* in 15% of the cases. This is hard to explain within ACT-R

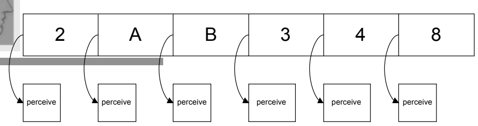


Carnegie Mellon RuG

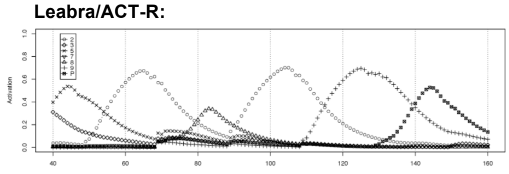
Replace visual module with Neural Network



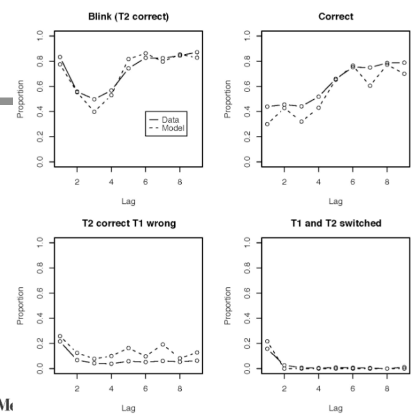
Classic ACT-R:



Leabra/ACT-R:



Carnegie Mellon RuG

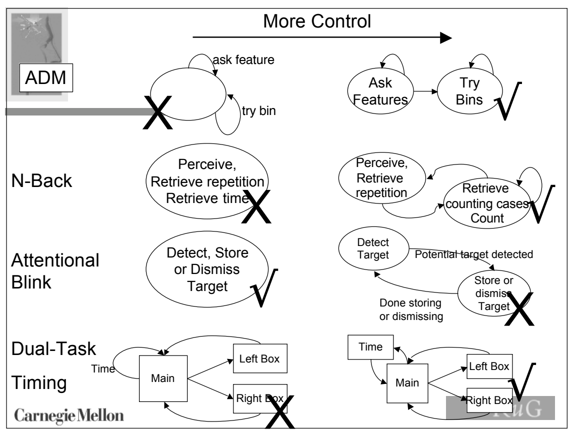


Carnegie Mellon RuG

Conclusions

- One vs. Two controls states can characterize Blinkers vs. non-Blinkers
- One of two successful demonstrations of an ACT-R/Leabra model

Carnegie Mellon RuG



Carnegie Mellon RuG

Individual differences in the abstract decision making task

Daniel Dickison and Niels Taatgen

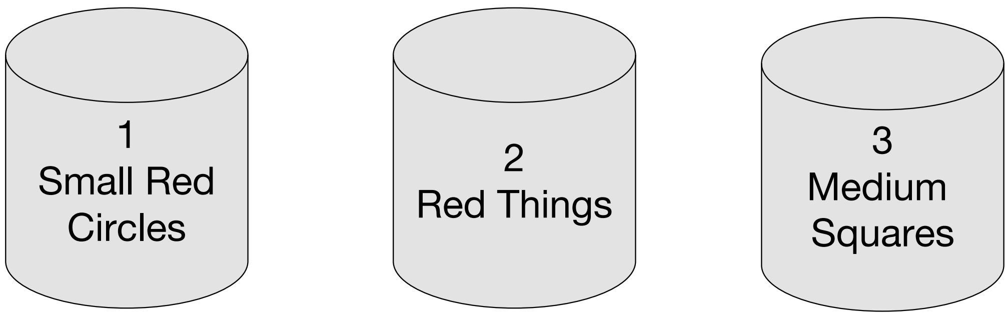
July 2006

Abstract Decision Making (ADM)

- Developed by Joslyn and Hunt to measure the capacity to make decisions under time pressure
- Task is to classify objects into bins by asking properties of the objects and then assigning them
- There is time pressure because new objects come in at a steady pace

Example Bin Preview

- Studied before each game
- Not visible during game, but can be reviewed



Example Trial

- What would you like to do?
  - Ask
- Which object?
  - 8
- Shape, color or size?
  - Color
- “Red”

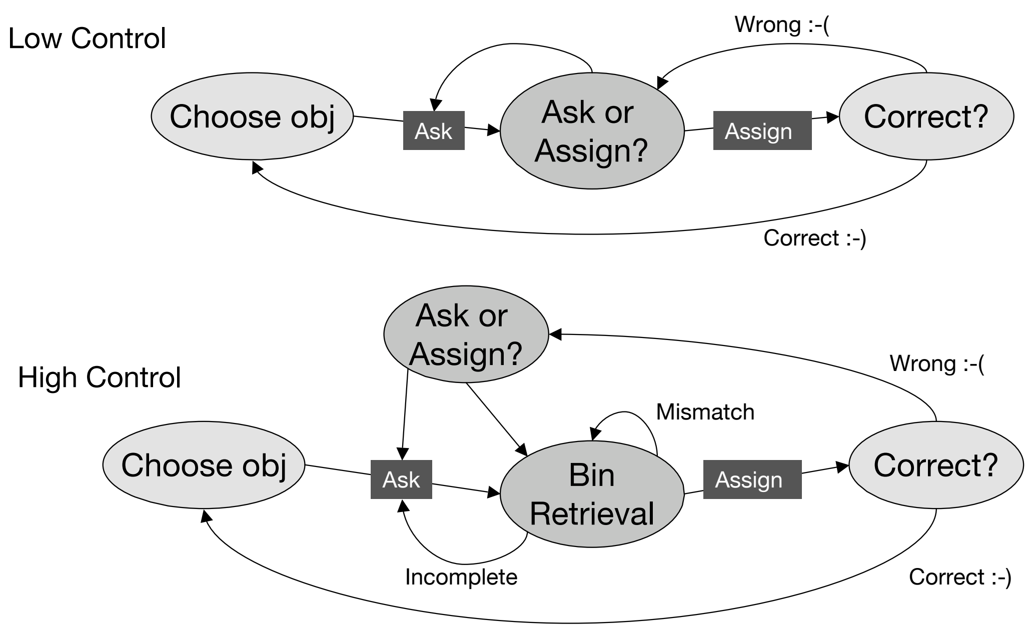
Available objects: (8 9 10)

Example Trial (continued)

- What would you like to do?
  - Assign
- Which object?
  - 8
- Which bin?
  - 2
- "Correct! 2.2 points added."

Available objects: (8 9 10)

Models Overview

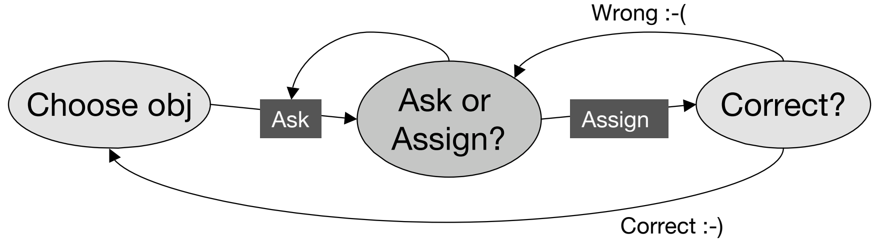


Common features

- “Working copy” of object stored in =imaginal>
- Bins retrieved via spreading activation from =imaginal>
- Text UI interaction via “pseudo” subgoal
  - Type-word slot in goal contains string while typing
  - Type-word is nil after typing, and feedback is encoded in =visual>
- GUI is virtually identical for humans and models

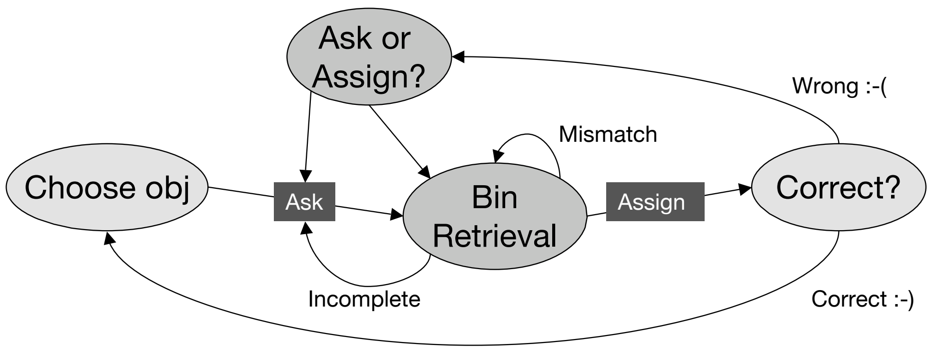
Low Control details

- (New) utility learning for “ask vs assign”
- Rewards proportional to points received
  - More specific bins receive higher rewards
  - Negative rewards prevents over-eagerness.



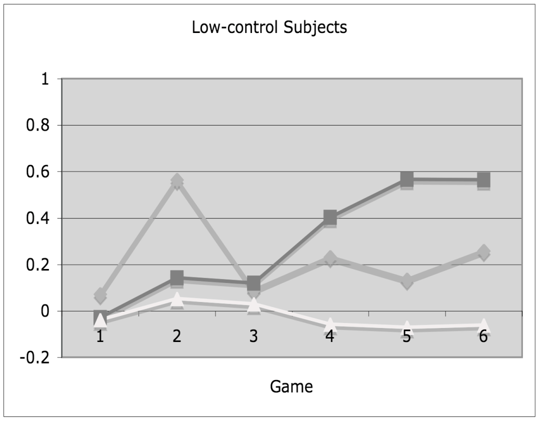
High Control details

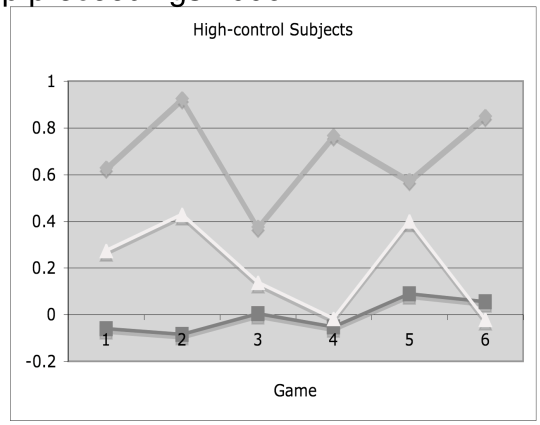
- More explicit control over when to ask vs assign
- Declarative finsts used to narrow down bin choices

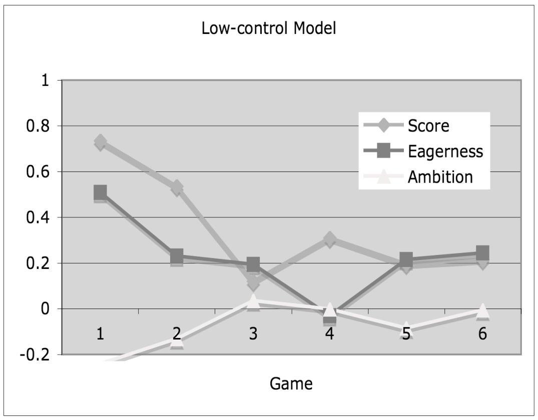


Evaluating Models

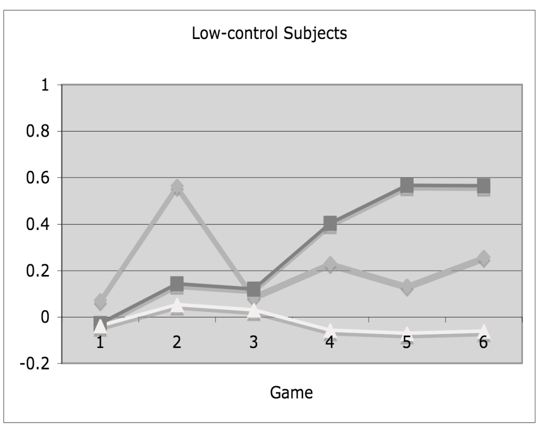
- ADM measures
  - Score: measures overall performance
  - Eagerness: Tendency to assign objects prematurely (i.e. before sufficient info)
  - Ambition: Tendency to ask extra questions in order to get higher points
  - Etc, etc, etc.
- Split subjects by score (high vs low control), then evaluate fit of various measures
- Eventually, group subjects not by score but by similarity to 1 of the 2 models

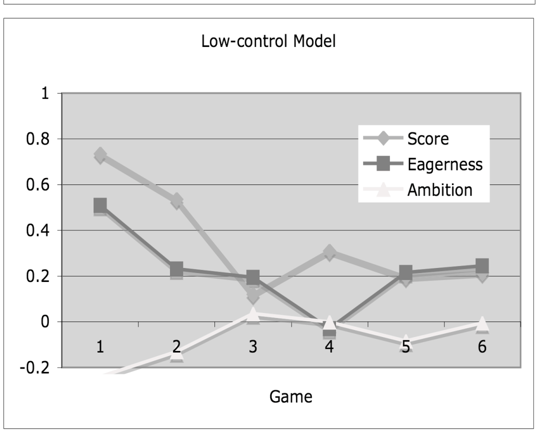






High-control Model
In development





- Subjects get more eager -- probably getting lazy
- Model roughly fits average score and eagerness
- Model learns to be *less* eager
- Experimenting with parameters
  – Reward magnitude
  – Retrieval latency

The Role of Top-Down Control in Working Memory Performance: Implications for Multi-Tasking

Ion Juvina, Niels A. Taatgen, & Daniel Hasumi-Dickison
Carnegie Mellon University

To be presented at ACT-R Workshop, July 2006.

1. Introduction

Previous research in cognitive modeling has suggested that top-down control improves multitasking performance. Specifically, increasing the number of control states maintained by the procedural module in the goal buffer has been shown to increase performance of a cognitive model taking a Dual Task and Timing (DTT) test (Taatgen, 2005). Based on this idea one could hypothesize that maintaining an elaborate control structure – referred hereto as *top-down control* – is one of the sources of individual differences in multitasking.

A correlational study has been conducted to investigate individual differences in multitasking. An Abstract Decision Making (ADM) task (Joslyn & Hunt, 1998) has been used as a dependent measure of multitasking, given previous findings showing a high positive correlation between ADM and DTT (Taatgen, unpublished). ADM requires assigning objects to bins based on their features while handling interruptions and under time pressure. The N-back task (NB) has been used to measure Working Memory (WM) performance. NB requires judging whether an item matches the *n*th-item back (e.g., 1-back, 2-back, 3-back) in a sequentially presented list of items. It challenges participants to maintain a changing stream of stimuli in working memory while comparing them with incoming stimuli. It has been speculated that the NB task places high demands on executive control processes (McElree, 2001). A Rapid Serial Visual Presentation task has been used to measure the Attentional Blink (AB) effect. AB is missing the second out of two targets presented rapidly (10 stimuli per second) in a stream of distractors. Limited cognitive resources are allocated to full processing of the first target, causing the second target to be missed (Martens, Wolters, & van Raamsdonk, 2002). These tasks have been performed by 37 subjects randomly selected from the subjects database of Carnegie Mellon University.

A *top-down control* factor has been postulated to underlie performance in all these tasks. Structural equation modeling (SEM) has been used to structure and analyze the pattern of correlations in the empirical data. Subsequent cognitive modeling activities have been performed to analyze the computational implications of this postulate.

2. Empirical results

A global performance score has been computed for each task by adding points for correct answers and subtracting points for errors. An exception is the score of the AB effect, which has been calculated as the frequency of missing the second target when the first target is correctly reported. Besides these global scores, detailed logs of individual actions of participants have been used in both analysis of empirical data and modeling.

Figure 1 shows the best SEM fit to the data (Model Chi square = 0.49, Df = 2, P = 0.78, Goodness-of-fit index = 0.99, Adjusted goodness-of-fit index = 0.97). Numbers next to arrows are standardized structural coefficients. Besides the global performance scores, one of the measures of participants' actions (Queries) was added in order to get an optimum number of indicators for the control factor. *Queries* recorded the number of questions participants asked about the features of the objects to be assigned. Note that there is no correlation between the number of questions asked in the ADM task and the global score on this task. While there is an optimum of questions one need to ask to get a high score, deviations from this optimum are sometimes beneficial and other times detrimental to the global performance on this task. For example, asking more questions could increase performance by allowing assignments to high scoring bins but can also decrease performance because it consumes time that can be used to make more assignments.

The SEM model shows that a latent *control* variable can indeed be defined. This factor has been interpreted as follows: high control involves actively gathering information from the environment (Queries), maintaining active and operating on recently processed information (N-Back), and suppressing incoming stimuli that could interfere with full and accurate processing of the current item (Blink). Ultimately, this control factor is involved – via working memory performance – in performing complex multitasking activities (ADM).

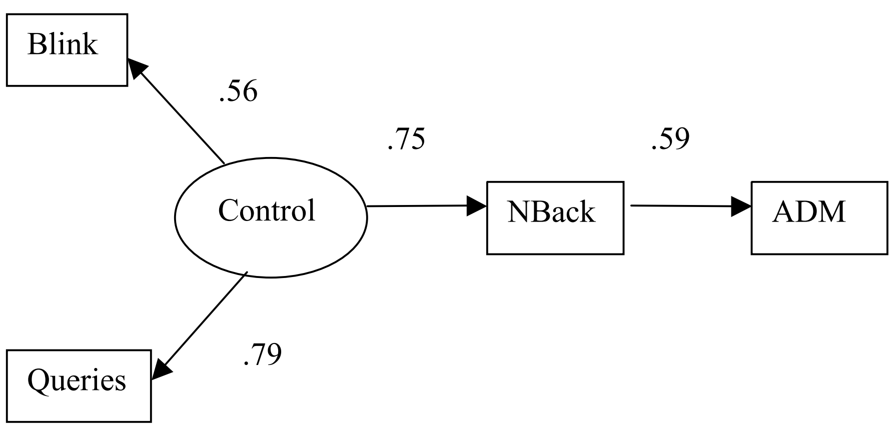


Figure 1. A SEM model showing correlates of and an underlying control factor involved in multitasking performance. Numbers next to arrows are standardized structural coefficients indicating the relative importance of each variable.

3. Models

ACT-R models of all tasks have been developed based on the principle previously used in modeling the DTT task: maintaining an elaborate control structure in the goal buffer is necessary to model high performance at multitasking.

This talk will focus on presenting ACT-R models of the N-Back task. These models use the built-in assumptions of the ACT-R architecture (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004) and recent findings in the working memory research (Baddeley, 2000; McElree, 2001; Akyurek & Hommel, 2005; Prabhakaran et al., 2000).

Two different models were developed in order to account for individual differences in the amount of top-down control dedicated to execution of the N-Back task: *low*- and *high-control* models. The *low-control* model uses previously acquired time estimation knowledge to decide whether the repeated item has occurred recently (lower *n*s in the N-Back series) or after some delay (higher *n*s in the N-Back series). Due to the error proneness of time estimations (Taatgen, Anderson, Dickison, D., & van Rijn, 2005), this model performs relatively poor at the N-Back task, fitting the data of low-performance human participants.

The *high-control* model uses a combination of two strategies: the *buffer* strategy and the *counting* strategy. The *buffer* strategy consists of maintaining a subsequence of presented items in the visual, aural, retrieval and imaginal buffers. This subsequence is updated by production rules that transfer information across buffers. The *counting* strategy uses a series of retrievals and the onsets of auditory events generated by sub-vocalizations of presented items to count back from the current item to the repeated one. Although this *high-control* model is also vulnerable to errors (e.g. chunk activation noise) its performance is relatively better at the N-Back task, fitting the data of high-performance human participants.

4. Conclusion and discussion

Results presented here (empirical data and simulations) suggest that top-down control is an important factor involved in working memory performance and multitasking. Achieving high performance at the N-Back task requires maintaining an elaborate control structure needed for coordination of retrievals (counting strategy) and transfers between buffers (buffer strategy).

The *counting* strategy was inspired by recent research showing that only one item can be maintained in focal attention at a particular moment and retrieval operations are used to reconstruct the linear order of recent events (McElree, 2001).

The *buffer* strategy was inspired by a new development of a classical theory of working memory (Baddeley, 2000) and fMRI research (Prabhakaran, Narayanan, Zhao, & Gabrieli, 2000) showing that temporary episodic information can be maintained in an efficient and accurate way by integrating current and recent information across different modalities. This integrative function is localized in the prefrontal regions of the brain (Prabhakaran et al., 2000), also thought to be responsible of control functions. Tulving (cited in Baddeley, 2000) reports a case of a densely amnesic patient who was able to play a good game of bridge; the patient was not only able to keep track of the contract but also of which cards had been played. These findings and anecdotic evidence suggests that processing of current and recent information is more a matter of control and integration than a matter of storage. Such a conclusion is also supported by research analyzing the relationship between working memory and the AB effect. It has been shown that decreasing the storage capacity of working memory (by giving items to be held in memory during the RSVP task) has no influence on the AB effect (Akyurek & Hommel 2005).

Baddeley (2000) and Prabhakaran et al. (2000) postulate the existence of a dedicated brain structure – an episodic buffer – that allows for temporary retention of information integrated across modalities. There is evidence that more information and in more efficient way can be maintained by the human brain when information is stored in poly-modal code than when it is stored in uni-modal code (Prabhakaran et al., 2000). We have used the goal buffer and the procedural module of the ACT-R architecture to simulate the control processes involved in maintaining availability of a changing sequence of items for current processing. Perhaps it is worth considering implementing a structural component dedicated to control and integration of information in the ACT-R architecture.

References

Akyurek, E.G., & Hommel, B. (2005). Short-term memory and the attentional blink: Capacity versus content. *Memory & Cognition*, 33, 654-663.

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

Baddeley, A. D. (2000). "The Episodic Buffer: a New Component of Working Memory?" *Trends in Cognitive Sciences* 4(11): 417-423.

McElree, B. (2001). Working Memory and Focal Attention. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 27(3), 817-835.

Prabhakaran, V., Narayanan, K., Zhao, Z., & Gabrieli, J.D.E. (2000). Integration of diverse information in working memory within the frontal lobe. *Nature Neuroscience*, 3, 85-90.

Taatgen, N., Anderson, J. R., Dickison, D., & van Rijn, H. (2005). Time Interval Estimation: Internal Clock or Attentional Mechanism? In B. Bara, L. Barsalou & M. Bucciarelli (Eds.), *Proceedings of the 27th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Associates.

Carnegie Mellon

The Role of Top-Down Control in Working Memory Performance: Implications for Multi-Tasking

Ion Juvina, Niels A. Taatgen, & Daniel Hasumi-Dickison
Carnegie Mellon University

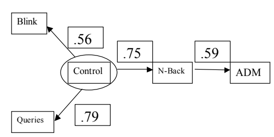
Background

- Akyurek & Hommel, 2005:
  - Look ma: no STM effect on AB!
  - STM effect on RSVP task but *not* on AB
    - i.e.: no interaction between memory load and *lag*
- We, 2006:
  - Correlation AB and N-Back *only at lag 3*!
- Thus: if no STM, then what?
  - McElree, 2001: *Executive control* in N-Back.

