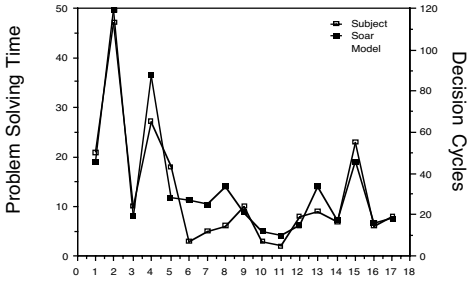
Slides presented at the ONR, NSF, and DARPA Symposium on Reasoning and Learning in Cognitive Systems, CSLI, Stanford, March 20-21, 2004.

Modelling How and When Learning Happens in a Diagrammatic Reasoning Task

Frank E. Ritter
School of Information Sciences and Technology, Psychology, and Computer Science
Penn State

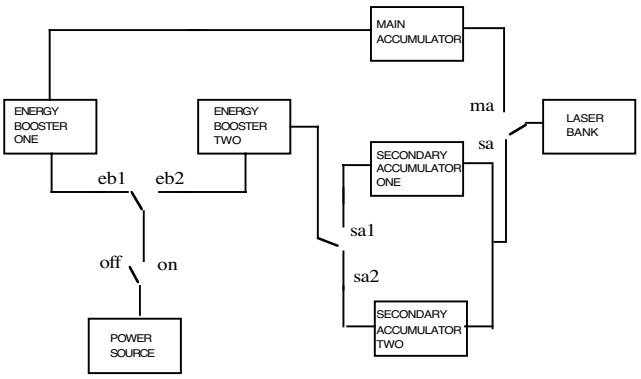
Peter A. Bibby
School of Psychology
U. of Nottingham



Supported by Joint Council Initiative in HCI and Cognitive Science

Diagrams, Interfaces, and Klingons

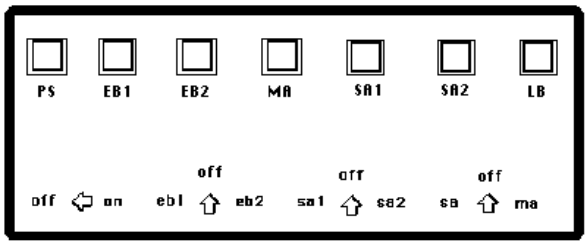
- Based on Kieras & Bovair's Starship+ (='Klingon') interface



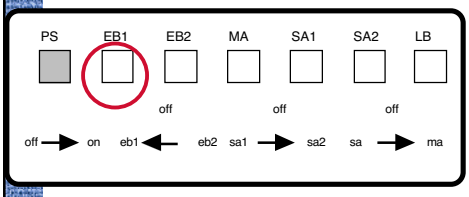
Device Schematic

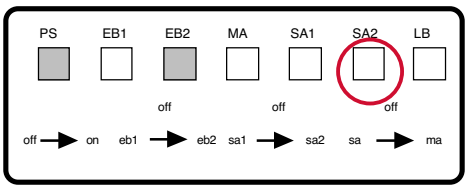
Task Details

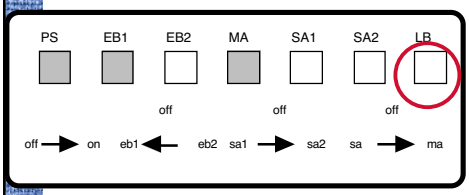
- Subjects are given:
  (i) a general intro. to the problem
  (ii) basic information on the interface
  (iii) a diagram of the underlying circuit
- Subjects are told that ONE component in the circuit is faulty, and are asked to indicate which one

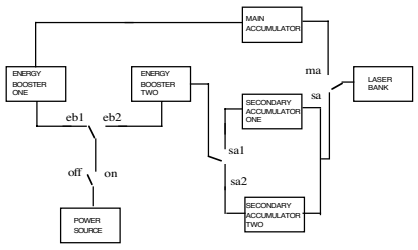


Some Example Faults









Device Schematic

Why pay attention?

