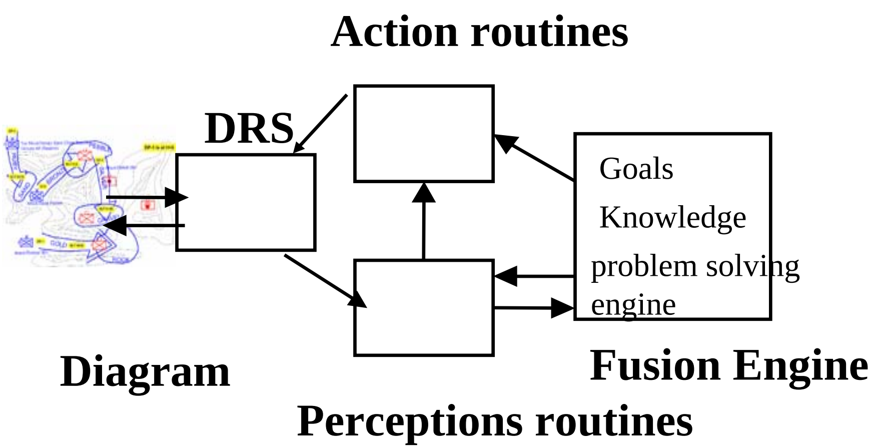
Augmenting Cognitive State with Diagrams: Proposal and Call for Discussion

B. Chandrasekaran and
Unmesh Kurup
Ohio State University LAIR
Soar Workshop, Ann Arbor, MI
June 15, 2005

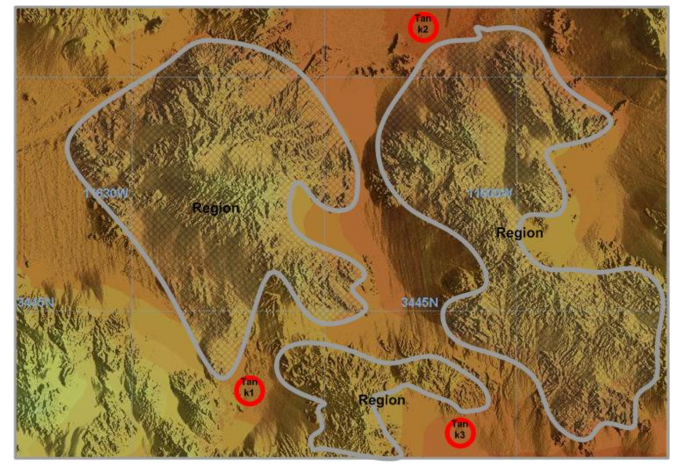
Cognitive State is Multi-Modal

- Traditional AI/CogSci state representation (Soar, ACT-R) is “predicate symbolic.”
- State change is accomplished by rule-based operators that transform the predicate symbolic representation. (Inference)
- Cognitive architectures such as SOAR should support an augmented notion of cognitive state
  - Traditional “symbolic” will be one, generally dominant, component of state, but other perceptual and kinesthetic modalities need to be supported
  - State change operations get more complex
  - Diagrammatic representations make a good place to start because of their ubiquity and usefulness in problem solving

Example Diagrammatic Reasoning

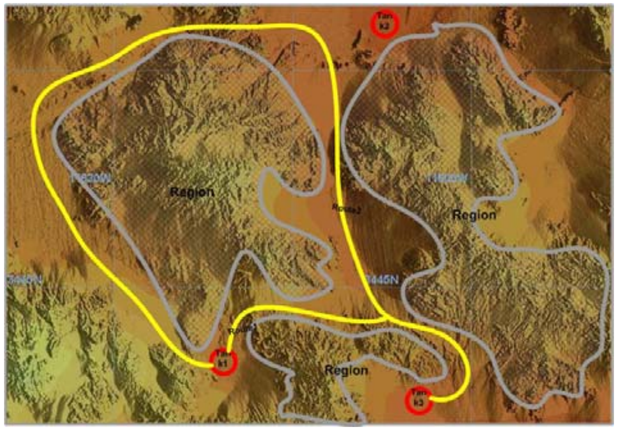


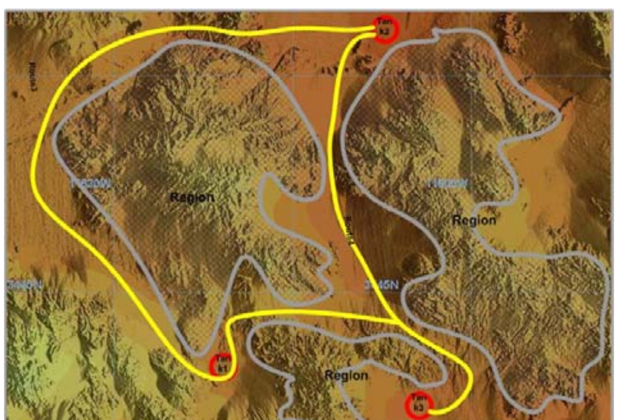
Area of Interest (AOI)