Outline

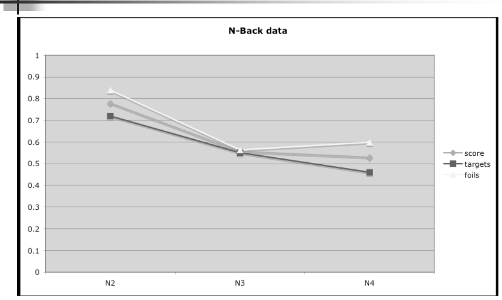
- Top-down control defined empirically
- N-Back data
- Predictions
- N-Back models
  - low control
  - high control
- Models vs. data (to be presented at the workshop)
- Conclusions

Control defined empirically

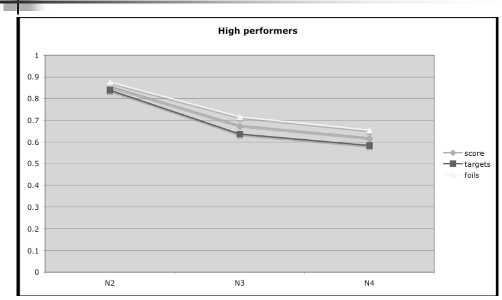


- Control = .312*Blink + .148*N-Back + .849*QUERIES

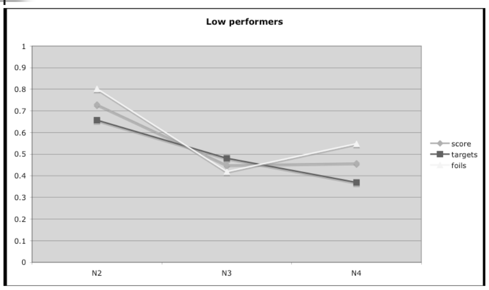
N-Back data



N-Back data (cont'd)



N-Back data (cont'd)



Predictions

- A low-control N-Back model (LCM) will fit the data of (empirically defined) low-control subjects (LCS)
- A high-control N-Back model (HCM) will fit the data of (empirically defined) high-control subjects (HCS)

Low-control model

- Associate a time stamp to each item
- Find repeated items in DM (not always)
- Find estimations of time distance in DM (not always)
- When estimation not found, react (or not) to repetition
  - Learn to react rather than not react based on feedback (utility learning)
- Learn new estimations from reactions
  - Learn correct estimations from feedback to reactions (declarative learning)

High-control model - "Buffers" strategy

- Keep as many items as possible in buffers
- The incoming item causes transfers across buffers:
  - From visual to aural
  - From retrieval to imaginal
- Specific productions identify targets and foils based on where items are (in which buffer)

High-control model - "Counting" strategy

- Find repeated item in DM
- Retrieve the items that have entered DM between the repeated item and the current one
- Increment a *counter* with each retrieval
- Compare the final counter with the *lag* slot of the goal
- Activation noise causes retrieval errors, thus counting errors.

Conclusions

- Control defined
  - In empirical terms: a factor underlying:
    - High performance at N-Back
    - High number of queries in ADM
    - High blink effect in RSVP
  - In ACT-R terms: number of control states in the goal buffer
- LCM fits LCS
- HCM fits HCS

An AI Planning Perspective on Abstraction in ACT-R Modeling: Toward an HLBR Language Manifesto

Robert St. Amant[1] **Sean P. McBride**[1] **Frank E. Ritter**[2]
(stamant@ncsu.edu, frank.ritter@psu.edu)
[1] Department of Computer Science, North Carolina State University, Raleigh, NC, 27695
[2] School of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802

Abstract

Researchers have again become interested in the translation of abstract specifications into the knowledge structures of executable cognitive models. Our work has adopted the Planning Domain Definition Language (PDDL), as an abstraction language for the automated generation of cognitive models, in a process we call search-based modeling. Our PDDL-based compiler, though incomplete, is currently being used to explore control issues in models for Towers of Hanoi problems. In this exploration, we have run into unexpected conceptual issues that we must address to move in the direction of the broader goals of abstract model specification. We discuss these issues: language simplicity versus search complexity, usability versus architectural complexity, and modularity versus veridicality, and suggest directions for further research.

Introduction

Researchers have again become interested in the translation of abstract specifications into the knowledge structures of executable cognitive models [Ritter et al., 2006], as part of an effort to develop high-level behavior representation (HLBR) languages. We have developed a system called G2A [St. Amant et al., 2004, St. Amant et al., 2006] that uses GOMS (specifically, models based on GOMSL [Kieras, 1999]) as an abstraction for cognitive models in the ACT-R architecture [Anderson et al., 2004]. G2A translates GOMSL models into ACT-R models using standard compiler techniques. GOMS has a number of desirable features as an abstract language; in particular, it shares with more detailed cognitive architectures many of the same basic assumptions about cognitive structure and performance (e.g. [Byrne, 2001, Kieras, 2002]). Nevertheless we believe that other possibilities for abstraction are still worth exploring.

In our recent work we have adopted PDDL, the Planning Domain Definition Language [Ghallab et al., 1998], in place of GOMSL. As with the translation process in G2A, ACT-R models are created by a search through the space of mappings from the states and actions of a plan to appropriate ACT-R constructs. Throughout this paper we will refer to this approach as “search-based modeling.” We chose a planning representation for practical and theoretical reasons. From a theoretical perspective, plans can be reasoned about more easily than ACT-R models expressed as productions and declarative memory initializations. A planning representation of a problem and its solution can be used to answer questions about models that would otherwise be difficult. From a practical perspective, we believe that a planning system may be able to reduce effort in modeling and to make cognitive modeling more accessible to designers of interactive systems. PDDL is not a perfect candidate for an abstract cognitive modeling language, but it allows us to exploit the decades-long history of AI planning research and system building.

This paper is divided into two parts. In the first, we describe the translation of PDDL domain and problem specifications into ACT-R models. Our PDDL-based compiler, though incomplete, is currently being used to explore control issues in models for Towers of Hanoi problems, a domain we use for illustration. In the second part, we discuss conceptual issues that arise in applying planning techniques to cognitive modeling.

ACT-R models and plans

ACT-R cognitive models and AI plans for specific domains share a number of conceptual commonalities in their knowledge structures. These include objects (chunks) with substructures, actions (productions) with conditions and effects, an initial state, and a goal state. In both models and plans, the basic approach involves representing what it is possible to do in some domain (e.g., stacking blocks, moving disks between towers, taking actions in a software system) and what information is gained through such interaction. Beyond this, however, we find significant differences between the ACT-R architecture and classical AI planners in how they approach problem solving.

Perhaps the most significant difference between AI planning and cognitive modeling, as represented by ACT-R, is in their treatment of control knowledge, in particular control knowledge specific to a given domain. Models are explanatory mechanisms for human cognitive behavior, and thus internal decision-making is fair game for representation. A planner, in contrast, can generally be treated as a black box: it is given a problem and a set of actions that reflect external environmental constraints on their execution, and it produces a solution.[1]

We can see the difference clearly in an example. An

[1] It is possible to represent internal cognitive constraints on problem solving in AI representations as well (e.g., [Howes and Payne, 2001, Howes et al., 2004]), but this is not common in domain-independent AI planning.

```
(define (domain hanoi)
   (:requirements :strips)
   (:predicates (clear ?x) (on ?x ?y) (smaller ?x ?y))

   (:action move
       :parameters (?disc ?from ?to)
       :precondition (and (smaller ?to ?disc) (on ?disc ?from)
                          (clear ?disc) (clear ?to))
       :effect (and (clear ?from) (on ?disc ?to)
                    (not (on ?disc ?from)) (not (clear ?to)))))

(define (problem hanoi4)
  (:domain hanoi)
  (:objects peg1 peg2 peg3 d1 d2 d3 d4)
  (:init
     (smaller peg1 d1) (smaller peg1 d2) (smaller peg1 d3) (smaller peg1 d4)
     (smaller peg2 d1) (smaller peg2 d2) (smaller peg2 d3) (smaller peg2 d4)
     (smaller peg3 d1) (smaller peg3 d2) (smaller peg3 d3) (smaller peg3 d4)
     (smaller d2 d1) (smaller d3 d1) (smaller d3 d2) (smaller d4 d1)
     (smaller d4 d2) (smaller d4 d3)
     (clear peg2) (clear peg3) (clear d1)
     (on d4 peg1) (on d3 d4) (on d2 d3) (on d1 d2))
  (:goal (and (on d4 peg3) (on d3 d4) (on d2 d3) (on d1 d2))))
```

Figure 1: PDDL representation for Towers of Hanoi

ACT-R 4 model for the Towers of Hanoi[2] contains five chunk-types, 21 chunks, and four productions. Of the four productions, two are used to modify disk locations (`final-move` and `move`) and two are used to push new goals onto the goal stack[3] (`start-tower` and `subgoal-blocker`). The two productions that modify the goal stack can be thought of as the primary algorithm (given ACT-R 4's computational machinery) for solving the problem. They continually push goals onto the stack, setting up future actions, until the `move` production fires, and an effect can be seen in the environment.

Contrast this model with a planning specification of the Towers of Hanoi problem,[4] as shown in Figure 1. There is a single action (`move`) that reflects only the physical and logical constraints of the problem environment: disks must be moved one at a time; larger disks cannot be placed on smaller disks; moving a disk "clears" any disk on the tower immediately below it and creates an "on" relationship with any disk already on the tower it is moved to. No information is given about the order in which actions must be carried out to solve the problem, aside from these constraints.

If we were to translate this single PDDL `move` action into an ACT-R production, the resulting model would not be able to solve the problem, because of the architectural differences between problem-solving approaches as described above. Naively, we might view the plan action as a template, to be filled in with different bindings to objects and translated into separate ACT-R productions. This will result in 210 different productions (one per ⟨ `from, disk, to` ⟩ combination), most of which will never apply; even with all these productions, there is still no explicit control knowledge to guide their execution—this model will take a long time to learn the solution, appears to lack much of the representation that people have about the task, and will likely provide a poor match to human performance.

It is possible to create a model from the problem specification if we add a solution generated by a planner, as shown in Figure 2. Taking a similar approach to that of G2A, we can generate ACT-R productions from these steps, with appropriate variable bindings and sequential execution constraints, to result in a model. Essentially we generate a state machine, represented as ACT-R productions, that encodes the necessary steps to solve the problem, with a "state" slot in successive goal chunks to maintain relevant state information.

While this approach proved effective in practice for G2A, in terms of predictiveness of expert behavior that could be represented with GOMS models for a relatively simple task [St. Amant et al., 2004], it remains conceptually unsatisfying: whatever explanatory power that more conventional models have for problems such as the Towers of Hanoi, in terms of internal problem-solving strategies, has been lost. In our future work we

[2] http://act.psy.cmu.edu/models/towerruiz.model

[3] In ACT-R 6, goal stack manipulation is deprecated, but the most direct translation of this model to ACT-R 6 still involves managing goals in a comparable problem-solving strategy [Leon Urbas, personal communication; Dan Bothell, personal communication].

[4] http://www.cs.washington.edu/homes/kautz/minichallenge/dagstuhl-mini-challenge.ppt

```
00: (move d1 d2 peg2)
01: (move d2 d3 peg3)
02: (move d1 peg2 d2)
03: (move d3 d4 peg4)
04: (move d1 d2 d4)
05: (move d2 peg3 d3)
06: (move d1 d4 d2)
07: (move d4 peg1 peg3)
08: (move d1 d2 d4)
09: (move d2 d3 peg1)
10: (move d1 d4 d2)
11: (move d3 peg2 d4)
12: (move d1 d2 peg2)
13: (move d2 peg1 d3)
14: (move d1 peg2 d2)
```

Figure 2: PDDL solution for Towers of Hanoi

will explore mechanisms to "generalize" plans to may reproduce internal strategies that may involve goal chunk manipulations, but many other questions remain open. We examine some of these issues in the next section.

Discussion

Our work with G2A and its PDDL-based successor is part of a broader effort in the research community, one that exploits (though perhaps only implicitly) an analogy:

> An abstract modeling language is to the ACT-R language as a high-level language is to assembly language.

By "high-level language" we include knowledge-based systems, planning languages, and the like, as well as high-level programming languages. As suggested in the introduction, while our preliminary work with PDDL does not yet offer many examples of this approach to cognitive modeling, we expect this situation to improve in the future. We nevertheless use it as a representative example of work in this area; for the purpose of discussion, imagine that we are able to solve the immediate issue discussed above: we are able to generate automatically from planning specifications ACT-R models that are largely indistinguishable from models built by hand, with comparable knowledge structures and comparable predictions about execution; the models may be easier to create, to use, less buggy, and perhaps able to explain themselves [Cohen et al., 2005]. Assessing the eventual goals of a research direction can help us better understand the pros and cons of pursuing it.

Opportunities

Many of the advantages of abstraction in cognitive modeling have been identified by others [Ritter et al., 2006]: abstract modeling languages may reduce effort in building models; they may make it possible to render higher-level idioms in explicit, consistent form; they may make cognitive modeling more accessible and thus more useful to interface developers.

A few advantages of search-based modeling have received less attention. PDDL is a planner- and architecture-independent specification language, which makes its planning constructs simple and explicit enough to reason about. As a basis for model generation, this offers the possibility of being able to answer questions about the structure of a model, possibly even before it is complete or directly executable. For example, *Can a given set of actions in principle reach a goal?* While planning is intractable in general, if a planner *is* able to generate a solution, this provides useful information about the ability of a model to solve the same problem. *What is the shortest path to a goal?* Many planning algorithms are defined such that they return the shortest plan possible (measured by the number of sequential steps), if one exists. (There are similarities here to cognitive constraint modeling [Howes et al., 2004].) *Are there different ways to reach a goal?* This is a less common question in planning research, but it still has a straightforward answer; a planning system can simply continue to search past its first solution for further possibilities. These questions treat a planning system as an analytical tool to improve model quality, analogous to tools for analyzing program properties. Even relatively abstract questions (concerning, say, the size of the space of plans searched before finding a solution) can give answers that are interesting from a modeling perspective (the size of the search space is related to the difficulty of a problem, given the representation).

Another implication of search-based modeling is that it should give modelers a better understanding of alternatives to their modeling decisions. In our work with G2A, search proceeded toward a set of target predictions provided either by an existing GOMSL model (e.g., the duration of a method) or by user data from a pilot experiment. Hill-climbing, with search steps based on alternative translations of GOMSL primitives to ACT-R productions, eventually produced a locally optimal model. We discovered that very similar predictions could be produced by different models. Given the restrictions on the structure of models that G2A was able to generate, these different models did not rely on significantly different modeling assumptions, but as our techniques improve, we believe this will change: an automated search should be able to generate models that incorporate differently structured knowledge and different problem-solving strategies.

Ideally, when an ACT-R model is presented, it should be accompanied by a discussion of alternative modeling decisions, if only to rule them out. By analogy, imagine building a regression model of a relationship for which there is incomplete knowledge about its functional form. In the process of building the model, we'll try different variable transformations, examine patterns in the residuals, check the significance of predictors, and so forth. In the end, we'll have accrued evidence that the final model is better than plausible alternatives.

This approach is difficult to apply to cognitive modeling because we have fewer computational tools to analyze our models, and in any case our models are less amenable to conventional statistical diagnostics. Further, the process itself of building models is difficult and time-consuming. Nevertheless we believe that it would be useful to be able to situate a given model in a space of modeling decisions, to say, "For this novel task, prior knowledge constrains a model to be of this form, and of the model variations that have such a form, this model provides the best predictions." An automated search through model space would be invaluable for this.

Tradeoffs

Because our research is in its early stages, it is appropriate to discuss the potential pitfalls of our general approach in terms of a set of design tradeoffs.

Language simplicity versus search complexity. A practical challenge in search-based modeling is deciding how much abstraction is appropriate. In the classical planning approach we have taken, for example, varying action duration and overlapping action execution are abstracted away during planning. (This was also the case in our G2A work.) While action durations can be represented in PDDL, the planners we have worked with do not take advantage of this information; actions are treated as atomic events that produce instantaneous change in the environment. The missing durations of actions are filled in when a model is executed in the ACT-R architecture. It would be reasonable to adapt temporal planning techniques to our approach, but we proceed under the assumption that the additional search complexity will not be worth the tradeoff in model abstraction.

Alternatively, we can move in a different direction: if we can represent existing ACT-R models of different tasks in a planning framework, it is a relatively small matter, from a theoretical planning perspective, to combine the separate task representations and allow a planning algorithm to interleave them appropriately for the purpose of model integration. This approach, done by hand, has worked well in some cases. However, doing this in general and across models by different authors appears to be difficult. We may need to extend the architecture, or modify how the knowledge is represented to support interleaving of actions for task switching. One question that arises is whether ACT-R models for executing multiple tasks can be built without the need for an executive, or (as with the function of a scheduler in an operating system) some meta-level control is required, or if a compiler is needed to create the knowledge so that it can be used this way, or some combination of these approaches.

It remains to be seen whether the planning representation we have chosen provides an appropriate level of abstraction, in terms of the benefit to modelers as well as the difficulty of managing search complexity. The next trade-off we can note explores this directly.

Modeling conveniences versus architecture extensions. As a software system, the ACT-R architecture naturally has many useful facilities, from well-tailored data structures to access functions to parsing utilities. We were tempted to take advantage of these facilities in building G2A, to translate GOMSL statements directly into internal ACT-R data structures. We resisted this temptation for good reason: the language of ACT-R acts as a specification for what constitutes a model.

To illustrate this issue, consider a software engineer building a large system. In a given module he might call a library function to sort a list, knowing nothing more than how to create the appropriate data structures for its input and output parameters and that the library function implements *QuickSort*. Many of the details are handled by the system. In compilation, data structures may be modified internally (e.g., in a Lisp system, a static list might be compiled into an array, improving the worst case performance of the algorithm); constant folding may eliminate some run-time computations; some function calls may be open-coded or even compiled away. In execution, a scheduler may swap out the process in which the sort function is executing; a multi-processor system (given a good compiler) may distribute the sort over multiple processors.

Even if the software engineer were to find the machine language version of his system incomprehensible, the system remains well-specified, in principle, with respect to a set of language-independent primitive operations and control constructs. Time and space complexity analysis can be carried out, for example, on a sort function independent of its implementation language, through examination of iterations and comparisons.

The ACT-R language provides a comparable specification level. Thus a G2A (or PDDL-based) search produces a well-defined model, explicit in the ACT-R language. If we had designed G2A to generate internal ACT-R data structures, we would have run the risk of blurring the boundary between the ACT-R architecture and G2A, with models being implicit in our system's output. Eventually it may be useful to think of add-on systems as extensions of the ACT-R architecture, but at the current stage of our work this seems premature.

Modularity versus veridicality. The most difficult theoretical challenge for search-based modeling is establishing the extent to which high-level descriptions can be mapped to low-level models.

From a software engineering perspective, it is possible to build large software systems in part because the problems they solve, as well as the systems themselves, are what Simon describes as nearly decomposable [Simon, 1996]. For programmers, decomposability means that complex solutions can be broken down into more easily handled parts. For programs, decomposability entails limitations on inter-module complexity and exchange of data.

While this is a common assumption in many models of cognition, it is an unresolved question whether modularity holds at different levels of abstraction for all cognitive processes that we might like to model. For example, Pylyshyn [**?**] argues that vision and cognition may be separable. At many levels and for much of the time, modular models (including ACT-R, particularly ACT-R 6) are modular, but there remain interactions between brain regions that may be important for some tasks and some analyses.

We conclude by observing that abstraction and modularity may bring benefits to model developers, but compiling abstract models will be more difficult than compiling conventional programs, because non-local interactions between memory structures and implicit procedures in models are not as well understood. We do not just want models to run or to fit data, but to predict and explain human performance.

Acknowledgments

This research is sponsored by the National Science Foundation under award ITR-0426852 and by ONR N00014-06-1-0164. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the United States Government.

References

[Anderson et al., 2004] Anderson, J., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Quin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111:1036–1060.

[Byrne, 2001] Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies*, 55(1):41–84.

[Cohen et al., 2005] Cohen, M. A., Ritter, F. E., and Haynes, S. R. (2005). Herbal: A high-level language and development environment for developing cognitive models in soar. In *Proceedings of 2005 Conference on Behavior Representation in Modeling and Simulation (BRIMS)*, Orlando, FL.

[Faraway, 1992] Faraway, J. J. (1992). On the cost of data analysis. *Journal of Computational and Graphical Statistics*, 1(3):213–229.

[Ghallab et al., 1998] Ghallab, M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D., and Wilkins, D. (1998). *PDDL—The Planning Domain Definition Language*. AIPS-98 Planning Committee.

[Howes and Payne, 2001] Howes, A. and Payne, S. J. (2001). The strategic use of memory for frequency and recency in search control. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*. Erlbaum.

[Howes et al., 2004] Howes, A., Vera, A., Lewis, R., and McCurdy, M. (2004). Cognitive constraint modeling: A formal approach to supporting reasoning about behavior. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, Chicago, IL.

[Kieras, 1999] Kieras, D. E. (1999). *A Guide to GOMS Model Usability Evaluation using GOMSL and GLEAN3*. University of Michigan, Ann Arbor, MI.

[Kieras, 2002] Kieras, D. E. (2002). Model-based evaluation. In Jacko, J. A. and Sears, A., editors, *The Human-Computer Interaction Handbook*. LEA, Mahwah, NJ.

[Ritter et al., 2006] Ritter, F. E., Haynes, S. R., Cohen, M., Howes, A., John, B., Best, B., Lebiere, C., Jones, R. M., Crossman, J., Lewis, R. L., St. Amant, R., McBride, S. P., Urbas, L., and Vera, A. (2006). High-level behavior representation languages revisited (group symposium). In Fum, D., del Missier, F., and Stocco, A., editors, *Proceedings of the Fifth International Conference on Cognitive Modeling (ICCM)*, pages 404–407, Trieste, Italy. Edizioni Goliardiche.

[Simon, 1996] Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press, 3rd edition.

[St. Amant et al., 2004] St. Amant, R., Freed, A., and Ritter, F. E. (2004). Specifying ACT-R models of user interaction with a GOMS language. *Journal of Cognitive Systems Research*, 6(1):71–88.

[St. Amant et al., 2006] St. Amant, R., McBride, S. P., and Ritter, F. E. (2006). AI support for building cognitive models. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI), Nectar Track*, Menlo Park, CA. AAAI Press.