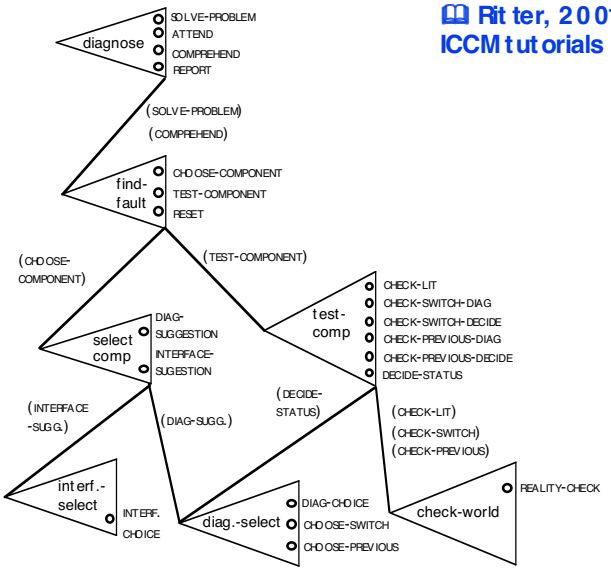
- How to learn with interaction
- Illustrates human-level learning
  - Forbus gold standard,
  - Langley notes "just on this task"
- Shows that Soar chunker models learning and transfer of learning
- Shows learning and/with reasoning
- There is no wireless network

Diag-Soar (v16)

(Ritter & Bibby, 2001)

- 173 rules + 220 chunks (new rules)
- Schematic knowledge represented as linked lists, organized as 'routes' through the circuit
- Visual interface information represented as declarative structures for lights & switches
- Status of interface diagram represented in top goal, accessed by *attend*

Diag-Soar's Problem Spaces



📖 Ritter, 2001. *Soar ICCM tutorials*

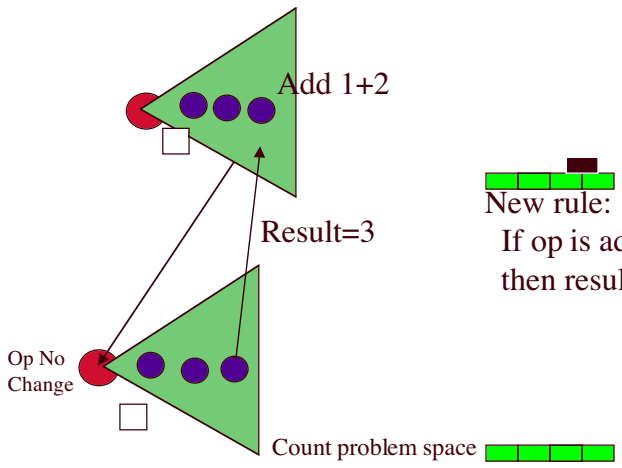
Diag-Soar (cont.)

- Organization of components on the interface diagram 'sequences' the checking of each component
- If a subgoal requires perceptual information, goal stack must be re-built
- *attend* and *comprehend* operators used to represent the perceptual components of the task

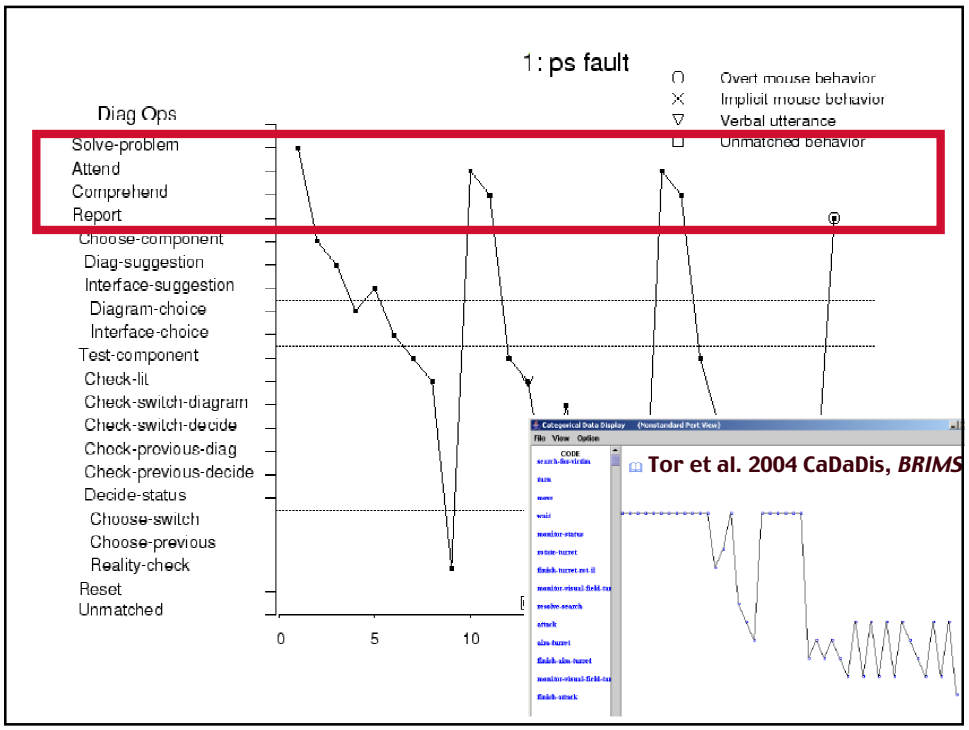
Representation?

