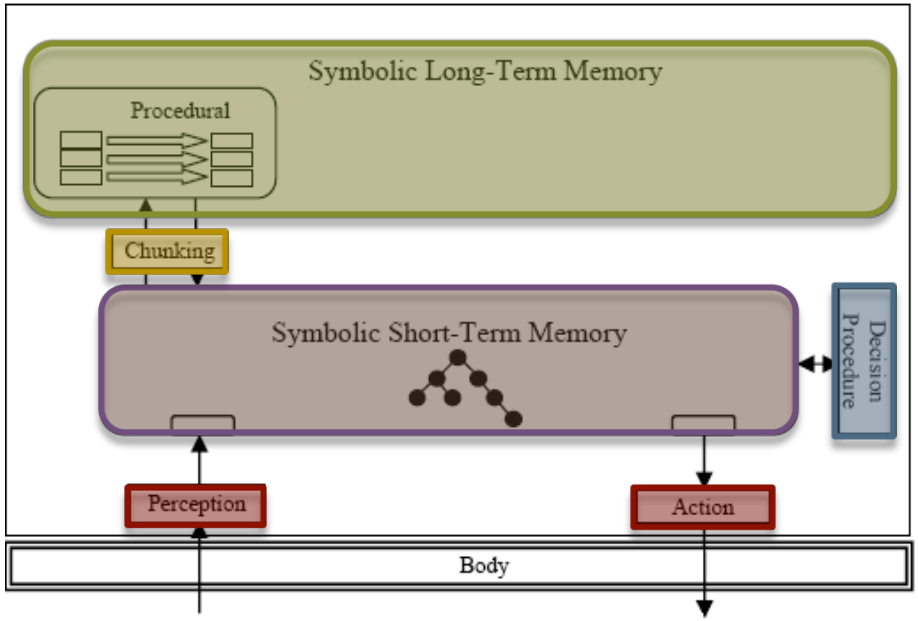
A Graphical Rethinking of the Cognitive Inner Loop

Paul S. Rosenbloom
Department of Computer Science & Institute for Creative Technologies
University of Southern California
July 13, 2009

Cognitive Architecture

- Fixed structure underlying cognition
  - Defines core memories, reasoning processes, learning mechanisms, external interfaces, etc.
- Yields intelligent behavior when combined with knowledge in memories
  - Including more advanced reasoning, learning, etc.
- May model human cognition, strive for human-like intelligence, or be purely artificial
  - Related to intelligent agent architectures
- Not been a major application area for graphical models of knowledge representation and reasoning

Example: Soar



Soar 3-8

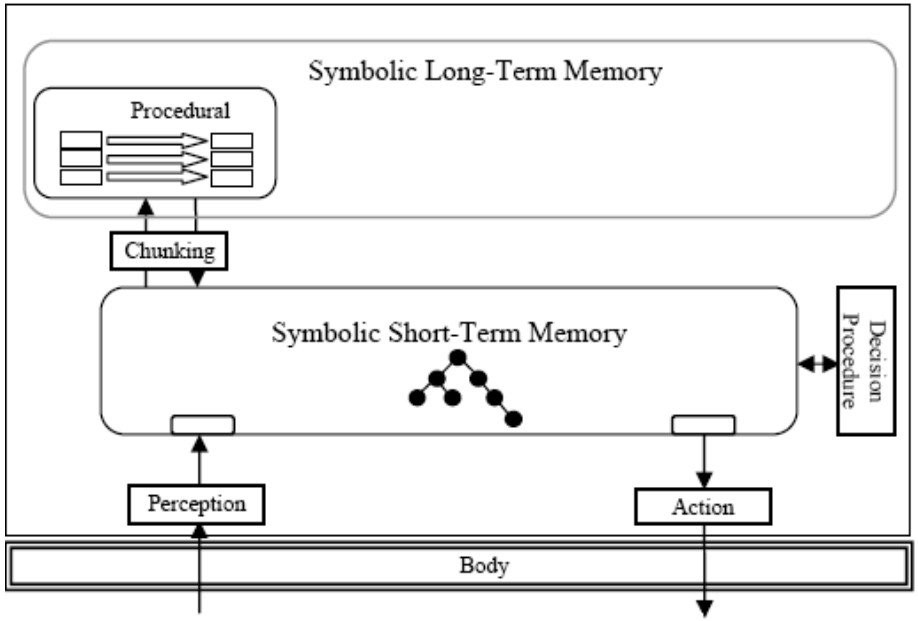
- Symbolic working memory
- Long-term memory of rules
- Decide what to do based on preferences retrieved into working memory
- Reflect when can't decide
- Learn results of reflection
- Interact with world