Constraints and Complexities of Information Retrieval

Christian Lebiere
Carnegie Mellon University
cl@cmu.edu

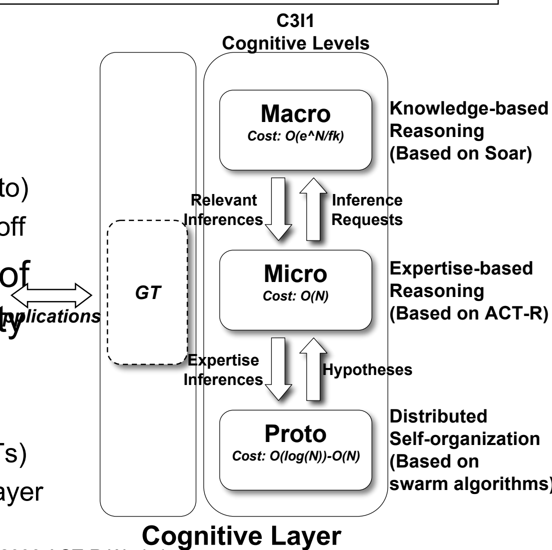
ACIP context

- Goal is to provide efficient real-time solutions to computationally demanding “cognitive” problems
- Must consider time/quality/resource tradeoffs
- Competing approaches span spectrum: cognitive architectures, machine learning bag-of-tools and unitary solutions (e.g. network message passing)
- Intelligence analysis (evidence marshalling) and UAV Mission Planning as domain examples
- Multi-level approach to cognitive architecture:
  - ACT-R as micro-cognition (expert pattern-matching)
  - Soar as macro/meta-cognition (inferencing, reasoning)
  - Swarming as proto-cognition (similarity/assoc clustering)

Design Motivation

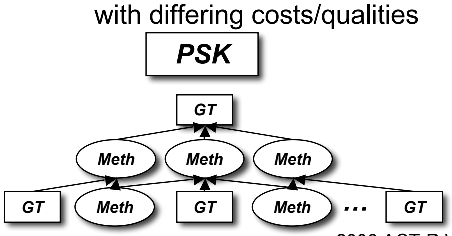
Controlled cost/quality tradeoff for complexity reduction

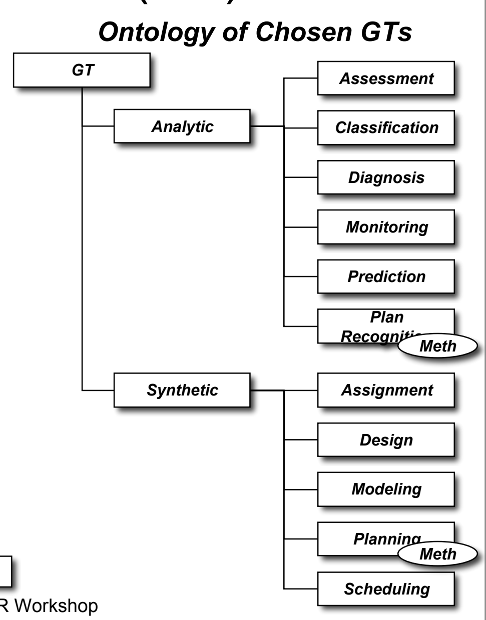
- Reduce effective complexity
  - O(Prob) = α O(Macro)+ β O(Micro)+ γO(Proto)
  - Quality/cost tradeoff
- Build a repertoire of reduced complexity solutions to hard problems
  - Generic tasks (GTs)
  - Interface to AppLayer



Generic Tasks (GT)

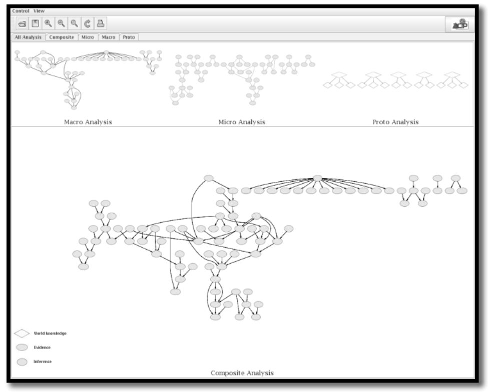
- Selected an ontology of GTs
  - Chosen to help solve a wide range of problems
- Generic Tasks (GT)
  - Provide generic problem solving knowledge (PSK)
  - May have multiple implementations (*methods*) with differing costs/qualities





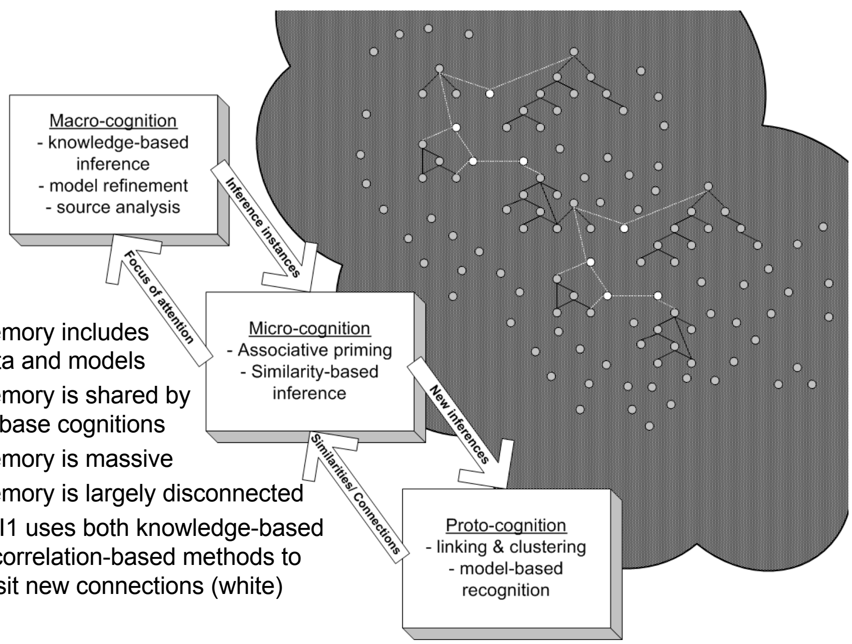
Evidence Marshalling

- Evidence Marshaling: Sign of Crescent (SoC) Problem
  - Three near-simultaneous terrorist acts being planned:
    - A dirty-bomb explosion aboard a ship in Boston harbor
    - A large bomb explosion aboard an Amtrak train (named “Crescent”) in Atlanta
    - A large bomb explosion inside NYSE
- SoC problem is used to train human analysts
  - Learning to marshal dynamic, incremental evidences (“trifles”)
  - Evidences are scattered spatially and temporally
  - Problem includes distractors – intelligence reports that have nothing to do with the plot



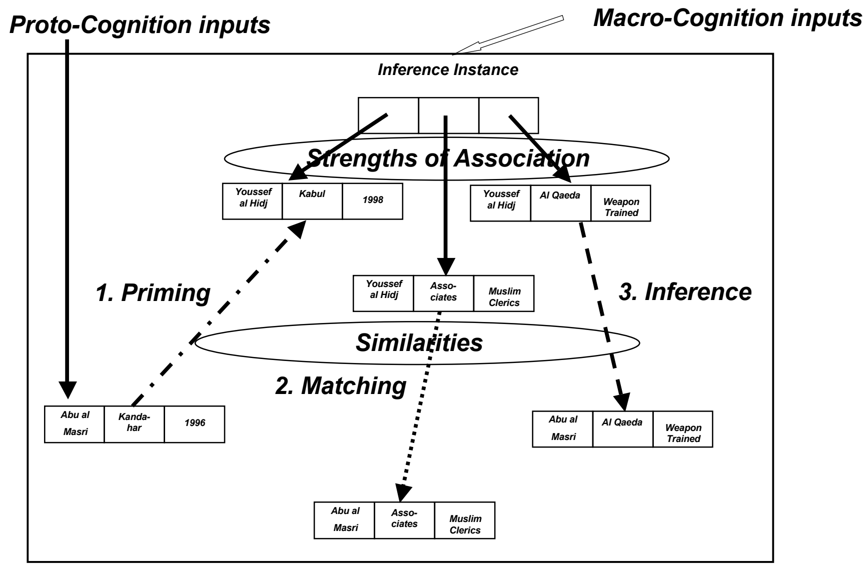
2006 ACT-R Workshop

Investigation: Data Flows in C3I1



- Memory includes data and models
- Memory is shared by all base cognitions
- Memory is massive
- Memory is largely disconnected
- C3I1 uses both knowledge-based & correlation-based methods to posit new connections (white)

Micro-Cognition Model



2006 ACT-R Workshop

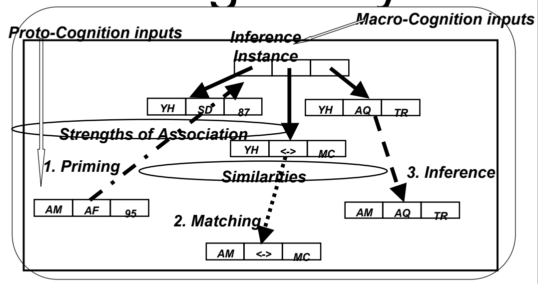
Supporting Evidence

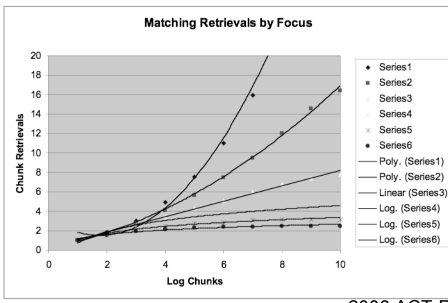
- ACT-R has also been applied to a number of information filtering and mapping problems in other domains including:
  - Analogy ***(Salvucci & Anderson, 2001)***
    - Analogical mapping very similar to our inference structure
  - Language Comprehension ***(Budiu & Anderson, 2001)***
    - Incremental building of linguistic representation has similar role for interpretation (hypothesis) and spreading activation/similarities (constraining possible interpretation)
  - Information Foraging ***(Pirolli, Fu, Reeder & Card, 2002)***
    - Strengths of association used to pinpoint relevant information amid large document clusters
- Similar use of spreading activation and similarity-based matching mechanisms and control structure provides supporting evidence for generic tasks

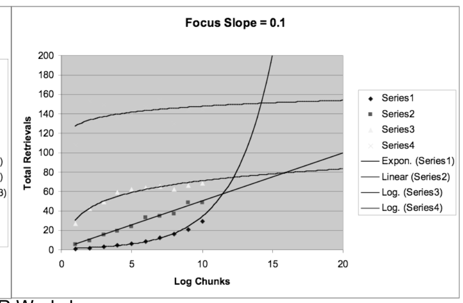
2006 ACT-R Workshop

Processes and Scaling Analysis

- Evidence Marshalling App (SoC): Plan Recognition Model
- Proto provides top focus cluster
- Macro provides expert knowledge
- *Quality of proto focus determines efficiency of micro expertise access*
- *Optimal amount of macro expertise depends on problem complexity*







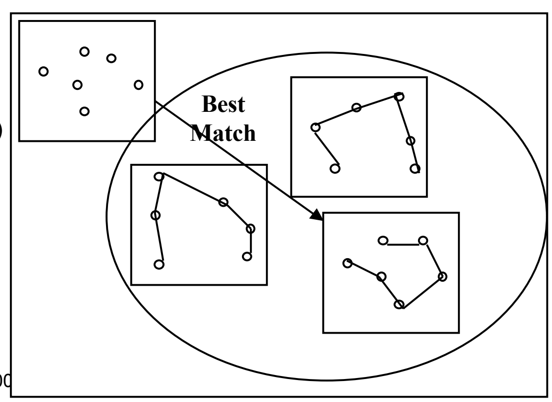
2006 ACT-R Workshop

Mission Planning

- Micro-cognition's role:
  - Provide a sequential control structure for hierarchical planning
  - Provide pattern matching-based primitives that improve planning performance with experience by accumulating expertise
  - Act as an integrating framework to structure the self-organizing proto computations and focus the macro deliberative reasoning

Expertise-based pattern-matching**:**

- **Basic trade-off between time (compute) and space (match)**
- **Hardware implementation (FPGA) yields constant-time complexity**
- **Improving efficiency of matching yields at least 2 orders of magnitude speedup**
- **Optimal performance ~60% of time**
- **Within 2% of optimum in average**



200

Lessons from ACIP

- Convergence between the purely functional goal of ACIP and the mixed biological-functional goal of BICA
  - Human cognition provides guidance in exploration of design space for efficient solutions (e.g. TSP)
- Multiple ways to optimize performance of architecture, including meta-cognition, learning and control system
- More flexibility in managing & representing context:
  - Automatic context management (working memory?)
  - Storing of richer processing context (episodic memory?)
  - Allow context to influence memory retrieval (2.0 vs 4.0 vs p*)
- Importance of addressing computational complexity
  - Theoretical complexity analysis (cognitive operations / pbm size)
  - Scaling of cognitive operations (DM/PM in FPGA/RAM)

2006 ACT-R Workshop

Parameter Space Explorations Using High Performance Computing

Glenn Gunzelmann (glenn.gunzelmann@mesa.afmc.af.mil)
Kevin A. Gluck (kevin.gluck@mesa.afmc.af.mil)
Jeffrey Kershner (jeffrey.kershner@mesa.afmc.af.mil)
Air Force Research Laboratory
6030 South Kent Street; Mesa, AZ 85212 USA

Using High Performance Computing

The combinatorics associated with an exhaustive parameter space searches for computational cognitive models typically prevents it from happening, because the computational demands are overwhelming. Yet such parameter space searches can be important for two main reasons. Firstly, in many models parameter space searches are conducted to identify best-fitting values for manipulated parameters. However parameter space searches can also be important to explore the robustness and flexibility of the model, by characterizing the range of behaviors it can produce (e.g., Estes, 2002; Pitt, Kim, Navarro, & Myung, 2006).

Our proposed solution to the challenge of accomplishing large-scale parameter-space searches for these kinds of analyses is to farm out parameter searches to the Aeronautical Systems Center's Major Shared Resource Center for High Performance Computing (HPC), which is located at Wright-Patterson AFB, Ohio. That facility has 2,048 processors running at 1.6 GHz, with 1GB of memory per processor and 100 Terabytes of data storage capacity. Over the last several months we have established a relationship with the HPC center and have begun exploring the use of this resource for testing and validating computational cognitive models. We have demonstrated the ability to execute a small model batch run on the HPC resources and we expect to complete some initial evaluations of the gain in efficiency from using the HPC processors before the ACT-R workshop begins.

The presentation will focus on the efficiency gains that can be achieved and the technical requirements for realizing these gains in computational modeling applications. Our calculations based on preliminary results suggest we can expect approximately a two order of magnitude improvement in turnaround time utilizing a relatively modest proportion of the resources that are available. A parameter search that took 17 days on a single processor in our lab should be complete in less than four hours via HPC. With a more aggressive use of HPC resources and larger parameter spaces, the gains could be even greater.

There are more challenging issues associated with improving the sophistication with which we are taking advantage of the HPC resources, however. For instance, small-scale, local parameter optimizations are often done with ACT-R models using gradient descent search algorithms that minimize deviations between model and human performance data. We would like to do that on a large scale, via distributed computing with the HPC. We also plan to use the HPC resources to continually validate new theoretical claims against previously used tasks and datasets, thereby objectively quantifying the cumulative progress we are making in our computational theories. This is rarely done in the computational cognitive modeling community, because there isn't an infrastructure with sufficient computational resources and adequate automation to support it.

Conclusion

We have made a commitment to investing in developing an infrastructure to facilitate large-scale parameter-space explorations for validating and testing computational cognitive models. A technician at the HPC Center at WPAFB referred to this task as "embarrassingly parallel." In other words, this kind of application is perfectly suited to being run more efficiently using HPC resources.

Using the substantial resources available at facilities like the HPC center at Wright-Patterson AFB should allow us to answer the challenge of Pitt et al. (2006) and others to explore the full range of behaviors that a model can produce. This addresses the robustness of the model, by characterizing the qualitative (and quantitative) patterns of data a model is able to produce. Such resources can also speed the model fitting process, by allowing researchers to distribute the search through the parameter space among hundreds of processors. We look forward to solidifying the foundation we have developed thus far, and applying the capability to our model validation efforts.

Acknowledgments

The research described here was sponsored partly by AFRL's Warfighter Readiness Research Division and partly by grant #04HE02COR from the Air Force Office of Scientific Research (AFOSR).

References

Pitt, M. A., Kim, W., Navarro, D. J., & Myung, J. I. (2006). Global model analysis by parameter space partitioning. *Psychological Review, 113(1),* 57-83.

Estes, W. K. (2002). Traps in the route to models of memory and decision. *Psychonomic Bulletin & Review, 9(1),* 3-25.

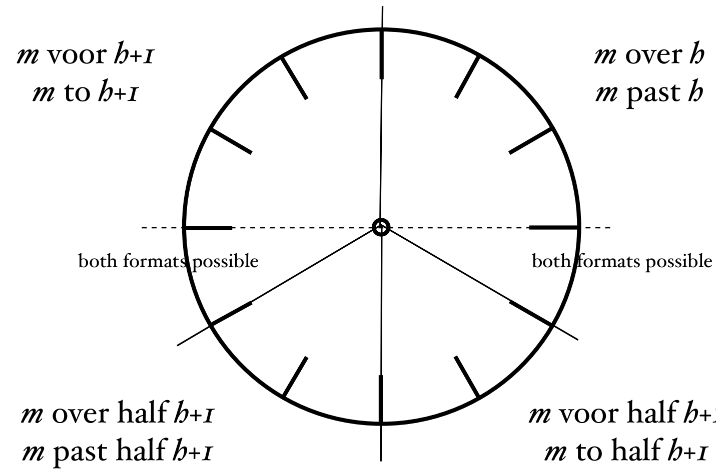
Complex model validation by multi-level modeling

or:
How do the Dutch tell the time?

Hedderik van Rijn
Artificial Intelligence
University of Groningen

Simone Sprenger
Max Planck Institute
for Psycholinguistics,
Nijmegen

Telling the Time in Dutch



Why do I care?
(Why do psycholinguists care?)

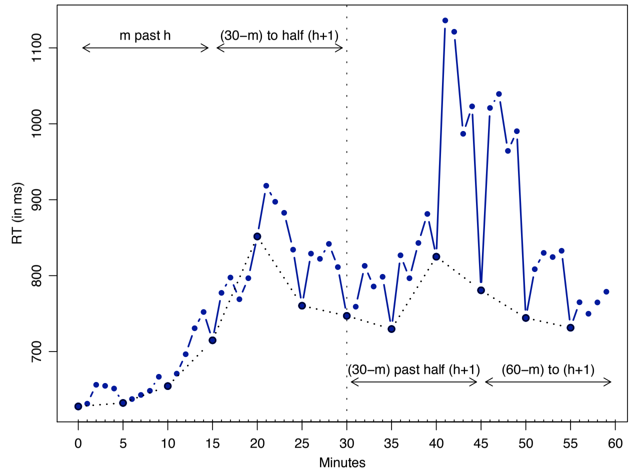
- Language perception is relative simple to study - the experimenter has full control over the input.
- Language production is more difficult to control
  - Pre-learned utterances
  - Very constrained settings - having the risk of being either too simple or too unnatural tasks
- How to elicit multi-word utterances without training in an relative unconstrained setting?

Enter Clock Time Naming

Stimulus	*Response*	
2:00	om twee uur	at two hour
2:10		en past two
2:15		arter past two
2:25		e to half three
2:30	om half drie	at eight past half three
2:38	om acht over half drie	at eight past half three
2:54	om zes voor drie	at six to three

Different utterances for similar inputs.

Experimental Data

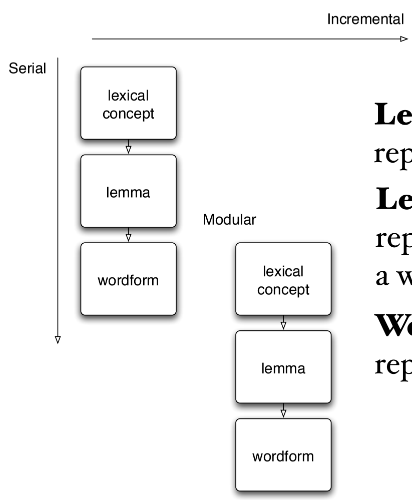


5

Language Production Model

- Levelt, Roelofs & Meyer (1999)
  - Partly implemented as WEAVER++ (Roelofs)
- Keywords:
  - ***Serial***: each part of the utterance is processed in a serial fashion
  - ***Modular***: different steps in the production of language are independent, output of one module is input for the next.
  - ***Incremental***: parts of utterances can be incrementally processes (serial tracks can run in parallel)

6



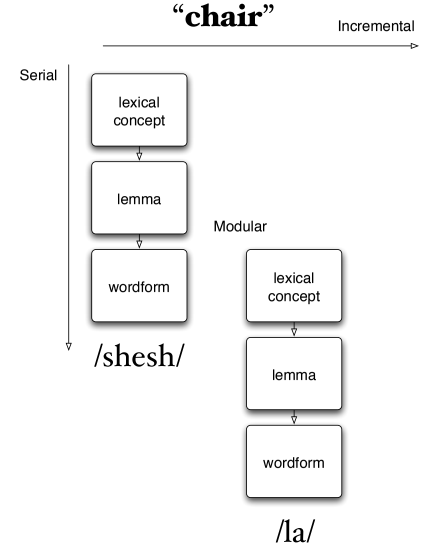
Lexical concept: concepts represented by a "word"

Lemma: abstract syntactical representation, e.g. gender of a word

Wordform: phonological representation

7

"chair"



la chaise

8

What are “words”?