The start: On receipt of sighting of T3, the problem solver – the Fusion Engine (FE) in this case -- queries the entity database for entities of the same type in the Area of Interest, and gets back two vehicles T1 and T2, their types, locations and times of sightings.

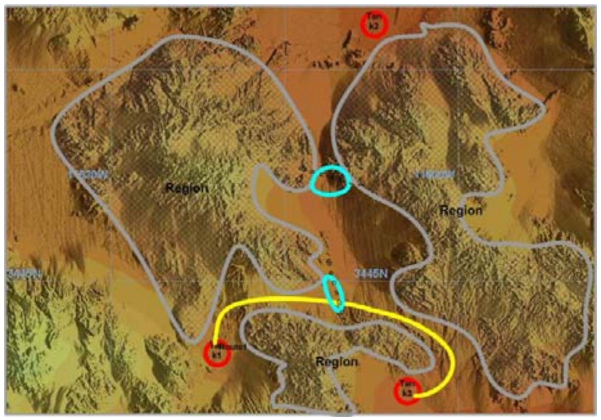
Examining Possible Routes





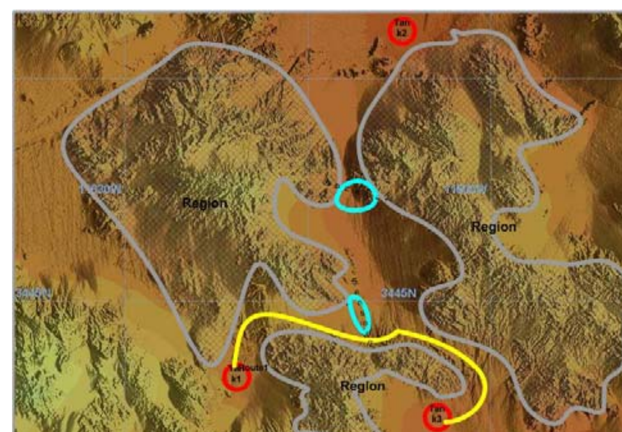
- Fusion Engine (FE) asks Diagrammatic Reasoner (DR) to identify routes that T1 and T2 might have taken to get to T3.
- FE rules out the longer route in each case as too long, based on time elapsed, leaving one route each
- FE asks database for information of tunnels, storage depots, etc., from which new vehicle might have made appearance. New vehicle hypothesis is ruled out.

Failure of Expectation Critiques: Crossing Sensor Fields



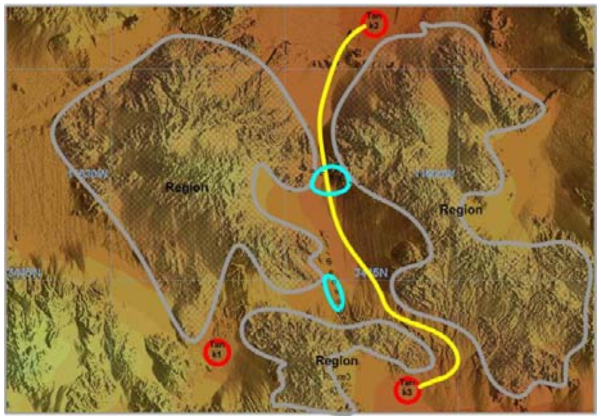
FE identifies from the database two sensor fields in the AOI, and information that neither of them reported any sightings. The fields are added to the diagram.

FE asks DR if Route 1 intersects a sensor field. PR identifies Sensor_Field2.



FE asks if the route could be modified so as to avoid the sensor. DR tries it and says, yes.

Repeat Failure of Expectation Critique for T2



- However, DR says this time that the route cannot be modified to avoid the sensor field.
- FE now proposes T1, along Route 1, as the most likely hypothesis.

Emergent objects and emergent relations key source of power of DR

A to the left of B, B to the left of C. Diagram enforces A to the left of C.