The Diversity Dilemma

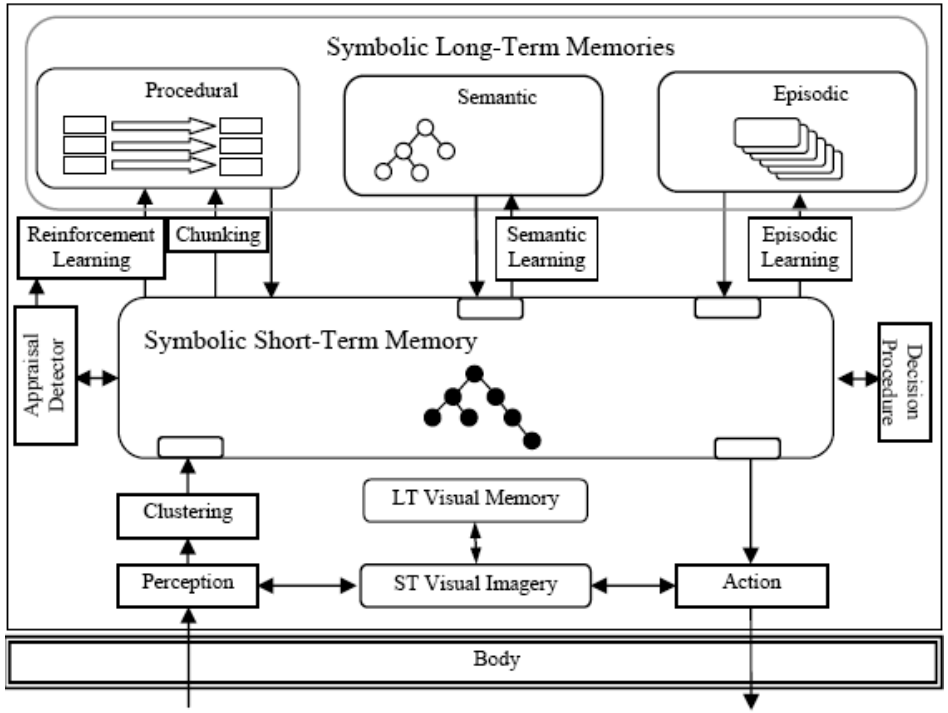
- Should an architecture's mechanisms be uniform or diverse?
- *Uniformity*: Minimal mechanisms combining in general ways
  - Appeals to simplicity and elegance
  - The "physicist's approach"
  - Achieving full range of functionality/utility can be problematic
- *Diversity*: Large variety of specialized mechanisms
  - Appeals to functionality and optimization
  - The "biologist's approach"
  - Integratability and maintainability can be problematic
- Want best of both worlds, but a choice seems inevitable
  - Functionality tends to win, leading to the predominance of diversity
  - But is there another way out?