- “chair” - /the chair/
  - “chair” - /la chaise/
  - “lemon” - /le citron/
- “” - /kicked the bucket/
- “” - /be in a bad [mood|temper]/
- “” - /a quarter to [hour]/

“Simple” words

“Superlemmas”

Superlemmas can contain slots

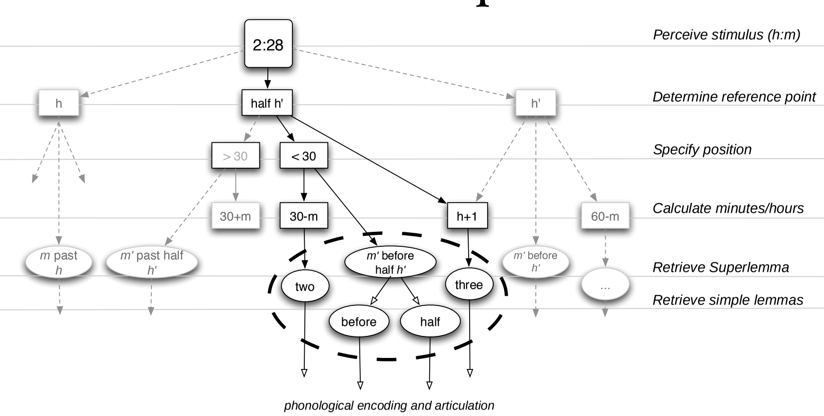
9

Limitations of the model

- Has been successfully used to predict different types of language production, interference phenomena, etc.
- Multiple word utterances can partly be explained by superlemmas, **but how to explain the underlying processing?** (“Thinking for speaking”)

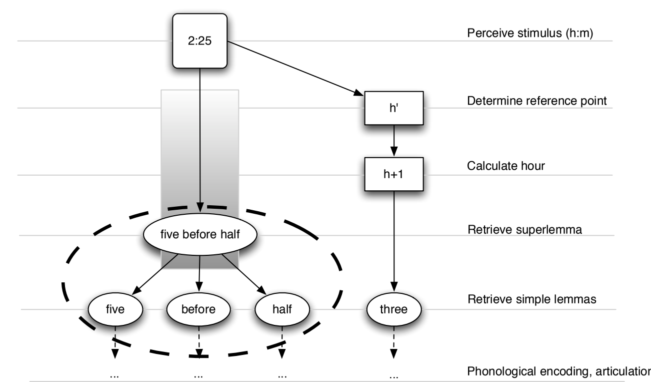
10

An ACT-Rish model of Clock Time Naming
Route I: Computation



11

An ACT-Rish model of Clock Time Naming
Route II: Retrieval



12

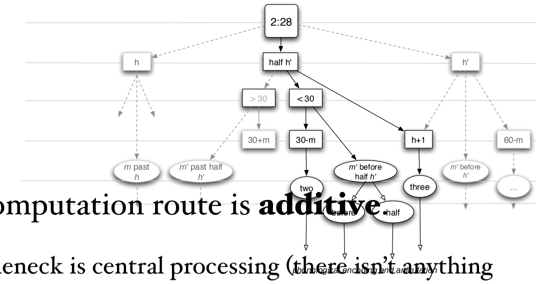
Elements of the model

Route I	Route II
Frequency of Superlemma	Reference point
	Association with reference point
	Equation building
	Arithmetic
	Frequency of utterance
	Competition with alternative formats
	Word properties (phon/syntax)

- **Are all these steps really necessary? Isn't this model overly complex?**
- **How to test if this model is correct?**

13

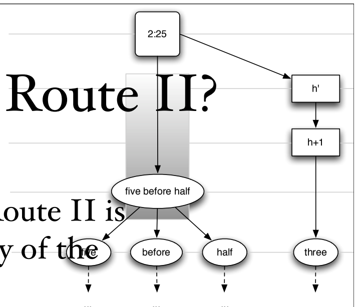
Lucky us!



- The proposed computation route is **additive**.
  - This model's bottleneck is central processing (there isn't anything else to do that would modify the voice-key onset), given the assumptions of Levelt et al's language production model.
- Therefore, Donders' (1868) "**subtraction method**" is viable - although Luce's (1986) comment on "**pure insertion**" remains valid.

14

But what about Route II?



- The probability of taking Route II is dependent on the frequency of the Superlemma.
- The higher the frequency, the higher the **chance** of taking that Route.
- Therefore, one would expect an interaction between the frequency of the Superlemma and the influence of Route I elements.

15

Multi-Level Models

- Statistical models with:
  - **fixed terms**: terms that are constant over participants.
  - **random terms**: terms that vary per participant
- Methods exist for model selection: "Does this extra variable explain sufficient variable to warrant inclusion in the model?"
- Fixed and random effects make averaging over data unnecessary

16

Constructing a statistical ACT-R model

Route I

Frequency of Superlemma

Route II

Reference point

Association with reference point

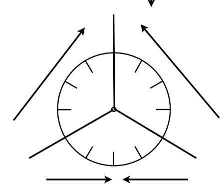
Equation building

Arithmetic

Frequency of utterance

Competition with alternative formats

Word properties (phon/syntax)



Constructing a statistical ACT-R model

Route I

Frequency of Superlemma

Route II

Reference point

Association with reference point

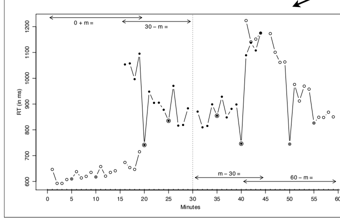
Equation building

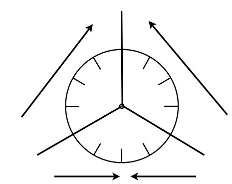
Arithmetic

Frequency of utterance

Competition with alternative formats

Word properties (phon/syntax)

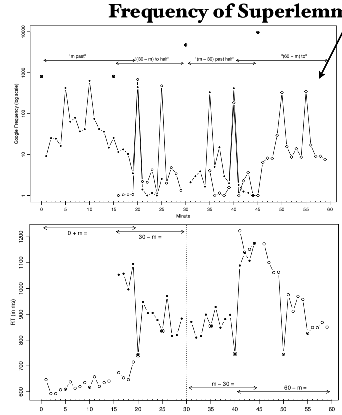




Constructing a statistical ACT-R model

Route I

Frequency of Superlemma



Route II

Reference point

Association with reference point

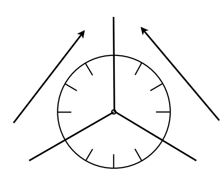
Equation building

Arithmetic

Frequency of utterance

Competition with alternative formats

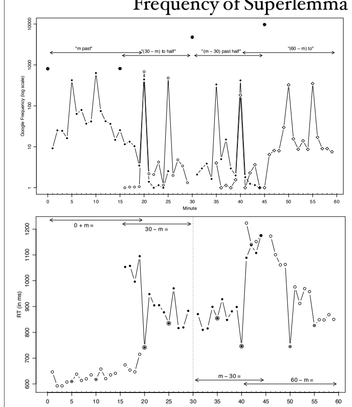
Word properties (phon/syntax)



Constructing a statistical ACT-R model

Route I

Frequency of Superlemma



Route II

Reference point

Association with reference point

Equation building

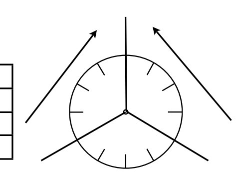
Arithmetic

Frequency of utterance

Competition with alternative formats

Word properties (phon/syntax)

	phonLen	nMorp	...
one	3	1	...
two	3	1	...
...	...	...	...



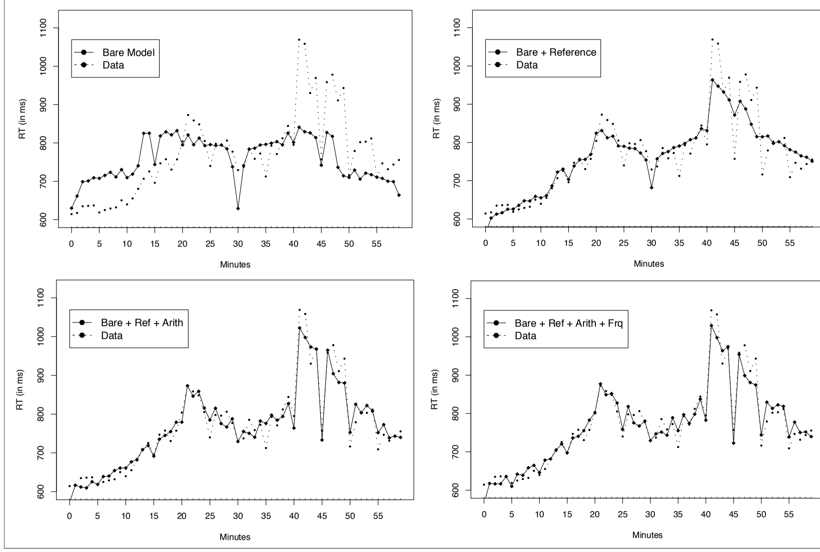
The Models

log(RT) = + (1 | Participant)

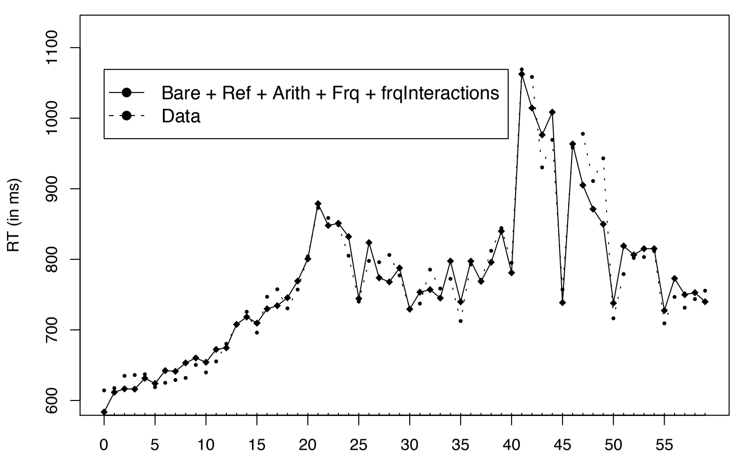
#	Fixed Effects	R^2	$\overline{R^2}$	df	logLik	X^2
1	phonLen + #Morph + Frq(Min) + hour	0.298	0.351	12	-165	
2	+ reference point + association with ref point	0.394	0.842	15	586	1504***
3	+ log(RTcalc) + equationToHalf + eqFromHalf	0.412	0.935	18	741	308***
4	+ Frq(PrefUtterance) + Frq(AltUtterance)	0.414	0.942	20	744	7*
5	+ Frq(PU) x log(RTcalc)	0.416	0.950	21	755	23***

Thus, each model is an improvement over the previous, simpler model - even though the increase in R^2 is very small.

Model 1-4 Fit



Model 5 Fit: Computation and Retrieval



Conclusions

- The combination of ACT-R and Levelt et al's language production model provide solid and testable hypotheses
- Complex data *and* models can be easily tested using multi-level statistical models
- Clock time naming, *even in Dutch*, is easily explained by a set of underlying variables

ACT-R versus not-ACT-R: Demonstrating Cross-Domain Validity

Terrence C. Stewart (terry@ccmlab.ca)
Robert L. West (robert_west@carleton.ca)
Institute of Cognitive Science, Carleton University
Ottawa, Ontario, Canada, K1S 5B6

Introduction

The goal of creating a cognitive architecture is to develop a single system that can account for results across all domains (Newell, 1990). ACT-R is currently the most promising candidate in this direction, having been validated in a wide variety of situations. However, Many critics of ACT-R (and computational modeling in general) believe that, with enough tweaking, an ACT-R model could be produced for any experimental observation (see Roberts & Pashler, 2000). To some degree this is can be dealt with by having fixed parameter settings or theories about when the parameter settings vary (Anderson & Lebiere, 1998). However, as argued more completely in (Stewart, 2006), another way to address this problem is to not only demonstrate that ACT-R models fit various observations, but also *that other models do not*. To do this, we need to be able to apply completely different architectures to the same situations as our ACT-R models. Furthermore, we should follow a similar approach for variations on ACT-R itself.

For example, to show that the PG-C learning rule is correct, we need to not only show that it results in predictively accurate models in a variety of situations; we also need to show that an alternate learning rule (such as Q-Learning, or some other Reinforcement Learning strategy) does not. Alternatively, we may determine that a variety of learning rules (over a specified range of parameter settings) all produce equivalently accurate results over a set of tasks. In this case, we can potentially identify the unique aspect that separates accurate models from inaccurate ones. Similar considerations exist for those researchers developing variations on ACT-R modules, such as the spacing effect (Pavlik & Anderson, 2005) or various production weighting schemes (Gray, Schoelles, & Sims, 2005).

Modular Model Creation

To achieve this goal of examining a wide variety of models (both ACT-R-based and non-ACT-R-based), we need to be able to rapidly construct models, and to easily reorganize the basic structure of ACT-R. This can include construction of new modules and buffers to extend ACT-R, or adjusting various fundamental formulae. Python ACT-R (Stewart & West, 2005), which is a re-implementation of ACT-R within the Python programming language, was created to facilitate this. In creating Python ACT-R the goal was to make it as open as possible to modify the ACT-R architecture.

Also, to create experimental environments for the resulting models and to analyze the data, the Carleton Cognitive Modelling Suite was created (Stewart, 2006). This includes tools for the exploration of parameter spaces, the use of equivalence testing rather than correlation or mean-squared-error for model evaluation, and a variety of non-ACT-R systems, including neural networks, reinforcement learning, and genetic algorithms.

All software, including implementations of the spacing effect (Pavlik & Anderson, 2005), production weighting (Gray, Schoelles, & Sims, 2005), the SOS vision system (West, Emond, & Tacoma, 2005), and both Q-Learning and TD-learning for productions (Fu & Anderson, 2004) are freely available at <http://ccmlab.ca/ccmsuite.html>.

References

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

Fu, W-T. & Anderson, J. R. (2004). Extending the Computational Abilities of the Procedural Learning Mechanism in ACT-R. In proceedings of the 26th Annual Conference of the Cognitive Science Society.

Gray, W. D., Schoelles, M. J., & Sims, C. R. (2005). Adapting to the task environment: Explorations in expected value. Cognitive Systems Research, 6(1), 27-40.

Newell, A. (1990) Unified Theories of Cognition. Cambridge, MA: Harvard University Press.

Pavlik, P. I. and Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. Cognitive Science, 29, 559-586.

Roberts, S. and Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. Psychological Review. 107 (2), 358-367.

Stewart, T.C. (2006) Tools and Techniques for Quantitative and Predictive Cognitive Science. 28th Annual Meeting of the Cognitive Science Society.

Stewart, T.C. & West, R. L. (2005) Python ACT-R: A New Implementation and a New Syntax. 12th Annual ACT-R Workshop

Stewart, T.C. & West, R. L. (2006) Deconstructing ACT-R. Seventh International Conference on Cognitive Modelling.

West, R. L., Emond, B., & Tacoma, J. (2005) Simple Object System (SOS) for creating ACT-R environments: A usability test, a test of the perceptual system, and an ACT-R 6 version. Proceedings of the Annual ACT-R Workshop

Act-R Theory *Almost* Provides a Formula for Predicting the Rate of Post-Completion Errors

Simon Li & Richard M Young

Context

- Postcompletion error
  - Forgetting to execute an action after the aim of the current subgoal has been achieved
    - E.g. leaving the original on the photocopier; forgetting your cash card, …
  - Sensitive to WM load
  - WM capacity in 3CAPS (Byrne & Bovair, 1997)
- Deterministic model
  - All-or-nothing error behaviour (0% or 100%)

Our question

- Can we achieve a simple non-deterministic model?
  - using just the basic (noisy) conflict resolution mechanism of ACT-R
  - which settles to a PCE rate of say ~5%
- Of course, this is far too simplistic a model to be psychologically real
  - but it serves as a baseline for more sophisticated models
- Issues of "parameter learning" will become relevant

Our approach in terms of ACT-R

- An interpreter to "carry out" a hierarchical task (chocolate vending machine)
  - will you remember to collect change?
- The PCE is based on ACT-R's conflict resolution
  - a mechanism for selecting the next atomic action
- PCE --> competition between two rules:
  - MTNG (move-to-next-goal) and TS (terminate subgoal)

"Success Rate" for PCE

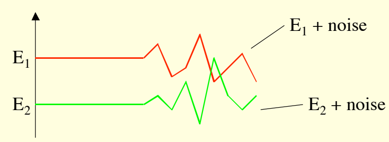
- Need to introduce one further wrinkle …
- Although a PCE is, by definition, an "error" for us as observers, it is not necessarily an error for the person (or the cog architecture)
  - e.g. you forget to collect your change from the ticket machine … but do you ever become aware that you've done so?
- If not, then it's not an "error" for the architecture
- Use P_e to represent the "success" rate of the PCE action

Mutual Dependence of Parameters and Behaviour

- Behaviour depends upon the value of the production parameters, P_i and C_i
- But the parameter values themselves depend upon (are learned from, are estimated from) the behaviour
- We need to understand both sides of this mutual dependence

Behaviour Depends on Parameters

- Add to each of the E_i a random quantity (noise) of spread determined by global *s*
- The action with momentarily largest E is chosen

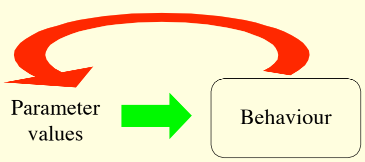


- P(choose $action_2$) depends on ratio $(E_1–E_2)/s$

Parameters Depend on Behaviour

- Parameter values are "learned" by (Bayesian) estimates from experience
- Benefit P is just the experienced success rate of the action
- Cost C is just the average experienced cost of the action

Mutual Dependence (Cyclic)



Consistency of Parameter Values

- The parameter values learned from a pattern behaviour do not necessarily coincide with the values producing that behaviour (!)
- Suppose we want the model to select the PCE action around 5% of the time
  1) then we need $E_{PCE} < E_{correct}$
     - otherwise make more errors than correct
  2) but also need $E_{PCE} \approx E_{correct}$
     - otherwise make no errors at all

Important Implication

- This means that

  The correct and PCE actions must have Es that are approximately equal

- But also, just being approximately equal will not of itself produce the 5% error rate

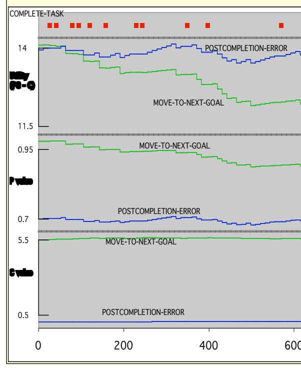
Calculating Parameter Values for Consistency

- We can use this result to calculate the conditions under which the parameter values will be *consistent*
  - i.e. the values experienced in the behaviour coincide with the values generating the behaviour

The Forward Calculation
(details suppressed)

- Suppose the rate *r* of making PCEs is set, say at 5%
- We calculate $E_{correct}$ and E_{PCE} in terms of the objective properties of the task and P_e. But we know $E_{PCE} \approx E_{correct}$. So we equate them, and solve.
- This gives values for all the parameters and around 0.7 for P_e (i.e. for Simon's task)

Run of model with $P_e = 0.7$



The Backward Calculation

- It occurred to me only later …
- Instead of assuming $r = 5\%$ and calculating the parameters and P_e, I could leave *r* as an unknown in the equations, then **solve** for it!
- Amazingly, the equation turns out to be linear in *r*. So, multiply out, collect terms, etc. and we get …

Formula for PCE Error Rate

$$r = \frac{M_s((P_e G - C_e) - (G - C_{ss}))}{(P_e G - C_e)(M_s - M_e) + M_e(P_e G - C_{es}) - M_s(G - C_{ss})}$$

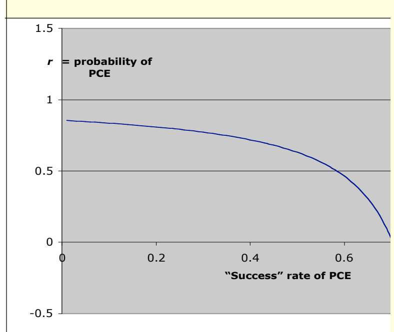
- Looks a mess, but all the terms on the right are known properties of the task (and/or task environment)

What does *r* Look Like?

WARNING

Don't try this at home!

r as a function of P_e



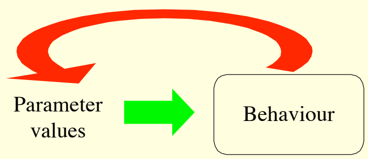
Do we Need to Worry?

- Not a worry that can be outside range [0,1]
  - just means no "real" solution for certain cases
  - after all, when did the forward calculation, didn't bother to note that r in range [0,1] and therefore a restricted range of values possible
- The whole idea of having such an formula seems a bit disturbing …
  - let's return to this at the end

Do we really have what we seem to have?

- i.e. a way of calculating PCE rate in terms of objective properties of the task?
- Unfortunately (or fortunately?), not
- Although the value given by the formula is *consistent* (i.e. a "fixed point"?), it is *unstable*
  - at least for the few cases we have analysed
- (Rather like balancing a pencil on its point)

Systems with Feedback can be Unstable



- If the "gain" round the loop is +ve, then the system is unstable

Stability Analysis
(details suppressed)

- Suppose the system happens (by chance) to make *n* PCEs in a row
- Can show that this *reduces* the difference between E_{PCE} and $E_{correct}$
  - i.e. makes it ***more likely*** that PCEs will occur in future
  - (argument is simple and quite elegant, but not given here)
- Therefore, system is unstable

Discussion — 1

- What would it have meant if we *hadn't* run into the problem of stability? Could we really have a formula for PCE rate
  - independent of individual (knowledge, motivation, concentration, …)?
  - independent of details of cognitive architecture (only very broad assumptions made)?

Discussion — 2

- I think we could
- There are just enough assumptions in the analysis to cover plausible individual variation
  - rationality & accuracy of decision making
  - accurate tracking of Ps and Cs

Discussion — 3

- How inevitable is that instability?
- Could there be similar cases where the solution is in fact stable?
  - don't know
  - would take further work
  - not sure I'd know how to do it

END

Version Control

- First given, with Simon, at UCLIC 3.11.04
- Shortened version, at Act-R workshop 23.7.06

Demonstration of Instability — 1

- I've carried through several different ways of deriving the result, but the easiest and most transparent seems to be a rigorous but quite informal argument, as follows.
- We have two critical actions, S and F, which compete at a certain point of the process. S leads to success, always. F leads to an observed PCE, but nonetheless "succeeds" a proportion P_f of the time (which in the body of the talk we have called P_e.)
- F is chosen on a proportion *r* of trials, which is thus the PCE rate. S is chosen on $(1 - r)$ of trials. Typically *r* is small, say around .05.
- For F, its probability of success is P_f, as just stated. Its cost is a small, fixed cost C_f, which in Act-R would be typically around 0.05 sec, since it terminates the task immediately.
- For S, its probability of success is P_s, which is typically high, say around .98. (It's less than 1 because S may fire more than once per run, and so may be involved in runs where F wins the competition and leads to failure.)

Proof of Instability — 2

- For S, its cost is C_{ss} on runs where S is chosen, but a lesser C_{sf} on runs where F is chosen and cuts the process short. Its actual parameter C_s is a weighted sum of the two, in fact $C_s = (1–r)C_{ss} + rC_{sf}$
- We've already seen that in order for errors to occur, the expected gains of S and F must be approximately equal, in other words
$$E_s = P_sG - C_s \approx E_f = P_fG - C_f \qquad (1)$$
- We now consider the effect of n consecutive choices of F on the system. The parameters for F, P_f and C_f, are not dependent on the choice of action, so we only need to consider the effect on P_s and C_s.
- Suppose that the parameters for S have been learned as experienced over N trials, where typically N will be quite large. The effect of n further trials is that P_s, C_s, and indeed $E_s = P_sG - C_s$ will all be updated as the weighted average of their values learned over N trials with the values experienced during those further n trials.

Proof of Instability — 3

- So from the original N trials, the experience for action S leads to learning $E_s = P_sG - C_s$.
- For the further n trials during which F is chosen, S experiences success at the rate of P_f, and cost at the level of C_{fs}. So based on just those n trials it would learn $E_s(n) = P_fG - Cf_{fs}$.
- Since $C_{fs} > C_f$, we have $E_s(n) < P_fG - C_f$, i.e. $E_s(n) < E_f$
- But we know $E_f \approx E_s$
- This means that $E_s(n) < E_s$, which in turn means that the experience of those n trials will *reduce* E_s when combined as a weighted average.
- Consequently, the probability of choosing F in the future is increased, which means that E_s will be reduced still further …

The Influence of Belief on Relational Reasoning: An ACT-R Model

Adrian P. Banks, University of Surrey, UK

Whilst early reasoning experiments sought to minimise the influence of background knowledge on performance, more recently the role that prior knowledge plays in reasoning has become an important topic in its own right. One approach to studying this is to test simple deductive logic problems that contrast logical conclusions with believable ones. For example:

Edinburgh is north of Cambridge	London is north of Cambridge
London is south of Cambridge	Edinburgh is south of Cambridge
Conclusion: London is south of Edinburgh	*Conclusion:* Edinburgh is south of London
This is logically valid and believable.	This is logically valid but unbelievable.