- Experience and knowledge stored in Soar production rules
- Rules organized as implementing problem spaces, operators, and choices between them
- All learned knowledge is also represented in rules with these structures

Soar, Learning in Action



New rule:
If op is add 1+2
then result = 3



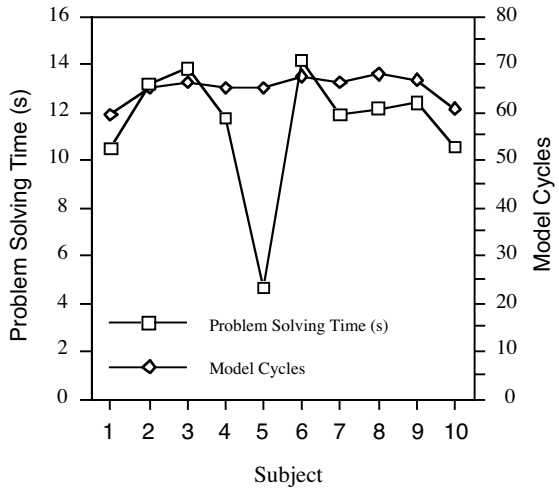
What is Learned?

- (All learning implemented as new rules)
- Where to look - implementation of Choose-component; and creation of Attend and additions to Attend
- What stimuli mean - implementation of Comprehend (and lower operations)
- Augmentations to the state from previous problem solving
- Huge amounts of transfer

General Results

- Does the task
- General Strategy — sequential components checking — emerges from the interface representation, consistent with Ss protocols
- Final ('fully chunked') behaviour reflects this strategy, behaving as if it were simply a recognition (immediate response) task

Average RTs by Subjects

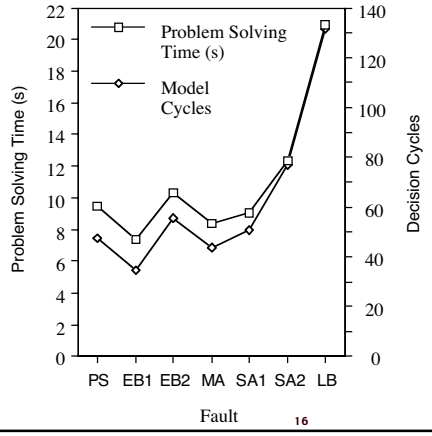


The data

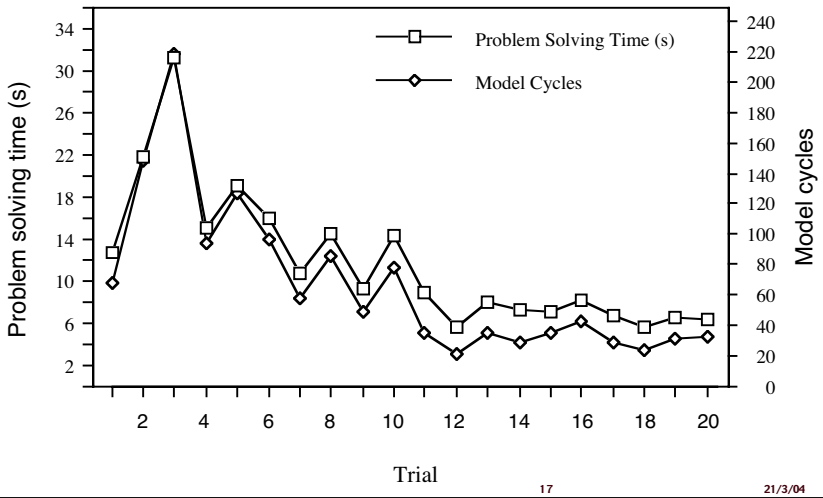
- 10 subjects solved
- 20 problems
- Each subject saw a different series of problems
- Problems sampled with replacement (nominally)
- RTs and answers recorded
- Incorrect trials discarded
- Some comments taken at end of trials