Example: Soar

- Traditionally a uniform architecture
- Version 9 has become highly diverse



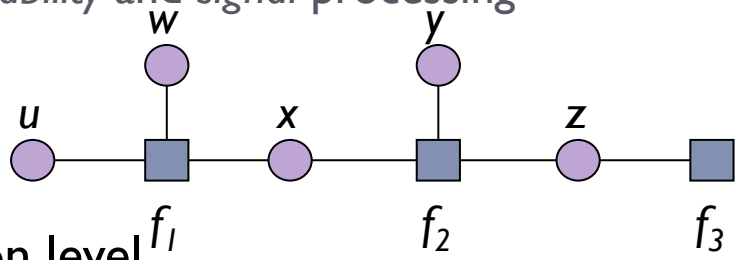
Soar 3-8



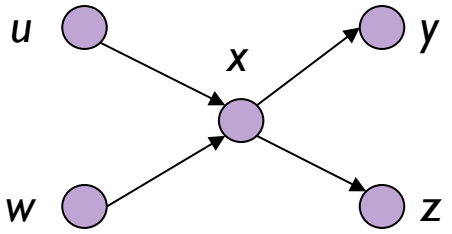
Soar 9

Proposal for Resolving the Dilemma

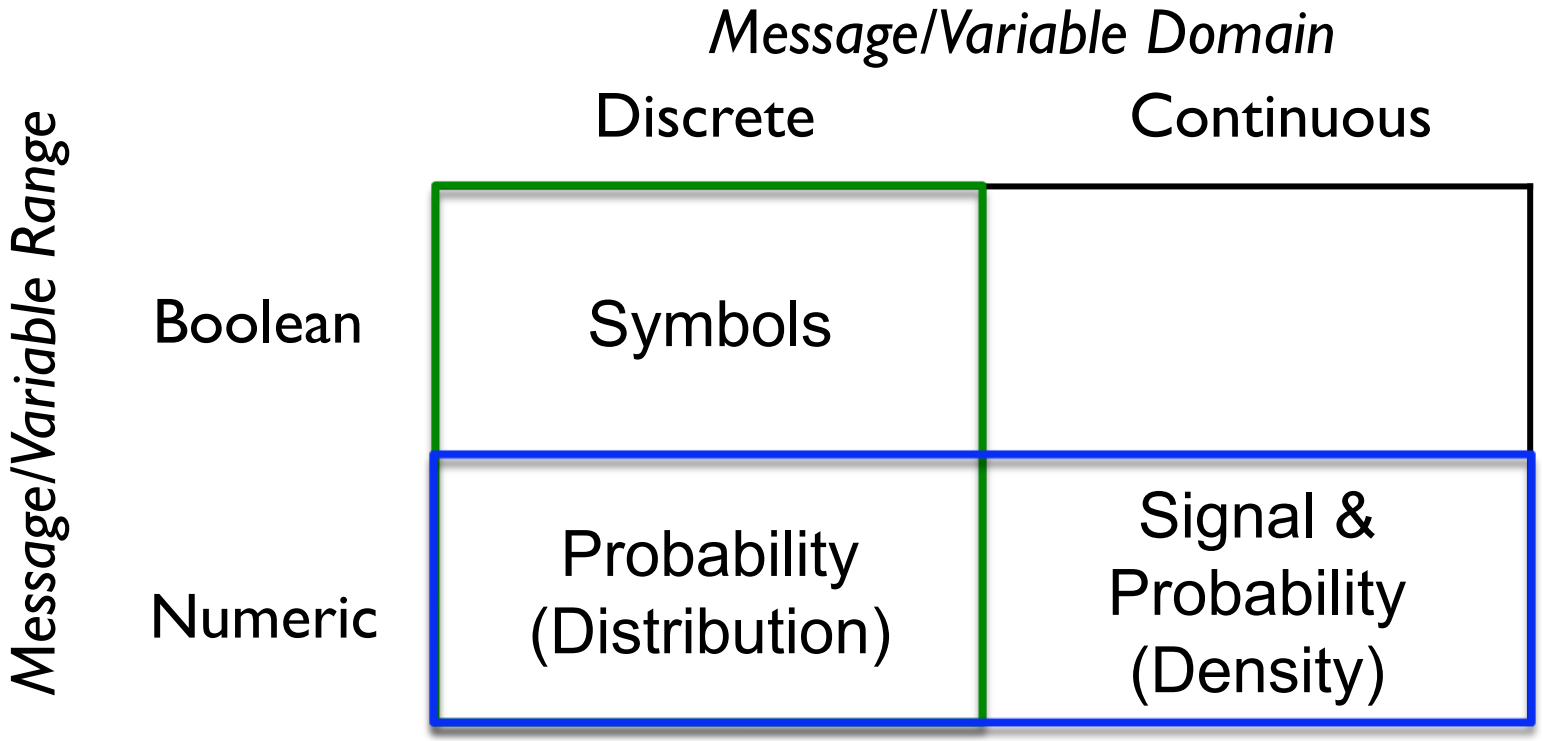
- Dig beneath architecture for uniformity at *implementation level* that supports architectural diversity/functionality
  - Implementation level is normally just Lisp, C, Java, etc.
  - Concept here is similar to Domingos's call for an *interface layer* for AI
- Base broad yet uniform implementation level on *graphical models*
  - Efficient computation over multivariate functions with partial independency
    - Bayesian ($p(u,w,x,y,z) = p(u)p(w)p(x|u,w)p(y|x)p(z|x)$) and Markov networks
    - Factor graphs ($f(u,w,x,y,z) = f_1(u,w,x)f_2(x,y,z)f_3(z)$)
  - *Sum-product* yields SOA algorithms for *symbol*, *probability* and *signal* processing
    - Belief propagation in Bayesian networks
    - Forward-backward in hidden Markov models
    - Kalman filters, Viterbi algorithm, FFT, turbo decoding
    - Arc-consistency in constraint diagrams