This research aims to understand how the believability of these problems influences our ability to draw logically valid conclusions from them.

Major Empirical Findings

Three major effects have been found: people are more likely to accept logically valid conclusions, people are more likely to accept believable conclusions, and - most interestingly - this effect of believability is stronger for invalid than valid conclusions. In particular, this latter finding occurs when conclusions are indeterminately invalid (conclusion is possible but not necessary) but not determinately invalid (conclusion is not possible). These effects have been found with categorical syllogisms (e.g. Evans, Barston & Pollard, 1983) and relational reasoning problems (Roberts & Sykes, 2003). Relational reasoning problems test the spatial and temporal relationships between things, and these are the problems that will be modelled here.

ACT-R Model of Belief Bias

There are three stages to the model's operation. (1) The premises are read and integrated into a single chunk. The chunk represents a 3x2 grid, with one slot per cell of the grid. This captures the spatial relationships of the elements in the problem. Premises may not uniquely identify a layout, so they are reread until all alternative layouts consistent with the premises have been found. (2) The conclusion is read and all chunks with sufficient activation are retrieved. (3) If all chunks retrieved are consistent with the conclusion then a valid response is made. If some are inconsistent, then an invalid response is made. If no chunks have sufficient activation, then the model guesses, with a slight bias towards supporting believable conclusions.

The effect of prior belief is modelled by placing a chunk in declarative memory which has some initial base level activation. When chunks are created from the premises that are consistent with this, they are merged with it raising its activation further. When the premises are not consistent with prior belief, new chunks are created which have a lower activation. Hence the influence of prior belief arises because chunks derived from the premises that match prior belief have higher activation because of the chunk merging, and so they are more likely to be retrieved and influence the conclusion evaluation than those that do not match prior belief.

Results

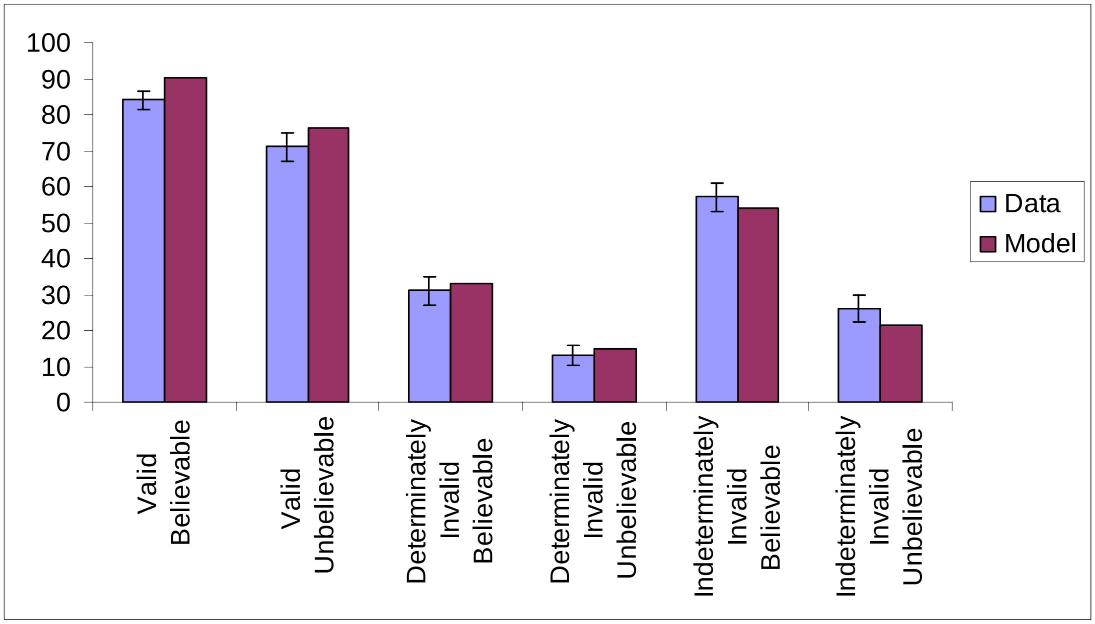


Figure 1: Comparison of model predictions with empirical data

This model has been compared with Roberts & Sykes's data and provides a good fit to these data ($R^2 = 0.985$). This provides support for the model and theoretical claims about the role of belief in relational reasoning.

Conclusion

The good fit of the model to the data supports the idea that belief influences reasoning by a form of source misattribution. That is, activation of belief chunks is increased during the reasoning process and this increases the chance of a belief being retrieved instead of a mental model. Not only is this a novel explanation, it is also a more parsimonious and well specified explanation than some dual process accounts of reasoning (e.g. Evans, Handley & Harper, 2001). Future work will test predictions of this model and extend it to other forms of reasoning.

References

Evans, J.St.B.T., Barston, J.L. & Pollard, P. (1983). On the conflict between logic and belief in syllogistic reasoning. *Memory and Cognition, 11,* 295-306.

Evans, J. St.B.T., Handley, S.J. & Harper, C.N.J. (2001). Necessity, possibility and belief: A study of syllogistic reasoning. *The Quarterly Journal of Experimental Psychology,54A,* 935-958.

Roberts, M.J. & Sykes, E.D.A. (2003). Belief bias and relational reasoning. *The Quarterly Journal of Experimental Psychology, 56A,* 131-154.

Cognitive Modeling of Web Search

Michael Schoelles, Wayne D. Gray, Vladislav Veksler,
Stephane Gamard, and Alex Grintsvayg

CogWorks Laboratory
Cognitive Science Department
Rensselaer Polytechnic Institute

A significant challenge for computational cognitive modeling is to develop high-fidelity models of web surfing. To successfully model the entire task both software and cognitive engineering problems must be solved. At ACT-R-2005, we addressed some of the software engineering challenges posed by the task of attaching an ACT-R model to a web browser (Gamard, Schoelles, Kofila, Veksler, & Gray, 2005). In this talk we focus on the cognitive engineering challenges posed by the need to navigate and search a near infinite number of heterogeneously designed web pages in pursuit of a weakly specified target.

PRIOR WORK

Our work builds on the pioneering efforts of others. The first effort to bring semantics into the search of an unbounded data source was SNIF-ACT (Pirolli & Fu, 2003). The SNIF-ACT model replaced ACT-R's expected utility function with one that was derived from the Rational Activation Theory (Anderson & Schooler, 1991) of declarative memory. Choice of actions was based on the activation spread to memory chunks based on similarity to the user's goal. Similarity was based on metrics derived from the Pointwise Mutual Information (Pirolli, 2005) measure of semantic distance (MSD).

An important class of models of web surfing are those based in the Construction-Integration Architecture (Kintsch, 1998). CoLiDeS (Kitajima, Blackmon, & Polson, 2000) claims that the perceived relevance of the Web page text or image to the goal determines what the users act on. Like SNIF-ACT, the similarity of the text to the goal is based on a MSD. In contrast to SNIF-ACT, CoLiDeS uses Latent Semantic Analysis (Dumais, 2003) as its MSD.

CoLiDeS+ (Juvina, Oostendorp, Karbor, & Pauw, 2005) extends CoLiDeS with the concept of *path adequacy,* which is a history of the similarities computed. This approach performs similar to humans in that it ends up at the same page; however, the model takes more steps. Juvina attributes the differences in decision making to the weakness of LSA. In particular, the "general reading " corpus was used. Juvina proposes that a more specialized semantic space would have given better results.

The SNIF-ACT and CoLiDeS work suffers from two issues. First, neither class of models performs a realistic search of a web page. Although SNIF-ACT is based on ACT-R, it did not use ACT-R's perceptual-motor capabilities. As far as we know, CoLiDeS has no perceptual-motor capability. Although the lack of perceptual-motor capabilities are a realistic simplification for an initial effort, it means that neither SNIF-ACT nor CoLiDeS can account for search time or search order as a function of the visual layout of a page. In other tasks, perceptual-motor costs defined by time have been shown to act as *soft constraints* which determines people's tendency to plan versus act (Fu & Gray, 2006). Small increments in perceptual-motor costs may lead to large tradeoffs between interaction-intensive and memory-intensive strategies (Gray, Sims, Fu, & Schoelles, 2006). If the visual layout of a page affects search order, it is also affecting search time. Hence, high-cognitive-fidelity models of web search will have to take account of the endogenous influence of visual features on search order.

Second, both SNIF-ACT and CoLiDeS used different MSDs to compute relatedness. It has been shown that all MSDs are not functionally equivalent (Kaur & Hornof, 2005). It is not clear to what extent which MSDs mimic human relatedness judgments (Veksler & Gray, 2006) for what web-based tasks.

CURRENT EFFORT

Realistic models of web search require a realistic accounting of the time required to search each new web page. Search time and the success of finding the most related target depends on how many prior items are visited and the semantic relatedness of those items to the searched for information. The order in which a new page is searched may be partially depended on exogenous features such as a tendency to search a new page from top-down and left-right. However, it also depends on endogenous influences of the visual design of a display. Hence, our research has turned to incorporating visual saliency metrics (Itti & Koch, 2001; Rosenholtz, 2001) into our models. Likewise, we have been impressed by the diversity of results returned by diverse measures of semantic distance (Kaur & Hornof, 2005). The problems of directly comparing results of various measures of semantic distance are very complex and require the development of new methodologies to compare various MSDs under various conditions (Veksler & Gray, 2006).

We are building ACT-R 6 models that incorporate both MSDs and visual saliency metrics. In contrast to SNIF-ACT and CoLiDeS+, we employ ACT-R's perceptual and motor processes to perceive and act on web pages. We feel it is essential to model the whole task, since human search is influenced by visual features of the task environment. The model has the capability to represent in the ACT-R's visual memory a web page and to access or calculate in real-time any one of 20 MSDs to assess the semantic relatedness of found text to a navigation goal.

Our model is a work-in-progress and during the talk we will present some of the problems we have encountered in web surfing that are easy for humans, but difficult for ACT-R. Some of these problems are software engineering issues, others relate to the theory and functioning of various modules, while others may inform central assumptions of the ACT-R architecture. In any case, we believe the challenged posed by the web is one that the modeling community must face. The ability to search a near-infinite source for information and to interact with heterogeneously designed web pages presents a significant challenge to the state-of-the-art in computational cognitive modeling.

References

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

Dumais, S. (2003). Data-driven approaches to information access. *Cognitive Science, 27*(3), 491-524.

Fu, W.-T., & Gray, W. D. (2006). Suboptimal tradeoffs in information seeking. *Cognitive Psychology, 52*(3), 195-242.

Gamard, S., Schoelles, M. J., Kofila, C., Veksler, V. D., & Gray, W. D. (2005, July 15-17). *CogWorks visualization architecture: Cognitively engineering next generation workstations for decision makers.* Paper presented at the ACTR-05 Workshop, Trieste, Italy.

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).

Itti, L., & Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews Neuroscience, 2*(3), 194-203.

Juvina, I., Oostendorp, v. H., Karbor, P., & Pauw, B. (2005). *Toward Modeling Contextual Information in Web Navigation.* Paper presented at the XXVII Annual Conference of the Cognitive Science Society, Stresa, Italy.

Kaur, I., & Hornof, A. J. (2005). A comparison of LSA, WordNet, and PMI-IR for predicting user click behavior. In *ACM CHI 2005 Conference on Human Factors in Computing Systems* (pp. 51-60). New York: ACM Press.

Kintsch, W. (1998). *Comprehension: A Paradigm for cognition*: Cambridge University Press.

Kitajima, M., Blackmon, M. H., & Polson, P. (2000). A Comprehension-based Model of Web Navigation and Its Application to Web Usability. *People and Computers, XIV*, 357-373.

Pirolli, P. (2005). Rational analyses of information foraging on the Web. *Cognitive Science, 29*(3), 343-373.

Pirolli, P., & Fu, W.-T. (2003). SNIF-ACT: A model of information foraging on the World Wide Web. *Lecture Notes in Computer Science, 2702*, 45-54.

Rosenholtz, R. (2001). Search asymmetries? What search asymmetries? *Perception & Psychophysics, 63*(3), 476-489.

Veksler, V. D., & Gray, W. D. (2006). Desiderata for comparing measures of semantic distance, *Poster published in the Proceedings of the 28th Annual Meeting of the Cognitive Science Society*. Vancouver, BC, Canada: Cognitive Science Society.

ATC in ACT-R: a Model of Aircraft Conflict Detection

Éric Raufaste[1]

[1] CNRS - Laboratoire Travail et Cognition
Université Toulouse - Le Mirail

2006 ACT-R Workshop

- Much of the Air Traffic Controller (ATC) task consists in maintaining a sufficient separation between aircraft
- In this study, conventional thresholds for minimal separation are 5 NM horizontal, and 1FL vertical
- Conflict: situation where two aircraft are at risk of an "air proximity" incident

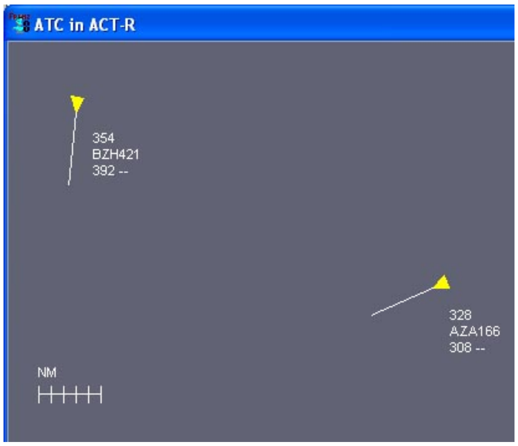
- The task : deciding whether a pair of aircraft is in conflict
- Manipulated factors
  - _ Altitude difference : same or different by at least 1 FL
  - _ Heading angles : 0°, 45°, 90°, 135°, 180°, 225°, 315°
  - _ Relative speeds : 0 vs. 10 to 50 knots
  - _ Miss distance : 2.5 vs. 7.5nm
- Controlled variables : Flight level and Speed

Mental workload minimization principle

Try to obtain the fastest decision possible, using the least effort

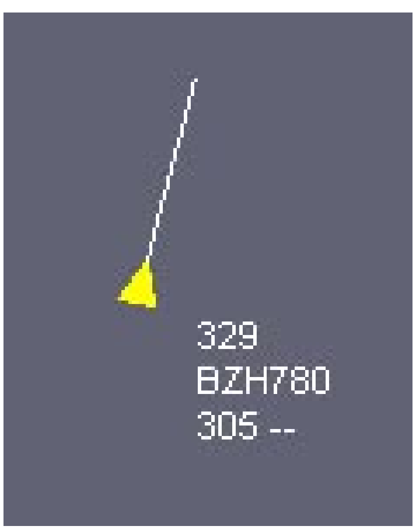
- It uses a lexicographic approach :
  1. Select a new criterion
  2. Apply the criterion
  3. If it is sufficient to decide, then decide and end
  4. Otherwise return to (1)

Headings



1. The model successively attends the aircraft and their associated speed vectors
2. The difference in headings is computed and stored
3. Further processing can this result as input
   - Diverging trajectories
   - Converging trajectories
   - Opposition
   - Pursuit

Altitudes

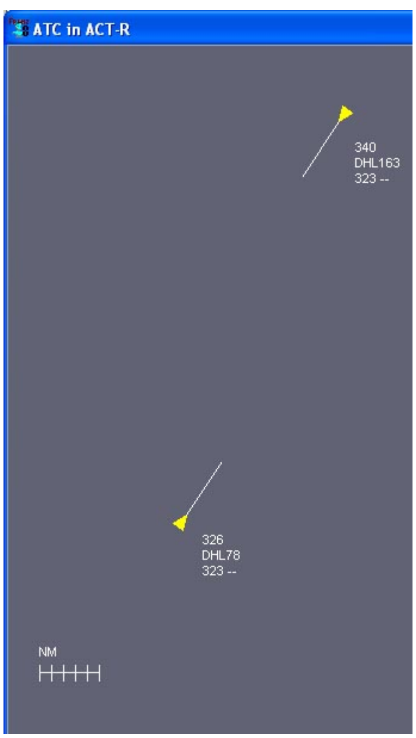


- Aircraft fly at constant altitudes then if altitudes differ by more than 1000 ft $\Longrightarrow$ No conflict
- Altitudes are only provided symbolically

Speeds

- Each aircraft speed appears under both symbolical and analogical forms
- Speed vector lengths are first accessed
- The difference in speed vector lengths is computed
- if the difference does not exceeds a perceptual threshold, numerical speeds are read

Lateral separation



1. Choose an aircraft
2. to be repeated until the target aircraft is reached:
    - a. Place a new mental anchor one speed vector farther
    - b. Move attention to this mental anchor
3. Draw a mental line between the last anchor and the target
4. Evaluate the size of the mental line

Initial processing rules

1. if headings diverge $\Longrightarrow$

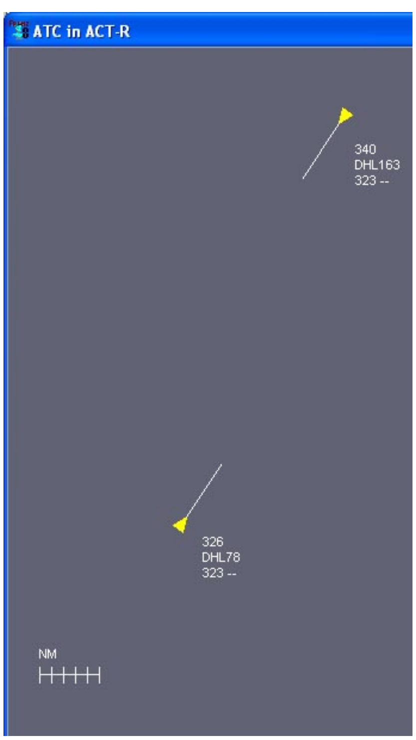
   Press [no conflict] ; End of trial

2. if altitude difference exceeds the minimal separation threshold (1000 ft) $\Longrightarrow$

   Press [no conflict] ; End of trial

3. Otherwise, continue processing

Opposition

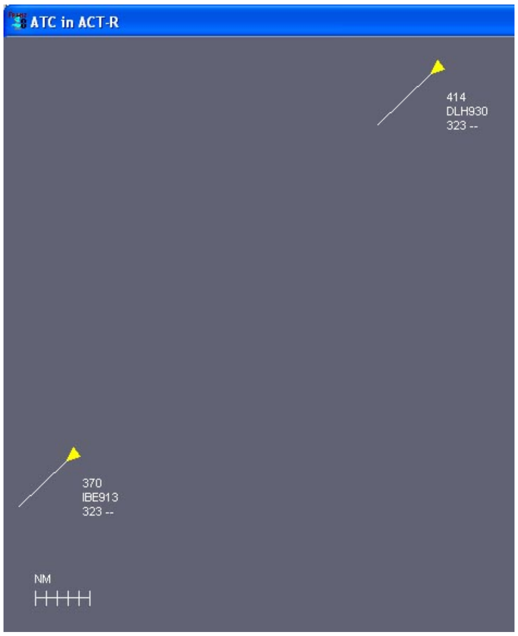


1. Get lateral distance
2. Get the width of the analogical scale ($= 5\,NM$)
   - 1. If lateral distance $\geq$ scale length

     Press [no conflict] ; End of trial

   - 2. If lateral distance > scale length

     Press [conflict] ; End of trial



1. Get speeds
2. If predator not faster than prey $\Longrightarrow$

   Press no conflict ; End of trial
3. Get lateral distance
4. Get the width of the analogical scale ($= 5NM$)
   - 1. If lateral distance $\geq$ scale length

     Press no conflict ; End of trial
   - 2. If lateral distance > scale length

     Press conflict ; End of trial

- In the current state of the model,
  - The average correlation with RTs from Rantanen and Nunes' participants and the model is better than .92
  - The average mean deviation is below 600 ms
  - But high error rates in humans remain to be explained on some conditions
- in the near future
  - Modeling the angle effect in convergent headings
  - Modeling finer details of the mental processes
  - Adding the vertical dimension (based on Averty, 2005)
  - Modeling individual differences (based on Stankovic, Raufaste, & Averty, 2006)
  - Modeling mental workload associated with those fine grain strategy variables

Modifying ACT-R for Visual Search of Complex Displays

Shawn Nicholson (shawn.nicholson@imts.us)
IMTS, 1000 Technology Dr, Suite 3220
Fairmont, WV 26554 USA

Michael D. Byrne (byrne@rice.edu)
Rice University, 6100 Main Street, MS-25
Houston, TX 77005 USA

Michael E. Fotta (mike.fotta@imts.us)
IMTS, 1000 Technology Dr, Suite 3220
Fairmont, WV 26554 USA

Abstract

Visual search is one of the more extensively studied areas in cognitive science. In the last few decades cognitive architectures have attempted to model visual search as a component of simulating human visual processing. Such attempts have usually focused on search based on a few object features, following most laboratory studies. However, applications of cognitive architectures are moving out of the laboratory and are being applied to complex displays that support real-world dynamic tasks. This necessitates the need to model more complex visual search tasks such as search with relational constraints between multiple objects in the scene. The current work addressed the need for guiding a search based on relationships by modifying the visual system of ACT-R.

Introduction

Over the last few decades a number of cognitive architectures (e.g., ACT-R, EPIC, Soar) have attempted to model human perceptual, cognitive and even motor interaction with an external environment, including the user interface of a computer system (for a review, see Byrne, 2003). In order for cognitive architectures to effectively model human interaction with a complex and dynamic display, the process of human visual search of the user interface must be realistically represented.

There has been modeling of visual search in such architectures, but this is usually constrained to searching for colors, for a particular position, a particular letter or number, etc. In these searches the target item is displayed among a group of distractor objects. This type of search is pre-attentive; properties of items in the scene, obtained before visual attention has been focused on the items, are enough to guide the search. The distractors can be differentiated from the target by a variable number of properties, the least complex being those where the search item can be distinguished on the basis of a single attribute (color, shape, orientation, etc). In this case the item is located with minimal visual processing. This is referred to as the "pop-out" effect (e.g., Triesman & Gelade, 1980).

However, as the number of features required to differentiate between objects in the scene increases, the complexity of the search process increases. In the most complex case the scene must be attended to serially, examining each object in turn in order to locate the target item. In between these extremes falls guided search, where there is a conjunctive set of features needed to distinguish the target item from the distractors (see, for example, Wolfe, 1994 for a theoretical perspective; see Fleetwood & Byrne, in press, for a more applied example).