“Free ride”

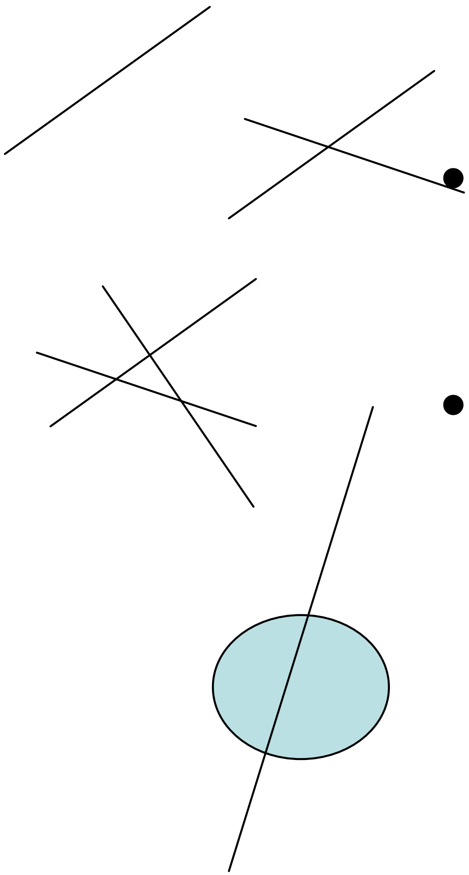
Need for State Augmentation

- Soar can handle the problem solving just described
  - Diagrams are external and perception delivers predicate symbolic relational information
- However:
  - "State" of problem solving spans the agent and paper.
  - If the relevant aspect of the representation includes the spatiality of the objects or their configuration, learning should commit to LTM the relevant diagrammatic aspects of the state. (May be.)
  - "Was Stephanie standing closer to John than to Bill at last night's party?"
    - Supporting this recall and applying a perceptual operation also calls for the diagrammatic component to be part of the cognitive state in the architecture.

DRS and the Diagrammatic Component of Cognitive State

- The diagrammatic component should be composable, i.e, it should be in some sense a composition of “visual symbols,” rather than a pixel array
- It is already “interpreted” as objects, i.e, a figure-ground distinction has been made.
- DRS has all these properties.
  - Configuration of objects each of which is one of point, curve, region type.
  - The spatiality of each of these objects is represented in some form (doesn’t matter what – functionality is all).
  - DRS is the functional equivalent of a diagram in the sense that it has the same information that a diagram has – objects and their spatiality – and can be operated on by routines that are equivalent to perception on external diagrams.
  - But it is structured and (de)composable, thus elements of DRS are *visual symbols* (Barsalou)
- DRS provides a way to break through a 4-decade old argument about images vs “propositions” regarding mental images.

Perception Operators



- Spatial properties of objects
  - Length of a curve, e.g.
  - That a curve is a straight line, a region is a triangle, etc
- Emergent object identification
  - Point, curve, and region objects that are created when diagrammatic objects are declared.
- Emergent relations
  - Inside, touches, left-of (a,b,pov),
  - Subsumption relations
  - Angular relations
  - …. (domain-independent …domain-specific.)

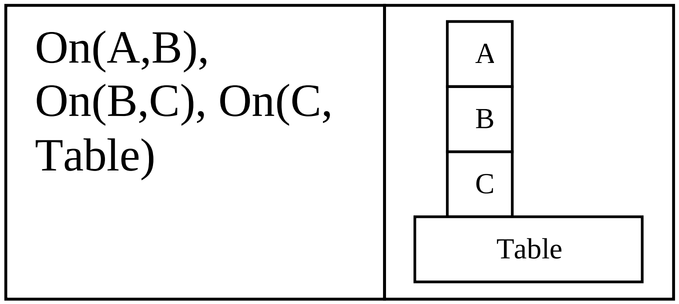
Perceptual Operators Operate on DRS

- DRS represents the result of figure-ground and object organization stage of early perception.
- Is present whether the agent is perceiving an external diagram or mentally composing one from LTM.
- Perceptual Routines operate on DRS elements.
  - However, degree to which composed DRS supports reliable relational perception is a controversial issue (Pylyshyn), especially between elements from different DRS fragments that went into the composed DRS.