RTs by Fault Type

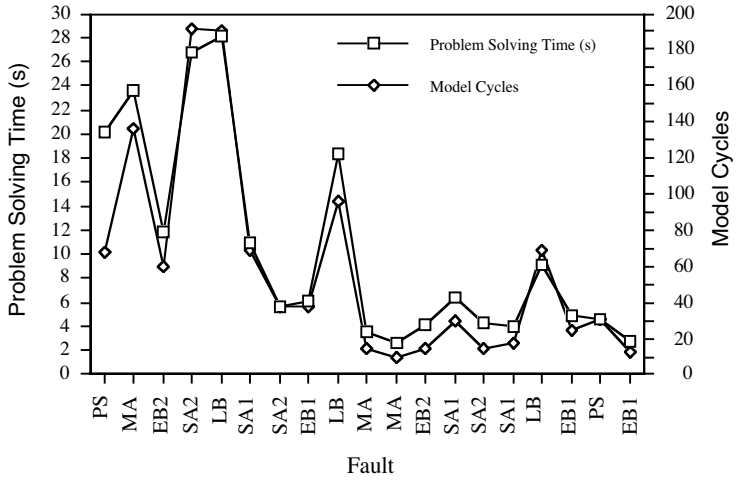
- Accounts for 90% of Ss' RT variability



RTs by Trial Number



Matches RT by Individual's Trials (s9)



Learns like No Other Model Tested with Data (!?)

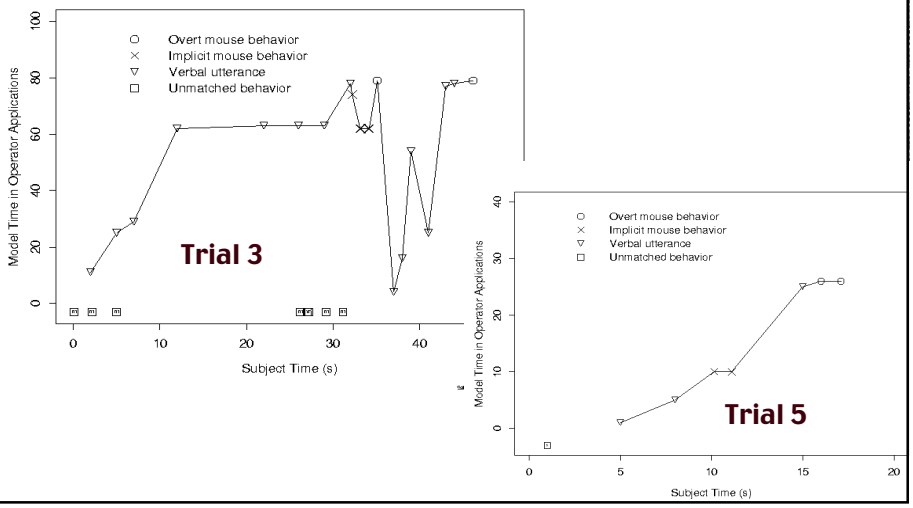
∴ No model with automatic learning, tested in detail across tasks while learning

- Able does not do transition on the fly
- Anderson's tutors just add rules
- VanLehn looked at new rule acquisition but added rules by hand
- Altmann ('99) programming model does this with recognition of objects

Data and Regularities Left Behind

- Modelling the perceptual improvements in motor skills omitted
- No account of *initial* learning of the task
- Model accounts for < 10% of the variability for some subjects (2 out of 10)
- 2nd trials on a problem are too fast in the model; Verbal protocols not reported here

Evidence for Reflection in Participants



What Can Diag-Soar Tell Us About Reasoning with Diagrams?

- Our subjects learn and **only** learn the diagram information that is relevant to the context of each stage of the problem-solving
- The model supports that subjects still use the diagram information in a cyclical, iterative fashion as an external resource to support the problem-solving sequence
- Result is recognition-driven problem solving, rather than model-driven behaviour arising out of problem solving
- Soar's chunking models transfer in this task

Where Next ?

- Model transfer on a fine level of detail
- Explaining where learning may have mismatched
- Apply to a more dynamic task such as dTank (a simple tank game)
- Modeling the type of reflection suggested in the protocols

Thank you

- Ritter, F. E., & Bibby, P. A. (accepted pending revisions). Modeling how and when learning happens in a diagrammatic reasoning task. To *Cognitive Science*.
- Ritter, F. E., & Bibby, P. (2001). Modeling how and when learning happens in a simple fault-finding task. In *Proceedings of ICCM - 2001 - Fourth International Conference on Cognitive Modeling*. 187-192. Mahwah, NJ: Lawrence Erlbaum. http://acs.ist.psu.edu/papers/ritterB01.pdf
- Ritter, F. E. (2003). Soar. In L. Nadel (Ed.), *Encyclopedia of cognitive science*. vol. 4, 60-65. London: Nature Publishing Group. http://acs.ist.psu.edu/soar-faq http://acs.ist.psu.edu/nottingham/pst
- Tor, K., Ritter, F. E., Haynes, S. R., & Cohen, M. A. (in press). CaDaDis: A tool for displaying the behavior of cognitive models and agents. In *Proceedings of the 13th Conference on Behavior Representation in Modeling and Simulation*. http://acs.ist.psu.edu/papers/torRHC04.pdf