Within guided search, in addition to the number of search criterion required for object differentiation, another source of complexity is the type of relationships expressed between scene object properties and search criterion. When executing a visual search, each search criterion specifies three things: 1) the relevant feature, 2) the value associated with that feature, and 3) the relationship desired to exist between the constraint value and the value objects in the scene have for that property. For example, when searching for a red object in a scene, there is an implied relationship between the constraint and the property value of the object in the scene. In particular, the target item in the scene is one which has a value for the *color* property that is equal to red; in this case, the relationship is one of equality. When searching for an object that is in the upper half of the scene, the relevant property is the spatial height coordinate, the value associated is the top half of the scene, and the desired relationship between the scene objects property value and the search constraint is the greater-than inequality relation. The simplest relations are those that compare an explicit value specified in the search constraint to fixed values of the scene objects. More complex relationships compare a property value of a scene object to something less straightforward, such as the relations *above* and *below*. These represent relative position relationships in which the spatial positions of objects in the scene are compared, not to some explicit constraint value, but to the positional values of other objects in the scene.

While there has been extensive research on visual search, the majority of experiments have involved modeling search with either fairly straightforward or minimal spatial constraints. These search constraints are the ones that specify that the target object in the scene should have a particular property value, typically independent of other objects in the scene (e.g. in the upper half of the display). The complexity of a search becomes higher when a target item is required to have a relational property; that is to have a particular property value that is related to another object in the scene (e.g. to be *above* or *beside* another object in the scene). As these relationships become more complex visual search becomes much more difficult to model.

While this problem is a general one for modeling human interaction with complex displays, a specific modeling framework is necessary to instantiate such models. The Human Error Modeling for Error Tolerant Systems (HEMETS) (Fotta, 2005) project is developing a software tool to assess the human errors likely to occur given a user interface for a system. HEMETS development uses and extends the ACT-R (Anderson, et al., 2004) cognitive modeling system which simulates human task performance. One of the major challenges in developing HEMETS is the modeling of the human interaction with a complex computer display. ACT-R contains a set of perceptual-motor (PM) modules which includes a visual system that allows modeling of this interaction to some extent. The visual system acts as an interface between the cognitive mechanisms of ACT-R and a simulation of any external environment, including computer displays. The visual system is responsible for maintaining information about what is in the visual environment. In performing visual search the cognitive system scans the visual scene and shifts the attention of the system to particular objects. However, the system has limitations in modeling human visual scanning which needed to be addressed in order to more realistically model visual search in HEMETS. This paper describes our current approach to address these limitations.

In order to prototype HEMETS a user interface from a simulation of air traffic change detection is being used. A screen shot of this simulation, the CHEX air warfare task (St.John, Smallman, & Manes., 2005), is shown in Figure 1.

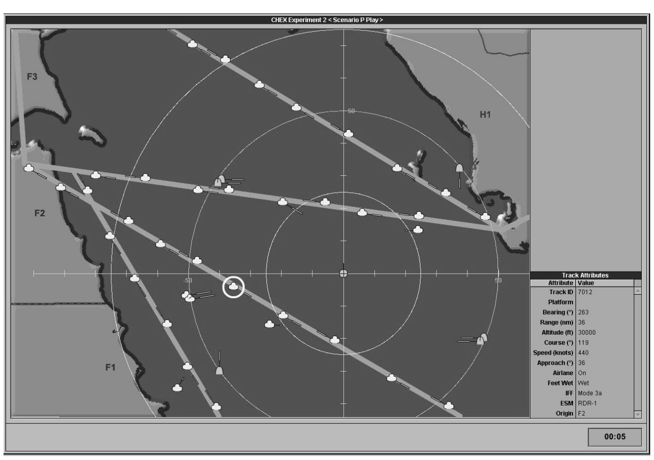


Figure 1: CHEX air warfare task simulation scene

This simulation mimics part of the user interface in a naval Combat Information Center. The operator's task with this interface is to detect certain changes in aircraft (the blob-like objects) that may constitute a threat to the user's ownship (cross-hatched object in center). Specific tasks include determining whether a particular object is an aircraft traveling along an air lane, leaving an air lane, turning inbound, or crossing a range ring.

Experience with this and other similarly designed interfaces led the developers of the CHEX air warfare task to conclude that operators approach this task by visually scanning around prominent features in the scene including the range rings (light rings in Figure 1) and the air lanes (solid straight lines). Thus, HEMETS must be able to model this type of scanning if it is to truly represent human interaction with this type of interface. In order to perform such scanning, the system needs to be able locate objects in the scene based on their relation to the prominent features. ACT-R, however, can not accommodate this type of visual search so modifications were needed to ACT-R's visual system.

We first explain the current visual system and its limitations in visual search and then discuss our approach to overcoming these limitations.

ACT-R Visual System

The ACT-R visual system (see Figure 2) is composed of two modules, the visual-location module and the visual-object module. The visual-location module is responsible for guiding visual search, locating objects in the scene that match a provided set of constraints. The visual-object module is responsible for shifting attention to, and extracting properties from an object at a given visual-location.

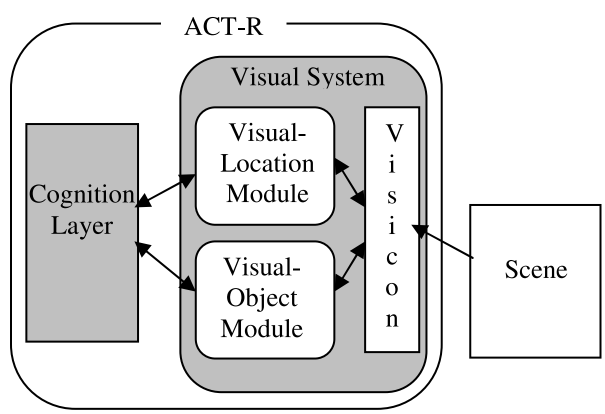


Figure 2: Visual System in ACT-R

Objects in a scene are represented in ACT-R by a set of features encoded in a list called the visicon (see Figure 3). These features describe where an object is, what kind of object it is, what color it is and so on. A request to the visual-location module is composed of a set of properties and values which describe an object to be located. For each of these, there is an expressed relationship that the objects in the scene must satisfy (e.g. the value of the specified property in the object in the scene must be greater-than the specified value). These property-value-relation triplets are used as a constraint set, filtering out objects from the visicon that do not satisfy the constraints. In the situation where more than one object in the scene satisfies all of the constraints, one is selected at random from the possible choices.

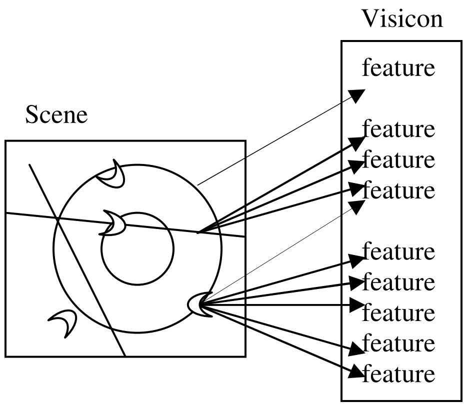


Figure 3: Visicon feature encoding

The visual location module currently supports two kinds of constraints: visual constraints and spatial constraints. Within these categories there is a fixed set of supported properties that can be used as constraints as well as a restricted set of relationships that can be required to exist between the target item properties and the scene object properties. Combinations of these constraints can be specified, under some restrictions, limiting the search to items matching the criterion, akin to guided preattentive visual search. Filtering objects this way allows the system to deal with pop-out effects; objects matching the particular constraints can be located rapidly, no matter how many other objects are cluttering the scene (e.g. locating a green X among a group of red Os).

Visual scanning is an extension to visual search where the goal is to perform repeated searches in an ordered fashion among multiple objects that satisfy a particular constraint set. ACT-R supports ordered scanning through a special constraint that indicates whether or not an object that has been searched for recently is a valid candidate. Thus, an object returned from a recent visual search request will not be returned on a subsequent search request, allowing the search to return the next item in the scan.

Current Visual Search in ACT-R

The visual attributes supported by this original system include: color, size, kind and value. The color attribute describes the color of the object. The size attribute restricts objects based on total area subtended (in degrees of visual angle). The kind attribute represents the classification of the object and value is a user-defined slot. The relationship that can be required to exist for the scene objects' property value is restricted to equality; the property value for the object in the scene must match exactly the specified constraint value (e.g. color must equal blue, size must equal 23, the kind must equal aircraft, etc).

The spatial attributes include the x and y coordinates of the object in the scene as well as the distance from the perceiver (depth distance). The relationships that can be specified for spatial constraints include equality (=) and inequalities (<, >, <=, >=), and the special relationship *nearest*. *Nearest* has the special property that it specifies that the scene object selected will be the one nearest the constraint value, it is a filter that is applied to the candidate scene objects that satisfy all of the other constraints.

The values for these spatial constraints are a scene coordinate, specified either explicitly (e.g. < screen-x 50, >= screen-y 100 would specify a valid scene object is one which has an x coordinate that is less than 50 and a y coordinate that is greater-than or equal to 100) or as references to a known visual-location such as the currently attended object (e.g. > screen-x CURRENT-X would specify that a valid scene object would be one whose x coordinate is greater-than the x coordinate of the currently attended object). There are also two other special values that can be used: *highest* and *lowest,* which, like *nearest*, are applied as filters to the scene objects that have already satisfied all of the other constraints. These select the object which has the highest (or lowest) appropriate property value (e.g. screen-x *highest* will select the scene object with the highest x coordinate that also satisfies all of the other constraints).

There are also limitations on how many constraints for a particular property can be expressed in a visual-location request. The *nearest* relationship, for example, can only be used once in a given request. The screen-x and screen-y properties are the only exception to this rule; they can be specified twice to represent the constraint that the position must fall within a specified range.

These property, value, and relationship specifications allow the visual location module to perform most visual searches with straightforward visual and spatial requirements. Using these, and one other type of constraint, the system can also perform a small selection of visual scanning routines.

Current Visual Scanning in ACT-R

When performing a visual search request, in the event that multiple objects satisfy the set of constraints, the response of the visual location module is to select randomly from among the candidates. If the same request is issued again, the system will return another random object (possibly the same one as before). In order to prevent the system from returning the same object repeatedly, allowing the system to iterate through each of the applicable scene objects, ACT-R has a third type of constraint: the *attended* constraint, which can be used to indicate that the system should return only those objects not returned as the result of a previous visual search request. Using this, it is possible to make repeated visual-location search requests and get each scene object in turn that satisfies the given constraints.

The sequence in which the valid scene objects (for a given set of constraints) are traversed is not specified. The system will skip randomly around the scene returning random valid scene objects, skipping over other valid scene objects. A more difficult task is to scan the scene in an ordered fashion, for example from left to right, top to bottom, not skipping any valid scene objects, but returning them in the sequence determined by the ordered scan.

The visual-location module supports these more advanced scanning tasks through a combination of the constraint properties described above. By using the *attended* constraint to prevent backtracking to objects already seen, and by specifying the *nearest* relationship to the last object seen in order to have the scan progress without skipping over valid objects in combination with other constraints it is possible to perform ordered scans. For example, using those constraints, and the additional constraint that the x coordinate needs to be greater than the last seen object, the visual search requests would return objects that progressed from left to right across the scene starting from the first returned object. In order to scan the scene entirely from left to right, top to bottom, it would first be necessary to issue a visual search request to locate the upper-left most object (by issuing a request using the *highest* and *lowest* keywords for the x and y locations), then scan from left to right by issuing a request using a constraint similar to the one above modified to also scan top to bottom.

HEMETS Modified Visual-Location Module

The original constraint specification system in ACT-R was designed to model simple experiments with fairly straightforward spatial requirements. In general, it has been adequate for such purposes. However, complex scenes requiring complex scanning strategies, such as the one depicted in Figure 1, requires more sophisticated specifications.

Modified Visual Search in HEMETS

Although the set of visual attributes supported (color, size, etc) allow the system to find most simple objects in a scene, as the complexity of those objects increases, it becomes necessary to use properties not currently available to distinguish between different objects. For example, in order to locate an object whose width was less than a certain value, it would be necessary to be able to pass in a constraint of the form: < *width 15*. This however, is not possible since width is not one of the fixed set of attributes currently useable as a constraint in the visual-location module. Additionally, the type of operators supported in the constraints is also fixed given a particular attribute type. For example, for color, constraints must specify that the color must equal some value (e.g. blue), so it is difficult to find an object whose color fell within a particular color range.

The first alteration to ACT-R's visual-location module, extending its visual search capabilities to handle more difficult visual scenes, was the addition of the ability for the visual-location module to use user-defined properties as search criterion. Four additional user-definable properties can be specified. This allows for searching for objects where the important visual attribute was not one of the standard set but specific to a particular kind of object. The addition of four usable properties is still a fixed, small number of properties about an object usable by the visual-location system during a search. As the number of visual attributes defining an object gets large, this solution fails as before. To address this issue, we further modified the system to accept an arbitrary object property as a comparison criterion; as long as the object has the particular attribute, it can be used as a feature criterion.

The next alteration to the visual-location module was to relax the restrictions on the relationships that can be specified for a particular type of constraint. The visual-location module required colors to be compared using symbol equality (= red, = green, etc), and the x and y screen positions to be compared using numerical equality and inequality functions (=, <, >, etc.) to some specific value (e.g. < *screen-x 100*) or to a currently attended location (e.g. < *screen-x CURRENT-X*), or by symbol equality to *highest* or *lowest* (e.g. = *screen-x HIGHEST*). These operations were fixed to support comparing specific object attributes. Rather than simply adding fixed relationships for all possible object properties that are used as search criterion, the system was modified to allow any operator to work on any valid input in a criterion. What is valid is determined by the operator (i.e. numbers for numeric inequality tests, etc). Additionally, support was added for the definition of custom methods for performing specialized comparisons between attributes that don't support the standard current comparison operators. For example, it is now possible to create a custom *color<* operator that takes two colors and indicates whether one is lower than the other on a color scale that can be used as a relationship in a visual-search request criterion.

Another side effect of these specialized operators and attributes was the removal of the limitation on the number of times a particular attribute could be used as a constraint in the same visual location request. The restriction that a particular property could be used only once as a constraint in a visual search request proved restrictive when attempting to do more complex scene scanning for a variety of reasons. For example, when attempting to perform an ordered search, it was necessary to use the *Nearest Current* criterion in order to progress across the scene in a fashion that did not skip over objects. Since the *Nearest* relationship can only be used once, it is not possible to perform a search that progresses in a spiral out from a particular object or location in the scene. To accomplish that, would require specifying *Nearest* *<spiral-location>* to keep the search on the path defined by the desired scan spiral in combination with *Nearest Current* in order to progress to the next nearest from the last attended object. With the generalization of the operator and object attribute constraint methods; this

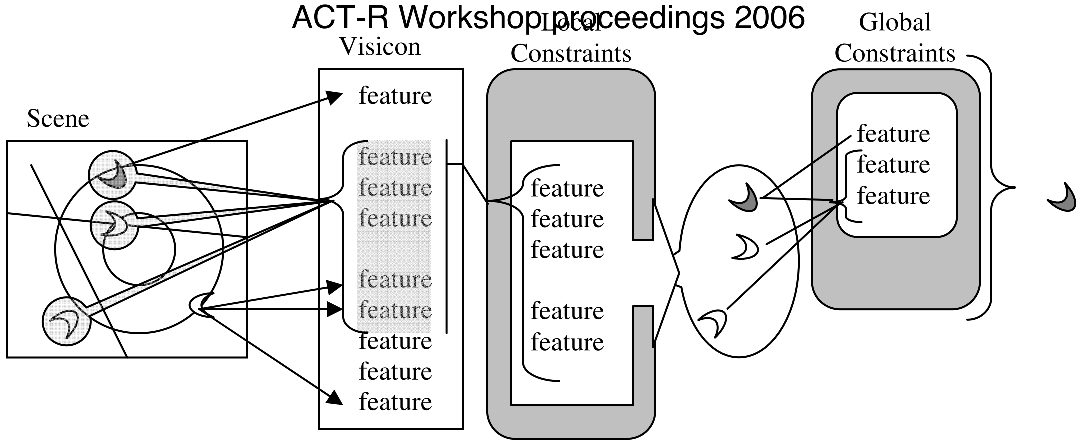


Figure 4: Visual-Location filtering process

restriction on the number of uses of a particular slot was removed.

There are two general classifications of search criteria shown in the current implementation. The *highest*, *lowest*, and *nearest* represent search constraints that are applied to the set of all objects in the scene that satisfy all of the other constraints in the visual-location search request. The other constraint properties and relations, such as color or position, apply to an individual object in the scene independent of what other objects in the scene satisfy any constraints. We have formalized this distinction by differentiating global and local constraints (see Figure 4). Local constraints are those constraints that are applied to each individual object in the scene that require only information specific to that object taken in isolation (e.g. testing whether the screen-x value of the object is > 100). The global constraints are then run over the collection of objects that satisfy the local constraints, filtering from among those (e.g. picking the one with the *highest* screen-x value, or that is *nearest* to a particular object or location). These constraints are specified on a per-model basis and all support custom relationships and any properties that objects in the scene might possess.

Additionally, when more than one object in the scene satisfies all of the property constraints, rather than selecting randomly among them, it is possible for the model author to define a method for selecting from among the candidate objects.

Modified Visual Scanning in HEMETS

Two general scanning methods which were significant in our current air traffic change detection task were enabled by these changes. The first is being able to direct an ordered scan based on the physical attributes of another object in the scene (e.g. in our task, following along an air lane in the scene looking for objects along that air lane). To accomplish this we define a special operator that is similar to Nearest in nature, called Nearest-Along. This operator takes an object and an allowable distance. The object is the item in the scene (the air lane in our task) to scan along with the allowable distance representing how far away from the object it is ok to be to satisfy the constraint. This new special relationship allows a visual-location search request to consider a scene object's location to another object in the scene as a relevant search criterion. This relationship used in conjunction with the normal Nearest operator to progress from one object to the next nearest without skipping objects allows us to follow along an air lane locating aircraft.

The second common task in our air traffic change detection scenario is scanning around the range rings. This generalizes to scanning in a ring around a fixed central location. Under the previous implementation it would be necessary for every aircraft to store its distance from a specific location as an object attribute. This required at the very least the extension of usable attributes in visual-location search requests. Even when successfully implemented, this approach could not easily determine when the ring was completely circled. With the new implementation, we define a specialized operator *Around* with a radius argument. It is then possible to issue a visual-location search request that circles the central point at the specified radius and terminates when the ring is completely traversed.

Summary

Visual search has usually been studied in the laboratory and modeled considering only the difference between intrinsic object features (e.g., color, position, orientation, size). The use of relationships between objects to guide visual search has received much less research attention. Modeling of relationships for use in visual search has been previously accomplished in some cognitive simulations but only at a simple level. For example, ACT-R can search for an object nearest the current object being attended to.

Our current work enables the modeling of more complex relationships to guide visual search. The combination of enabling specialized operators, attributes, local and global constraints, and using object attributes more than once permits custom search methods that model scanning the scene in complex patterns. Although the current development was driven by the necessity to model scanning for a particular interface, the techniques developed can be applied to a wide range of user interfaces or simulated environments.

The extensions enable modeling of some of the more complex aspects of visual search as performed outside the laboratory. Thus, models built using these techniques can be studied in a variety of settings in order to extend our knowledge of human visual search.

Acknowledgments

This work was supported by an Office of Naval Research Phase II Small Business Innovative Research grant for the Human Error Modeling for Error Tolerant Systems (HEMETS) contract, No. N00014-05-C-0140.

References

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

Byrne, M.D. (2003). Cognitive Architecture. In J.A. Jacko and A. Sears (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 97 - 117). Mahwah, NJ: Lawrence Erlbaum Associates.

Fleetwood, M. D. & Byrne, M. D. (in press). Modeling the Visual Search of Displays: A Revised ACT-R/PM Model of Icon Search Based on Eye Tracking Data. *Human Computer Interaction*.

Fotta, M.E. (2005). Human Error Modeling for Error Tolerant Systems (HEMETS): Upcoming Work. *ONR Workshop on Attention, Perception and Modeling for Complex Displays*, May 10 -12.

Starreveld, P.A., Theeuwes, J., & Mortier, K. (2004). Response Selection in Visual Search: The Influence of Response Compatibility of Nontargets. *Journal of Experimental Psychology: Human Perception and Performance*, 30 (1), 56–78.

St. John, M., (2005), personal communication, June 29.

St. John, M., Smallman, H. S., & Manes, D. I. (2005). Recovery from interruptions to a dynamic monitoring task: The beguiling utility of instant replay. In *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting*. Santa Monica, CA: Human Factors and Ergonomics Society.

Triesman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology, 12*, 97–136.

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review, 1(2)*, 202-238.

Human Error Modeling Architecture Systems Interface (HEMA SI)

Talk Outline

- SegMan
- CHEX Air Warfare Task
- Object Recognition Issues
  - Scene Elements
- ACT-R Integration Issues
  - Performance
  - Persistency
- Relation/Benefits/Accomplishments to overall task

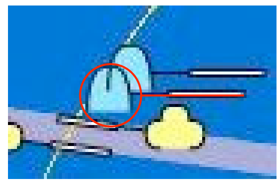
Segmentation/Manipulation (SegMan)

- Serves as an intermediary between an environment (visual scene) and the cognitive modeling system (ACT-R).
- Processes visual scene in any graphical user interface
  - Groups pixels by color and identifies pixel-groups via patterns
  - Identifies both objects and text

Object Recognition Issues

- Speed/Direction indicators
  - Associating with proper aircraft
- Air Lanes
  - Compensating for occlusion/segmentation by aircraft
- Range Rings
  - Recognizing thin-line/occluded rings
- Aircraft Occlusion
  - Compensating for occlusion by other aircraft

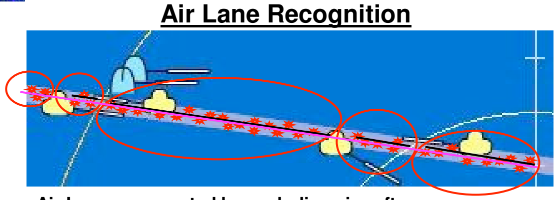
Speed/Direction Indicators Association



- Incorrect association of direction indicators to aircraft
- Solution: 2-stage algorithm
  1. Circle aircraft looking for potential indicator connection
  2. Trace indicator, ensuring it projects directly away from the aircraft

ACT-R Workshop July 21 - 23, 2006 IMTS 5

Air Lane Recognition

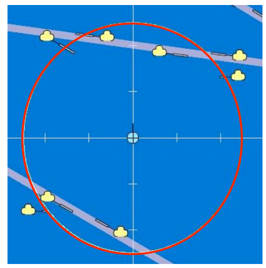


- Air Lanes segmented by occluding aircraft
  - Segmentation changes over time (moving aircraft)
- Solution: Line projection algorithm
  1. Fill Algorithm - identifies each segment point
  2. Linear Least-Squared line fitting – compute line equations for each segment
  3. Line Projection – merge lines to form a single line object

ACT-R Workshop July 21 - 23, 2006 IMTS 6

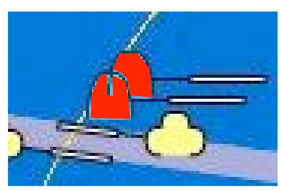
Range Ring Recognition

- **Normal fill method fails**
  - Rings are thin (~1 pixel)
  - Primarily diagonal lines
  - **Solution: Leaky Fill Algorithm**
    - Allows following diagonal points
- **Rings segmented by occluding aircraft**
  - **Solution: Radius connection**
    - Multiple segments of ring joined based on radius.