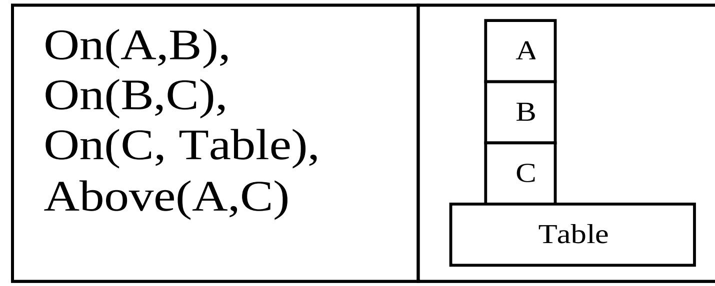
Action Operators

- Create objects satisfying properties or relations
  - E.g, Point to the left of region, a point on curve,..
  - Curve such that it connects points A and B, and avoids region R.
- Action operators make use of perception operators to satisfy constraints.
- If Action operators apply to external world or representation, external perception of agent can construct appropriate DRS corresponding to the new situation.
- However, degree to which human cognition contains the complete causal structure of external space to correctly create, purely by internal operations, the DRS for the resulting external situation very controversial, especially if actions take place in a 3-D world.
  - Pylyshyn's critiques over the years.

A Bimodal State



A bi-modal state



Alternate symbolic description of same world state as in Fig. 1.

- The DR component is "complete" in a way the symbolic component is not
  - e.g., see alternate description of same world state.
  - Potential for helping with aspects of the Frame Problem.

aSoar: Soar with a diagrammatic component

- Problem state is a bi-modal
- A perceptual routine can be executed on a diagram by calling the routine in the RHS of an aSoar production.
  - Problem solver in aSoar needs to translate from the domain dependent perceptual questions to generic ones supported by DRS. E.g., if question is: “Is block A inside of box B1?”, the question is translated into “Is region A inside of region B1?”
- aSoar problem solver can also
  - modify the diagram by invoking the action routines
  - modify the symbolic components by applying perceptual routines to the diagram.

Example: Move(A,Table) in Soar

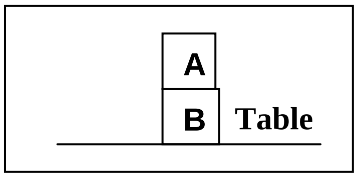


Fig 3: A simple blocks world example

Goal State:

On(A,Table)

World State:

Block(A)
Block(B)
Table(T1)
On(A,B)
On(B,Table)

Operators:

Fig 4: Initial contents of Soar's WM

Goal State:

On(A,Table)

World State:

Block(A)
Block(B)
Table(T1)
On(A,B)
On(B,Table)

Operators: move(A,Table)

Fig 5: Soar's WM after operator proposal

Goal State:

On(A,Table)

World State:

Block(A)
Block(B)
Table(T1)
On(A,Table)
On(B,Table)

Operators:

Fig 6: Soar's WM after Move applied

A highly de-syntaxified version of how standard Soar might work in this case

Move(A,Table) in aSoar

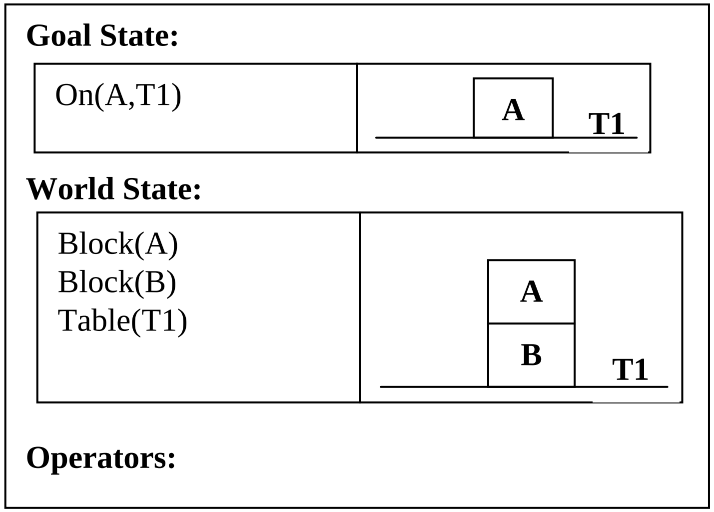


Fig 7: Initial contents of aSoar's WM

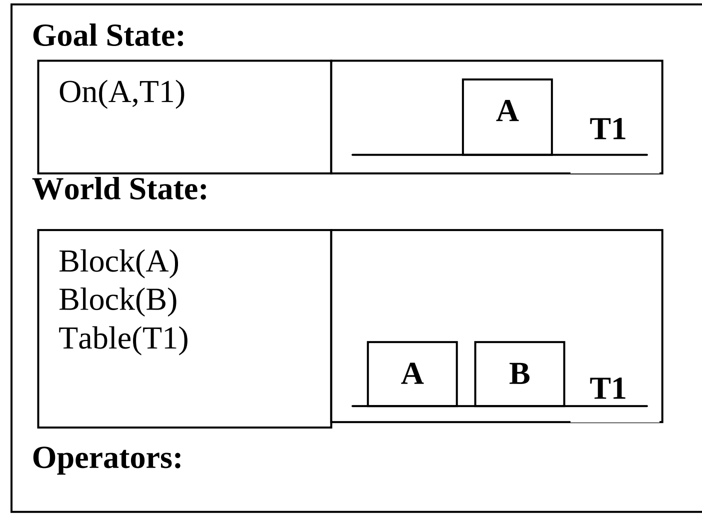


Fig 8: aSoar's WM after Move applied

- During proposal phase, the rule that proposes the Move operator fires (not shown in figures).
- During the application phase, instead of updating the symbolic part, the rule calls the action routine to update the diagram to reflect Move(A,Table). Checking for preconditions can be done directly by the relevant perceptual routines.

Potential for Savings on Soar Cycles

- See Example 2 (at end) for example in which # of aSoar cycles is much smaller than the # of Soar cycles
  - Extra Soar cycles are taken up with inferring relations that are available for pick up by perception from the DRS component. That is one cycle involving perception may require an arbitrary large # of Soar cycles without perception.
  - If perception is “free,” and more or less independent of the # of objects, the savings can be big.

Issues

- A jerry-rigged version of aSoar has been implemented, but we’d like to discuss issues in building a more robust version.
- Issues in chunking involving diagrammatic components

Nuggets and Coals

- Nuggets:
  - The idea that cognitive states should have perceptual components and cognitive architectures should support them as a matter of course seems compelling.
  - That Diagrams are great window to look into the phenomenon seems right.
  - DRS as a visual symbol system seems on the right track.

- Coals:
  - Argument for bi-modal state seems less compelling based on need for problem solving than for episodic memory.
  - Many aspects of a clean elegant implementation that carries the theory with conviction are unclear at this point.

Appendix:Example 2

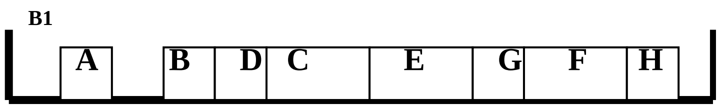


Fig 9. The initial state for Example 2.

The goals to be achieved are: B *inside-of* B1, E *above* A, F *above* A, H *above* A, D *above* B, G *above* A.

Blocks can be one of two sizes: a standard size and one whose length is twice that of the standard. A box is open at the top, size of double-length block, and can contain a block or blocks.

Relations between blocks: *On-top-of*, *under*, *above*, *below*, *imm-right-of*, *imm-left-of*, *right-of*, *left-of* and *inside-of*.

Soar:
on-top-of, *under*, *imm-right-of*, *imm-left-of* and *inside-of* are the primitive relations, used in add and delete lists. Remainder inferred as needed

aSoar:
Whatever is needed is directly perceived.

Making a Dent on the Frame Problem

- Number of aSoar cycles can be much less than for Soar.
- Depending on assumptions about the cost of perception in general and with respect to number of objects in particular, this can be significant.

Standard Soar	Multi-modal Soar
Check if Box B1 clear	Check if box B1 clear
Move B to inside of B1	Move B to inside of B1
Check if A is clear	Find topmost block above A
Move E onto A	Move E onto A
Check if A is clear.	Find topmost block above B
No. Find block on-top-of A	Move F onto A
Check if E is clear	Find topmost block above A
Move F onto E	Move H onto F
Check if A is clear	Find topmost block above B
No. Find block on-top-of A	Move D onto F
Check if E is clear	Find topmost block above A
No. Find block on-top-of E	Move G onto H
Check if F is clear	
Move H onto left-half of F	
Check if B is clear	
No. Find block on-top-of B	
Check if E is clear	
No. Find block on-top-of E	
Check if F is clear	
Move D onto right-half of F	
Check if A is clear	
No. Find block on-top-of A	
Check if E is clear	
No. Find block on-top-of E	
Check if F is clear	
No. Find block on-top-of F	
Check if H is clear	
Move G onto H	

Fig 10: The problem solving sequences for standard and multi-modal Soar for the problem in Fig 9.