Aircraft Occlusion

- **Occlusion by other aircraft**
  - **Causes failure to locate aircraft**
- **Solution: Fill Algorithm**
  - **Fill algorithm that starts from original discovered location and expands outward to the objects edge**



ACT-R Integration Issues

- Set up SegMan as a device in ACT-R
- Allows translation of scene into ACT-R environment representation (Visicon)
- Issues for ACT-R visual scanning
  - Scene scanning time
  - Object persistency over time

Optimization of Multiple Scene Scans

- Rescanning scene entirely repeatedly is too slow
- Solution: Caching of static items
  - Initial scan (~4sec)
    - Identifies and caches static items (air lanes, range rings, distance hash lines, land masses)
    - Identifies moving items (aircraft)
  - Subsequent scans (~0.25sec)
    - Identifies moving items (aircraft)

Object Time-Persistency

- Each scan of the scene is independent of previous
  - No information connecting one blob (aircraft) to another in two scenes
- Solution: Position-Based Correlation
  - Aircraft move slowly
  - Two objects in close proximity to each other in two scenes are identified as the same object (aircraft)
- Generalization for faster moving scenes
  - Add direction/speed analysis to correlate two objects

Relation, Benefits, Accomplishments

- High Performance visual scene processing allowing for real-time cognitive modeling
- Visual scene processing can connect arbitrary air space control interfaces to ACT-R
- These problems can be generalized to many other interfaces, reducing the time required for future deployments of HEMETS

PENN STATE

Rensselaer Cognitive Science

ACT-R Workshop 2006

IST

8899
- 7
8892
- 7
8885
- 7
8878
- 7
8871
- 7
8864
- 7
8857
- 7
8850
- 7
8843
- 7
8836

Cognitive Aspects of Serial Subtraction

Frank E. Ritter, Sue Kase, Jeanette Bennett, Laura Klein,
& Mike Schoelles

IST, BBH @Penn State, RPI

http://acs.ist.psu.edu/papers/

Part of understanding the effects of stress and caffeine on cognition

Human data obtained from an empirical study utilizing the serial subtraction task, part of the TSS Task

Analyses yield an understanding of problem types that generated errors and individual differences in subtraction rates

ACT-R 6 model of process and of potential stress theories

This project was supported by the ONR, N00014-02-1-0021, and NIH through GCRC grant MO1-RR-10732. The views expressed in this article do not necessarily reflect the positions or the policies of the U.S. Government, and no official endorsement should be inferred.

PENN STATE

Rensselaer Cognitive Science

ACT-R Workshop 2006

IST

8836
- 7
8829
- 7
8822
- 7
8815
- 7
8808
- 7
8801
- 7
8794
- 7
8787
- 7
8780
- 7
8773

Methodology Overview

15 male subjects performing serial subtraction task

4 blocks (4 minutes each) subtracting 4-digit numbers

1. subtract by 7
2. subtract by 13
3. brief task change – mathematical word problems
4. subtract by 7
5. subtract by 13

Subject reports answers verbally to experimenter

- reprimanded when answers are incorrect
- 'harassed' by being asked to go faster

Audio recording of subject performance

- Errors, pace (within and across blocks), variance, IDs

Pre- and post-task appraisals

Mathematics anxiety surveys (sMARS, CMAQ)

Mood Assessment Scale (self-reported stress)

Rensselaer Cognitive Science — ACT-R Workshop 2006

PENN STATE

Subtracting by 7s – frequency of errors

Subjects made 1511 subtraction attempts, 1346 correct responses, 165 errors – 10.9% error

107 (7.1%) errors resulting in subtractions other than -7

Less than -7			Correct -7	Greater than -7		
Value	Frequency	Percent	Frequency Percent	Value	Frequency	Percent
-6807	1	0.1	Total 1511	-6	25	1.7
-6007	1	0.1	Correct 1346 89.1%	-5	18	1.2
-1007	1	0.1	Error 165 10.9%	-4	1	0.1
-907	4	0.3	Duplicates 2 0.1%	-3	2	0.1
-207	4	0.3	Additions 56 3.7%	1	3	0.2
-114	1	0.1		2	1	0.1
-17	7	0.5		3	12	0.8
-16	1	0.1		5	2	0.1
-14	2	0.1		13	3	0.2
-13	4	0.3		20	1	0.1
-10	4	0.3		76	1	0.1
-9	4	0.3		83	2	0.1
-8	27	1.8		93	12	0.8
Total	61	4.4		100	1	0.1
				293	2	0.1
				693	2	0.1
				893	5	0.3
				983	1	0.1
				991	1	0.1
				992	2	0.1
				993	2	0.1
				1003	1	0.1
				1793	2	0.1
				Total	102	6.8

Errors resulting in additions

56 (3.7%) errors resulting in additions

Rensselaer Cognitive Science — ACT-R Workshop 2006

PENN STATE

Subtracting by 7s: errors -8, -6, -5

The most common errors were misremembering to subtract 7; instead, subtracting by 8, 6, or 5

Misremembering 7		
Value	Frequency	Percent
-8	27	1.8
-6	25	1.7
-5	18	1.2

Subtract by 5 error: the digit in the 1s column is a 1, 2, 3, or 4.

A one was in the 1s column 66.7% of the time; a two 22.2% of the time

All problems required a carry-over from 10s.

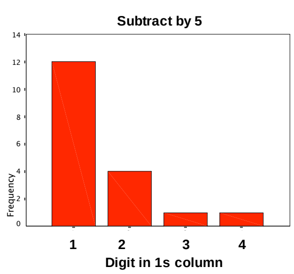
Subtract by 6 error: a 3 was the 1s column digit 52% of the time.

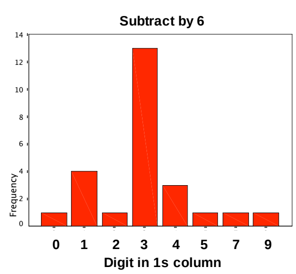
8% of the problems did not require a carry-over from 10s.

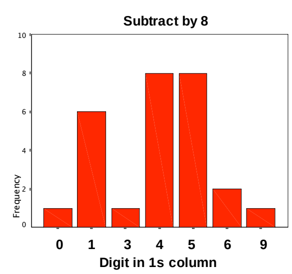
Subtract by 8 error: a 1, 4, or 5 appeared in the 1s column a total of 81.4% of the time.

Digit	Freq
4	29.6%
5	29.6%
1	22.2%

3.5% of the problems did not require a carry-over from 10s.







Subtracting by 7s: error +93

Misremembering the 100s column, not effected by a carry, by incrementing it by 1, results in adding 93.

Misremembering 100s		
Value	Frequency	Percent
93	12	0.8

- The erroneous increment of the 100s column is either from 7 to 8, or 8 to 9.
- The incorrectly answered problems tend to fall into 3-problem sequences.

Subtraction problems causing addition by 93 error

Frequency missed:	[2]	[2]		[2]	[3]		
	8899	8892	8885	8794	8787	8780	8759
	- 7	- 7	- 7	- 7	- 7	- 7	- 7
	8892	8885	8878	8787	8780	8773	8752
	$\underbrace{}_{\text{In sequence}}$			$\underbrace{}_{\text{In sequence}}$			
Error:	8992	8985	8978	8887	8880	8873	8852

Subtracting by 7s: error +893

If 1,000s and 100s column values are 8 and 9, and then transposed as 9 and 8, the result is an addition of 893.

Misremembering and Transposing 100s and 1,000s		
Value	Frequency	Percent
893	5	0.3

- This error also occurred when the 100s column value is 1 and a carry-over is required; subjects mistakenly reported 9 instead of 0.

Subtraction problems causing addition by 893 error

Frequency missed:	[2]			
	8983	8976	8962	8101
	- 7	- 7	- 7	- 7
	8976	8969	8955	8094
	$\underbrace{}_{\text{In sequence}}$			
Error:	9876	9869	9855	8994

PENN STATE

Rensselaer Cognitive Science

ACT-R Workshop 2006

Subtracting by 7s: errors -17 and +3

When the value in the 10s column is mistakenly decremented by 1, after a correct subtraction of the 1s column, the result is subtracting by 17 instead of 7.

If the 10s column is mistakenly incremented by 1, after a correct subtraction of the 1s column, the result is an addition of 3.

-17 error: Only 2 of the 7 problems involved a carry-over from the 10s column

Misremembering 10s		
Value	Frequency	Percent
-17	7	0.5
3	12	0.8

+3 error: 10 out of 12 problems involved a carry-over from 10s column. This could mean the decrement of the carry is forgotten.

In 3 of the 10 problems with the carry, the forgotten carry occurs when 9 is not decremented to 8.

Problems causing -17 error

Frequency missed: [2]

	9039	8948	8787
	- 7	- 7	- 7
	9032	8941	8780
Error:	9022	8931	8770
	8157	7996	7674
	- 7	- 7	- 7
	8150	7989	7667
Error:	8140	7979	7657

Some problems causing +3 error

	9046	8969	8934	8913	8871
	- 7	- 7	- 7	- 7	- 7
	9039	8962	8927	8906	8864
Error:	9049	8972	8937	8916	8874
	8864	8773	8094	7940	7891
	- 7	- 7	- 7	- 7	- 7
	8857	8766	8087	7933	7884
Error:	8867	8776	8097	7943	7894

PENN STATE

Rensselaer Cognitive Science

ACT-R Workshop 2006

Individual Differences in Subtraction Rates

Cluster analysis conducted using Ward's method of minimum-variance clustering and squared Euclidean distance as the distance metric

Variables:

- Subtraction by 7s, total attempts (2 sessions)
- Subtraction by 7s, total errors (2 sessions)
- Pre-task appraisal, sum
- Post-task appraisal, sum
- Self-reported stress (from Mood Assessment Scale), sum of 4 reports
- Mathematics anxiety (from CMAQ), total score

Clustering identified two groups of subjects

Evidenced by individual pace plots characterizing two primary levels of subtraction performance

PENN STATE

Rensselaer Cognitive Science

ACT-R Workshop 2006

ID Example: Group 1

Green dashed line - subject told to go faster at session mid-point

Red bars - subtraction errors

Group 1 mean

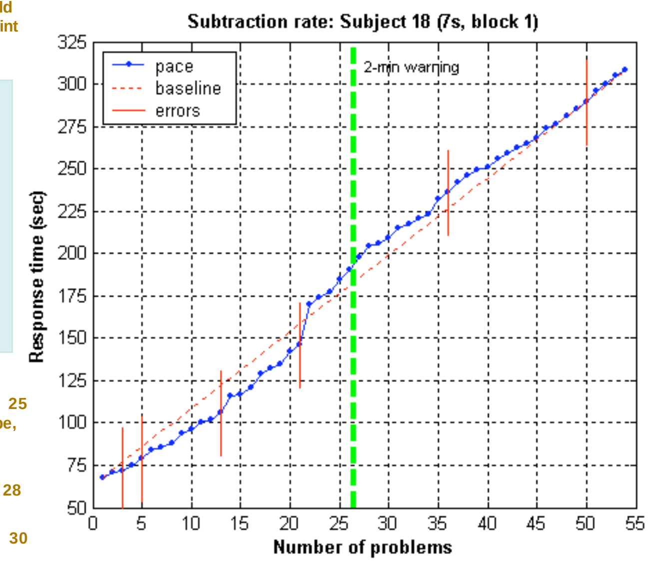
Total attempts	130.8
Total errors	9.7
Appraisal	
Pre-task	9.4
Post-task	13.2
Self-report stress	8.1
CMAQ math anx.	9.8

Instrument ranges

Task appraisal min 5 max 25 (25=very stressful, cannot cope, demanding task, threatening, poor performance)

Self-rep stress min 4 max 28 (28=very much stressed)

CMAQ min 6 max 30 (30=math anxious)



PENN STATE

Rensselaer Cognitive Science

ACT-R Workshop 2006

ID Example: Group 2

Green dashed line - subject told to go faster at session mid-point

Red bars - subtraction errors

Group 2 mean

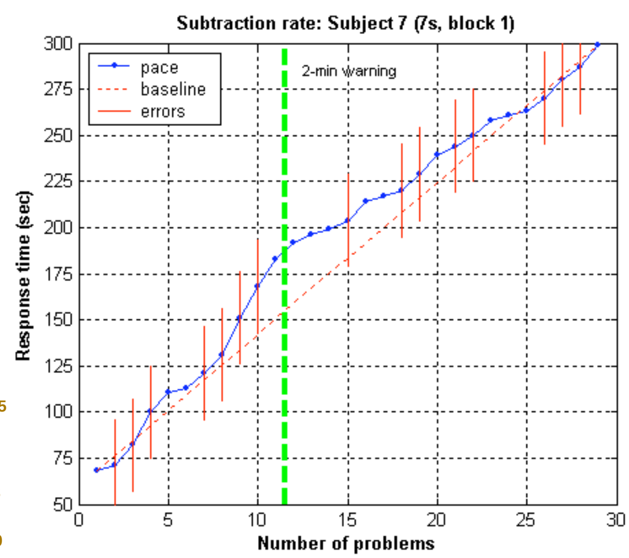
Total attempts	83.5
Total errors	21.5
Appraisal	
Pre-task	11.3
Post-task	16.3
Self-report stress	9.5
CMAQ math anx.	14.8

Instrument ranges

Task appraisal min 5 max 25 (25=very stressful, cannot cope, demanding task, threatening, poor performance)

Self-rep stress min 4 max 28 (28=very much stressed)

CMAQ min 6 max 30 (30=math anxious)



PENN STATE

Rensselaer Cognitive Science

ACT-R Workshop 2006

ACT-R 6.0 Model

Non-negative subtraction facts

Sub-goal for borrow, initiated when subtraction fact retrieval fails

Speaks answer in one of three ways - digit-by-digit, entire number, two halves

Trial time is mostly time to speak answer

No overlap of retrieval and speaking in version 1

ACT-R 6.0 features

- Problem represented in the imaginal buffer
- Variable slot names for current column and minuend (kept in goal buffer)
- New declarative memory element created for each subtraction

PENN STATE

Rensselaer Cognitive Science

ACT-R Workshop 2006

Conclusions

8773
\- 7
8766
\- 7
8759
\- 7
8752
\- 7
8745
\- 7
8738
\- 7
8731
\- 7
8724
\- 7
8717
\- 7
8710

Have a detailed data set of cognition (and physiology) under stress

Pace suggests learning is important

The details suggest several and different changes to subtraction process and mechanisms

Model is in hand, and overlays to model are also in hand

The overlays do not appear to support these changes, but formal comparisons can start shortly

Modeling Emotion in ACT-R

Robert L. West (robert_west@carleton.ca)
Institute of Cognitive Science, Department of Psychology Carleton University
Ottawa, Ontario, Canada, K1S 5B6

Terrence C. Stewart (terry@ccmlab.ca)
Institute of Cognitive Science, Carleton University
Ottawa, Ontario, Canada, K1S 5B6

Bruno Emond (bruno_emond@uqah.uquebec.ca)
Institute for Information technology
National Research Council Canada, Ottawa, Canada

In this ongoing project we are exploring how to represent emotion in ACT-R. However, rather than starting by modeling a specific experimental finding, our approach has been to first create emotional structures in ACT-R such that it can qualitatively model a wide variety of emotional effects. The goal is to create a single emotional system that can then be tested by modeling the many diverse experimental results related to emotion, without changing the way emotion is represented in ACT-R. To create these structures we used buffers and production system modules that run in parallel with the ACT-R procedural module. The production systems we used were identical to the ACT-R procedural production system except for parameter values. This approach is consistent with viewing ACT-R as a general framework for understanding the modular nature of the mind (Stewart & West, 2006). The first issue we faced was that emotions are often triggered by bottom up attention to an object in the environment. To deal with this we created a visual production system that scans the environment whenever top down commands are not being issued by the procedural module. We also created an emotional production system to represent the activity of the amygdala in terms of identifying threat or reward. Consistent with neurological findings, the emotional production system fires based on the contents of the visual buffer and has a faster firing time than procedural productions. The emotion module exerts influence on the procedural module in two ways, (1) by placing chunks representing emotional states into an emotion buffer that the procedural module has access to, and (2) by spreading activation into the declarative memory system, thus influencing the production module retrieval results (similar to Stocco & Fum, 2005). Likewise the procedural module can influence the emotional module by altering what is in the visual buffer, the goal buffer and the imaginal buffer.

References

Stewart, T.C. and West, R. L. (2006) Deconstructing ACT-R. Seventh International Conference on Cognitive Modeling. Trieste

Stocco, A., & Fum, D., (2005). From emotion to memory: An ACT-R view on the Somatic Marker Hypothesis. *Twelfth ACT-R Workshop*, Trieste, July 15-17, 2005.

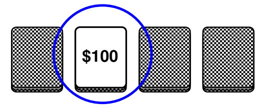
Laboratorio di Sistemi Cognitivi

Expected values and loss frequencies:

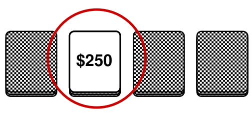
A new view on the choice process in the Iowa Gambling Task

Danilo Fum
Università di Trieste

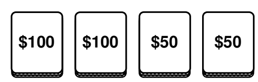
The Iowa Gambling Task



The Iowa Gambling Task



The Iowa Gambling Task

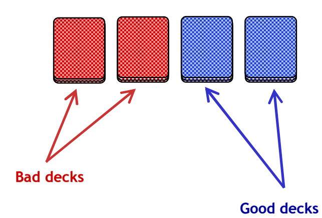


The Iowa Gambling Task

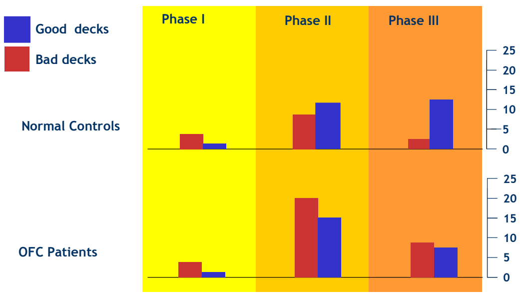
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A (+100)			-150		-300		-200		-250	-350		-350		-250	-200		-300	-150		
B (+100)									-1250					-1250						
C (+50)			-50		-50		-50		-50	-50		-25	-75				-25	-75		-50
D (+50)										-250										-250

	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
A (+100)		-300		-350		-200	-250	-150			-350	-200	-250				-150	-200		
B (+100)	-1250											-1250								
C (+50)				-50	-25	-50			-75	-50				-25	-25		-75		-50	-75
D (+50)									-250						-250					

The Iowa Gambling Task



The Iowa Gambling Task



The best kept secrets in the IGT

Secret #1:

Many normal participants give performances similar to OFC patients !!!

Usual performance measure: Δ = Good deck - Bad decks

Pilot A:	N = 74	36 partecipants with Δ < 10	27 with Δ <= 0
Pilot B:	N = 30	18 partecipants with Δ < 10	13 with Δ <= 0

The best kept secrets in the IGT (part two)

Secret # 2:

There are differences between decks with the same long-term expected value but different short-term contingencies.

	A	B	C	D
Pilot A	15.26	27.62	21.00	36.10
Pilot B	17.23	29.50	21.03	32.23
Fernie & Tunney (2006)	18.80	31.60	21.05	28.55

Hypothesis

The performance in the IGT is governed not only by the expected value.

In addition to the magnitude of the outcomes, a critical role is played by their frequency, too.

Experiment #1

A between factor (frequency of losses) with two levels:

HIGH: high number of losses (6 out of 10)
LOW: low number of losses (2 out of 10)

The expected values of the decks in the two conditions are the same.

Experiment #1

A within factor (feedback) with two levels:

DOUBLE: decks A and C provide the usual feedback (wins + losses)
SINGLE: decks B and D provide only the algebraic sum (win only or loss only).

Experiment #1

HIGH							
A		B		C		D	
90	-125	-35	0	30	-40	-10	0
100	0	100	0	20	-25	-5	0
80	-250	-170	0	40	-50	-10	0
110	0	110	0	30	-40	-10	0
100	-300	-200	0	90	0	90	0
110	-125	-15	0	70	0	70	0
120	0	120	0	90	0	90	0
120	-250	-130	0	40	-60	-20	0
90	0	90	0	20	-35	-15	0
80	-200	-120	0	70	0	70	0

LOW							
A		B		C		D	
90	0	90	0	30	0	30	0
100	0	100	0	20	0	20	0
80	0	80	0	40	0	40	0
110	0	110	0	30	-130	-100	0
100	0	100	0	90	-120	-30	0
110	0	110	0	70	0	70	0
120	-650	-530	0	90	0	90	0
120	0	120	0	40	0	40	0
90	0	90	0	20	0	20	0
80	-600	-520	0	70	0	70	0

$$\left.\begin{array}{l}\text{Expected value good decks (C \& D) = 250}\\ \text{Expected value bad decks (A \& D) = -250}\end{array}\right\} \text{for both the conditions}$$

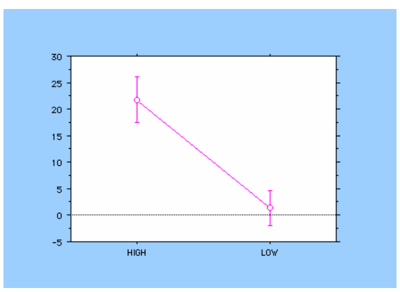
LabSiCo

Experiment #1: Results

- The frequency of losses affects the performance—measured as difference between good and bad decks—of participants in the IGT.

LabSiCo

Experiment #1: Results



LabSiCo

Experiment #1: Results

- The frequency of losses affects the performance—measured as difference between good and bad decks—of participants in the IGT.
- In the LOW losses condition, the participants are unable to discriminate between good and bad decks

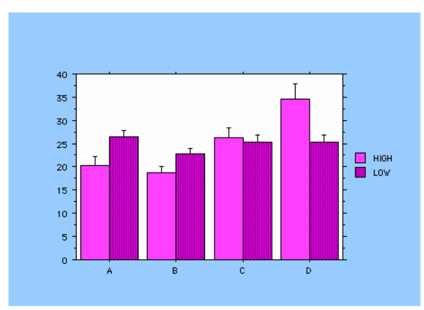
LabSiCo

Experiment #1: Results

- The frequency of losses affects the performance—measured as difference between good and bad decks—of participants in the IGT.
- In the LOW losses condition, the participants are unable to discriminate between good and bad decks
... and probably they have not VMPFC lesions!

LabSiCo

Experiment #1: Results



LabSiCo

Experiment #1: Results

- The frequency of losses affects the performance—measured as difference between good and bad decks—of participants in the IGT.
- In the LOW losses condition, the participants are unable to discriminate between good and bad decks
... and probably they have not VMPFC lesions!
- No feedback effect: the number of choices from “win & losses” decks are similar to choices from “win only/ loss only” decks.

Experiment #1: Results

- The frequency of losses affects the performance—measured as difference between good and bad decks—of participants in the IGT.
- In the LOW losses condition, the participants seem unable to discriminate between good and bad decks.
- No feedback effect: the number of choices from “win & losses” decks are similar to choices from “win only/ loss only” decks.

Experiment #1

HIGH							
A		B		C		D	
90	-125	-35	0	30	-40	-10	0
100	0	100	0	20	-25	-5	0
80	-250	-170	0	40	-50	-10	0
110	0	110	0	30	-40	-10	0
100	-300	-200	0	90	0	90	0
110	-125	-15	0	70	0	70	0
120	0	120	0	90	0	90	0
120	-250	-130	0	40	-60	-20	0
90	0	90	0	20	-35	-15	0
80	-200	-120	0	70	0	70	0

LOW							
A		B		C		D	
90	0	90	0	30	0	30	0
100	0	100	0	20	0	20	0
80	0	80	0	40	0	40	0
110	0	110	0	30	-130	-100	0
100	0	100	0	90	-120	-30	0
110	0	110	0	70	0	70	0
120	-650	-530	0	90	0	90	0
120	0	120	0	40	0	40	0
90	0	90	0	20	0	20	0
80	-600	-520	0	70	0	70	0

$$\left.\begin{array}{l}\text{Expected value good decks (C \& D) = 250}\\ \text{Expected value bad decks (A \& D) = -250}\end{array}\right\}\ \text{for both the conditions}$$

Experiment #2

In the previous experiment all losses were associated with negative outcomes (i.e. after a trial with a loss you have less money then before).

It is possible to modify the payoff matrix to distinguish between “fatal” and “non-fatal” losses.

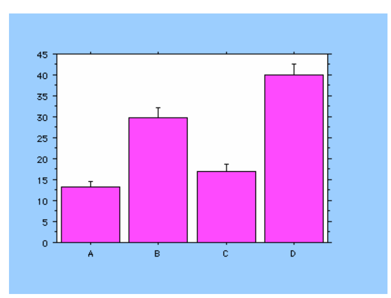
Experiment #2

MIXED							
A		B		C		D	
90	-55	35	0	30	-10	20	0
100	-860	-760	0	20	-10	10	0
80	-75	5	0	40	-30	10	0
110	0	110	0	30	0	30	0
100	-75	25	0	90	-30	60	0
110	-55	55	0	70	-20	50	0
120	-65	55	0	90	-20	70	0
120	0	120	0	40	0	40	0
90	-65	25	0	20	0	20	0
80	0	80	0	70	-130	-60	0

Lots of losses but very few of them are “fatal”
In the single feedback decks (B & D) only fatal losses are reported.

Note: the expected values of the decks are similar to those of Experiment #1.

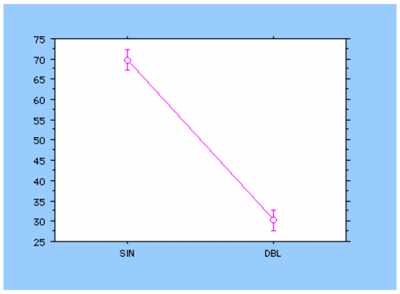
Experiment #2: Results



Experiment #2: Results

- Participants are able to discriminate between good and bad decks.
- Participants concentrate their choices on the single feedback decks (which have a very low number of losses).

Experiment #2: Results

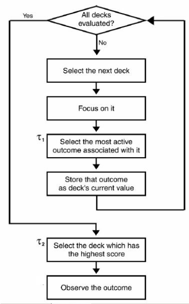


The underachievers

		Δ < 10	Δ <= 0
HIGH	N = 38	14	9
LOW	N = 38	24	18
MIX	N = 38	19	15

The previous model

The control flow



Fum & Stocco (2004)
Stocco, Fum, & Zalla (2005)

The previous model

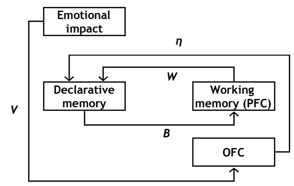
The new activation equation:

$$A_o = B_o + W_d S_{do} + \eta V_o$$

with

$$V_o = \log(O) / \log(\max(O))$$

The previous model



Fum & Stocco (2004)
Stocco, Fum, & Zalla (2005)

The bad news

The previous model is unable to replicate the behavior of our participants, even at the level of the differences between good and bad decks.

(Results too embarassing to be shown here!)

Looking for something else

Busemeyer and Stout's (2002) expectancy-valence learning model:

- People integrate the gains and losses experienced on each trial into a single affective reaction called a valence.
- Expectancies about the valence produced by each deck are learned by an adaptive learning mechanism.
- These expectancies serve as the inputs into a probabilistic choice mechanism that selects the choice on each trial.

The details

For computing the valence of an outcome the prospect theory formula has been adopted:

$$V(o) = \begin{cases} o^{\alpha} & \text{if o is positive} \\ -\lambda(o^{\alpha}) & \text{if o is negative} \end{cases}$$

The details

Updating (learning) the expectancies about the deck valency:

$$E(d_{t+1}) = E(d_t) + a(V(o) - E(d_t))$$

The details

The probability of choosing deck D_i is an increasing function of the expectancy for that deck and a decreasing function of the expectancies for the other decks.

$$Pr[D_i|t+1] = \frac{e^{Ev[D_i|t]\theta(t)}}{\sum_{j=1}^{4} e^{Ev[D_j|t]\theta(t)}}$$

The results

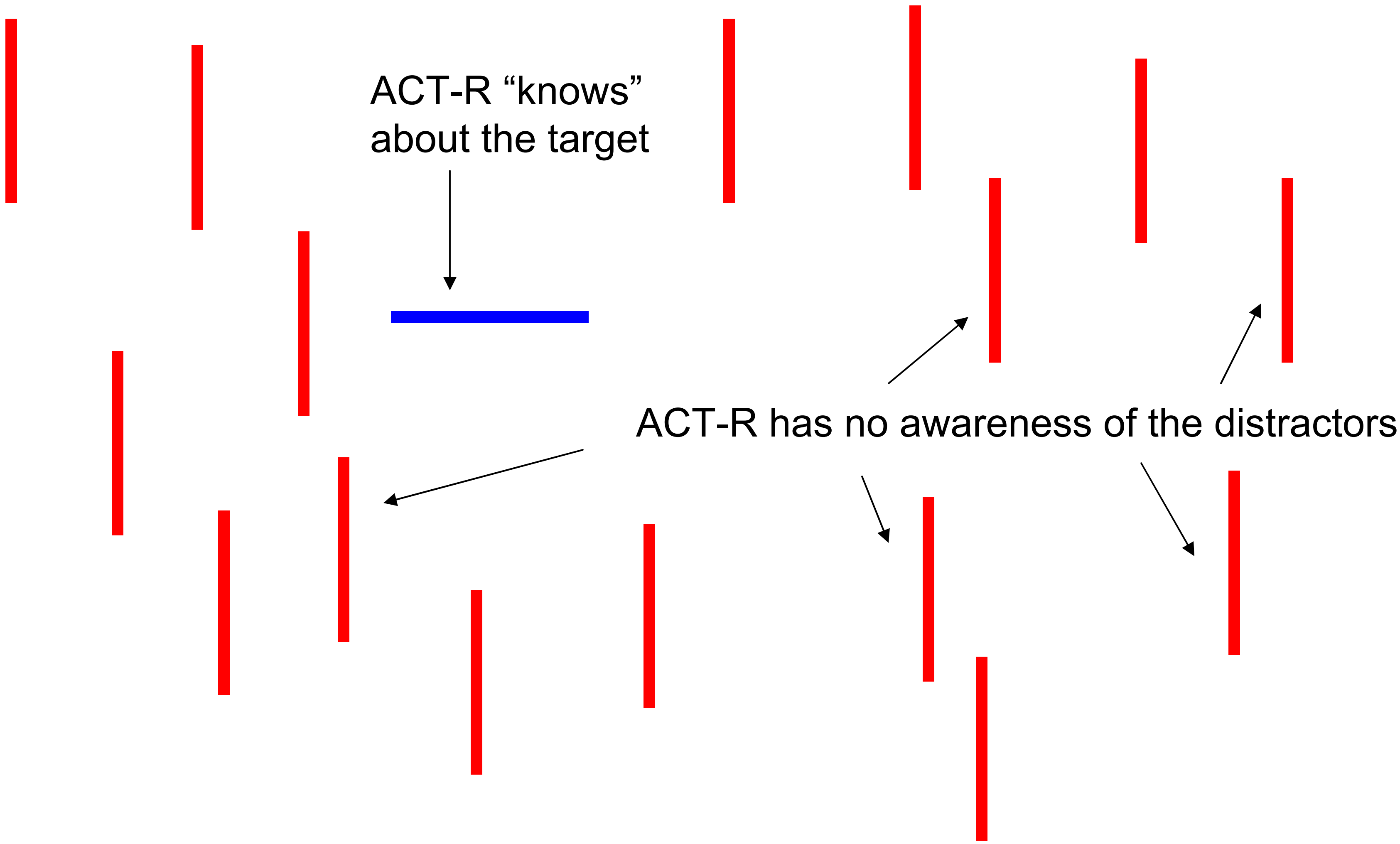
- The model produces interesting, even if not fully satisfying, results (to be shown at the workshop).
- While the qualitative structure of the data is replied, there are some difficulties in finding a single set of parameters capable of providing a good fit with the data of the three groups employed in the experiments (and of the pilot studies, too).
- Some of the model predictions are currently being tested in a new set of experiments.

Modeling visual search tasks: Is there a memory trace for unattended information?

Troy Kelley
U.S. Army Research Laboratory
Human Research and Engineering Directorate
AMSRD-ARL-HR-SE, APG, MD 21005
Tel: 410-278-5859
Fax: 410-278-9694
email: tkelley@arl.army.mil

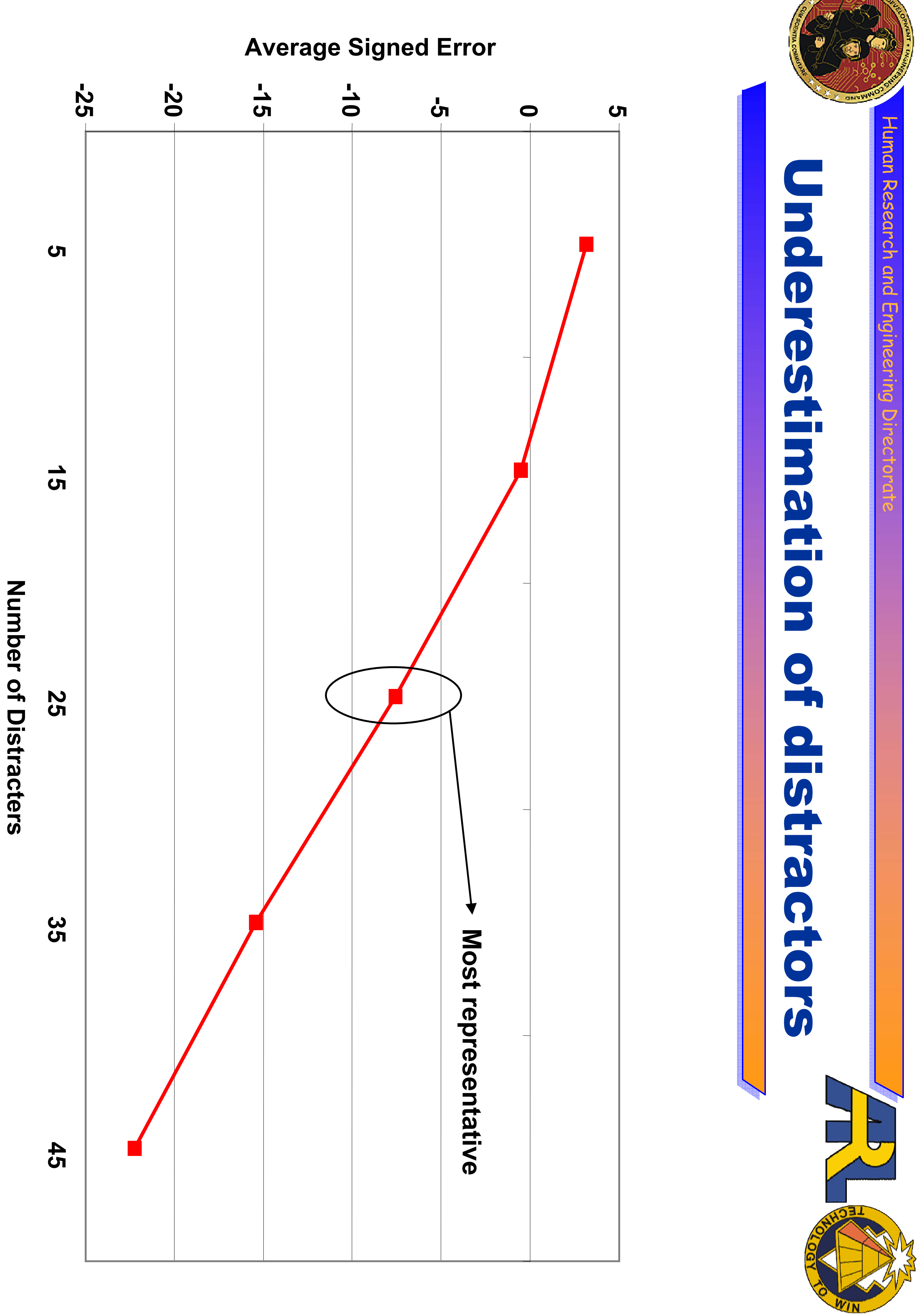
Human Research and Engineering Directorate

ACT-R awareness during a pop-out search



Question?

- Do people have an awareness for distractors during a pop-out search?
  - **Results** – Participants in our studies **do** have a limited memory for the number of distractors in a pop-out search
  - **However!!** – The number of distractors estimated by participants was consistently lower than the actual number of distractors


Human Research and Engineering Directorate
Underestimation of distractors
Average Signed Error
5
0
-5
-10
-15
-20
-25
5
15
25
35
45
Number of Distracters
Most representative
TECHNOLOGY TO WIN

Why do people underestimate the number of distractors?

- Additional studies (7 studies total) indicated that subjects were grouping distractors based on gestalt principles of proximity as well as the time given to conduct the search
  - **Evidence** – Participants have higher underestimates if the distractors clustered together as apposed to spread out.
  - **Evidence** – If subjects are explicitly told to estimate the number of distractors (i.e. attend to the distractors) their underestimates were effected by the amount of time given to attend to the distractors (i.e. the more time allotted, the better the estimates)

Human Research and Engineering Directorate

Conclusions

- The is a certain amount of information that gets into the cognitive system even if the information is **not** directly attended
- The visual buffer of ACT-R needs to create chunks that contain certain types of information (i.e. numerosity, perhaps shape), even if the items are **not** directly attended
- Further research needs to be done to determine how to express this phenomena computationally
- Gestalt principles have proven to be notoriously difficult to represent computationally

RICE

A Theory of Visual Salience Computation in ACT-R

Michael D. Byrne
Department of Psychology
Rice University
Houston, TX
byrne@acm.org
http://chil.rice.edu/

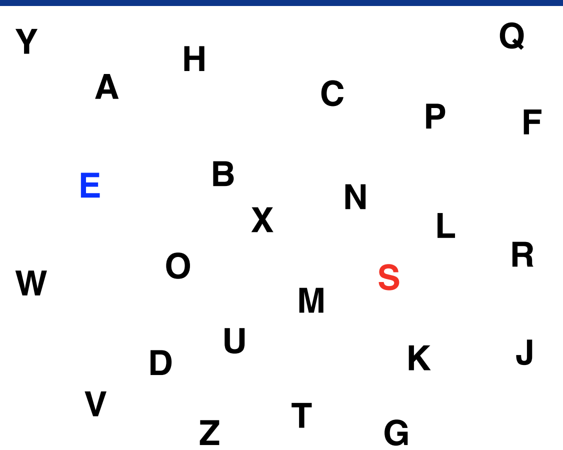
chil
THE COMPUTER-HUMAN INTERACTION LABORATORY AT RICE UNIVERSITY

Overview

- Visual salience and search issues and extant approaches
- A rational analysis approach
- Base-level salience
- Spatial constraints
- Value constraints
- Limitations and future work
- Demo (if time)

ONR 60 Years 1946-2006
Relevant Research and Results . . . Yesterday, Today, and Tomorrow

Identify the BLUE letter



Salience and Search

- Fundamental problem: where to look next
- Issues (partial list)
  - Bottom-up salience
  - Preference for feature values (e.g., look for blue things)
  - Spatial preferences
    - Including very complex ones
  - Dynamic displays
    - Onsets
    - Movement
  - Relationship to eye movements

Extant Approaches/Models

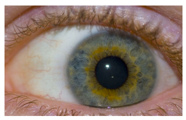
- So, there's good news and there's bad news
- This problem (or parts of it) is an extremely popular one, so why re-invent the wheel?
- Examples (also a partial list):
  - Triesman
  - Wolfe
  - Itti & Koch
  - Deco & co.
  - Rosenholtz
  - Duncan & Humphreys
  - Pomplun
  - ACT-R
  - Cave
  - Logan
  - Nakayama & co.
  - Chelazzi
  - Humphreys & Müller
  - Desimone

Steal only from the best...

- But who's the best, and why?
  - Current ACT-R not it, as it doesn't handle bottom-up salience, nor does it ever err on conjunctive searches, etc.
- Approaches vary on a great many dimensions
  - General focus, central data, computational properties, degree of neural inspiration, and many more
  - How to reconcile/synthesize all this?
- Identify common principles, find unifying theoretical basis

Rational Analysis

- The great missing question in all these models: why do these models work the way that they do?
  - Because they can fit some interesting data (e.g., Wolfe)
  - Because they believe it maps to the neuroscience (e.g., Deco)
- Rational analysis approach: What's the problem that the visual system solves by being salience-sensitive?
  - A resource allocation/limited bandwidth problem
- What's the limited resource?



7

Rational Analysis

- Have only one set of eyeballs
- Severe acuity limitations over most of the visual field
  - Therefore, move them around to sample from a probabilistic environment
- Want to maximize the amount of information which gets through the system per unit time
  - In a context-sensitive way
  - Give priority to high-information items
- Something sort of like our old buddy

$$A_i = B_i + \sum w_j S_{ji} + \varepsilon$$

8

Also, ACT-R Considerations

- Meet the needs of ACT-R modelers and be consistent with structure of the overall architecture
- Many ACT-R models look around rather a lot, in worlds where the visual scene changes regularly
- This means computational complexity has to be low
  - Can't spend all the model's time computing salience
    - Rules out the more elaborate neural and dynamical models
- Complex scenes and cases of strong knowledge about where to look when
- Handling scenes with relatively well-defined objects relative to background helps simplify the problem

9

Base-level Salience

- Want to give priority to items which carry most information
- What's the index of information carried in an alternative?
  - Get this from the Hick-Hyman law
  - Consider a display with 5 blue items, 4 green items, and 1 red item
    - The red item carries the most information because it is the least likely (like being told "it's sunny" here in Pittsburgh)

$$H(v) = \log_2 \frac{1}{p(v)} \qquad H(blue) = \log_2 \frac{1}{0.5}$$

$$H(green) = \log_2 \frac{1}{0.4} \qquad H(red) = \log_2 \frac{1}{0.1}$$

10

Complications

- Visual objects have multiple attributes which support salience (color, shape, size, etc.)
  - Fine, just iterate through the attributes and add
- Computing p(v) for continuous attributes (e.g., size)
  - One option is to simply discretize and count frequencies (and numerous models do something like this, e.g. Wolfe)
    - Puts a lot of load on the modeler to specify how categories are defined
  - Current alternative (inspired by Rosenholtz)
    - Compute absolute z-scores for attribute values
    - Transform to probabilities through normal distribution
      - For example, *z* of 1.96 yields probability of .05

11

Base-level Salience

- One more issue: certain attributes seem to carry more weight than others
  - For example, color generates more effective pop-out than shape
  - Weight each attribute in the summation
    - Determining weight values would fall under "research issues"
    - Sum of weights constrained to be 1

$$B_i = \sum_{k=1}^{\#attr(i)} \log_2 \frac{1}{p_i(v_k)} \gamma_k$$

- Terms
  - *k* iterates across the non-nil attributes of object *i*
  - Gamma is weighting factor for attribute *j*
  - $p_i(v_k)$ is probability of value on attribute *k* for object *i*

12

Spatial Constraints

- When a `+visual-location>` request is issued, various spatial constraints are allowed
  - See earlier talk on doing this better
- But how do such constraints figure into visual guidance?
- For "absolute" constraints (e.g., screen-x > 50) this can be fairly straightforward
  - Identify the items which meet all such constraints
  - Count 'em, and use that to compute p(v)
  - If the constraints identify few items, then they get a big boost; small boost if many meet constraints

13

Relational Spatial Constraints

- "Relational" constraints (e.g., "highest," "nearest") are less clear
- Current approach
  - Among objects which meet local constraints, count objects which also satisfy relational constraints
  - Use that frequency to compute another p(v) and count bits
  - Note this "resolves" the order ambiguity problem
  - Doesn't currently use new spatial specification system
- Other approaches should be explored

14

Value Constraints

- `+visual-location>` requests can also specify constraints on attribute values, such as "color: blue"
- There are multiple options here as well
- Took a simple approach
  - Again, go back to an old friend $\sum w_j S_{ji}$
  - But how to set S_{ji}?
  - Again, be simple
    - If $value_{jk} = value_{ik}$, then $S_{ji} = S_{max}$
    - $S_{max} = 1$, but this is settable
- Again, evaluating other approaches is important
  - Hook function available allow alternatives

15

Current Equation

$$L_i = \log_2 \frac{1}{p_i(v_k)}$$

- Noise is logistic, settable
- Base-levels updated when scene changes
  - That is, on every `proc-display` call
- Context parts updated on `+visual-location>` request
- `+visual-location>` request returns location with highest salience if above threshold
  - Must match on specification of `:attended` as well
  - Threshold is settable as well

Limitations and Future Work

- Details still to be worked out (e.g., attribute weighting)
- Need a better model of the retina
  - Acuity limitations
  - Insensitive to certain attributes as eccentricity increases
    - For example, very limited color vision outside of fovea
- Need tighter integration with EMMA
- Proximity/clutter effects
  - Nothing in the information content suggests this should be a factor (I think)
  - This is probably the biology “showing through”
- Effects of onsets and other changes

Demo