- Reconceive architectures via new implementation level
  - Provide general symbolic reasoning under uncertainty
  - Incorporate perception and action into cognitive inner loop
  - Possibly yield a new bridge from symbolic to neural systems



Scope of Sum-Product Algorithm



- *Mixed* models combine Boolean and numeric ranges
- *Hybrid* models combine discrete and continuous domains
- *Hybrid mixed* models combine all possibilities
- *Dynamic hybrid mixed* models add a temporal dimension

Research Strategy

- Goals
  - Evaluate extent to which graphical models can provide a uniform implementation layer for existing architectures
  - Develop novel, more functional architectures
    - Enhancing and/or hybridizing existing architectures
    - Starting from scratch leveraging strengths of graphical models
- Initial approach
  - Reimplement and enhance the Soar architecture
    - One of the longest standing and most broadly applied architectures
    - Exists in both uniform (Soar ≤8) and diverse (Soar 9) forms
  - Start from the bottom up, implementing uniform version while looking for opportunities to more uniformly incorporate Soar 9's diversity plus critical capabilities beyond all versions of Soar

Level View of Uniform Soar

Scale	Functionality	Mechanism	Details
1 sec	Reflective	Problem Space Search	Impasse/Subgoal if can't Decide
100 ms	Deliberate	Decision Cycle	Preference-based Decisions upon Quiescence
10 ms	Reactive	Elaboration Cycle	Parallel Rule Match & Firing

Cognitive Inner Loop

Previous Work with Factor Graphs

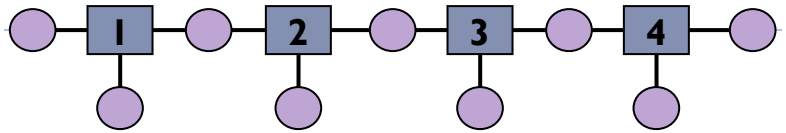
Production Match via Message Passing

Learning by Chunking

From Newell's (1990) analysis of time scales in human cognition

Exploring an Implementation of Soar's Decision Cycle via Alchemy (Markov logic)

- Markov logic = First order logic + Markov networks
  - Node for each ground predicate
  - Weight for each ground clause (clique potentials)
    - Along with links among all nodes in ground clause
- Goals for implementation
  - Explore a *mixed* elaboration phase (rules & probabilities)
    - Parallel rule firing until quiescence
  - Explore incorporating semantic (fact) memory and *trellises*
  - Enable bidirectional message flow across rules
    - Normal rule firing only propagates information forward
    - Need bidirectional flow for correct probability and trellis processing
  - Only minimal investigation of actual decision procedure



Encoding

- Convert productions into logical implications
  - Define types for objects and values of (object ^attribute value) triples
    - colors={Red, Blue, Green} and objects = {A, B, C, D, E, F}
  - Define predicates for attributes
    - Color(objects, colors) and Type(objects, objects)
  - Specify implications/clauses for rules
    - (Type(v0, v1) ^ Color(v1, v2)) => Color(v0, v2).
  - Add weights to clauses as appropriate

P1: Inherit Color

C1: (<v0> ^type <v1>)

C2: (<v1> ^color <v2>)

-->

A1: (<v0> ^color <v2>)

- Initialize evidence (db file) with working memory
  - Color(C, Red), Color(D, Blue), Type(A, C), Type(B, D)
- Semantic memory: weighted ground predicates: 10 Color(F, Green)
- Trellis: define via a pair of implications (*accept* & *reject* prefs.)
  - Size(step, size) => Size(step+1, size*2).
  - (Size(step, size1) ^ size1!=size2) => !Size(step, size2).

Alchemy Results

- Mapping basically works (modulo trellis strangeness)
  - Mixed representation with simple semantic memory and trellises
- Match occurs via graph compilation not message passing
  - As Alchemy compiles first-order clauses to ground network
    - A ground node for every element of working memory
    - A ground clause for every production instantiation
  - Should all symbolic processing occur in compilation and probabilistic in propagation?
    - Falls short of uniform processing in the graph itself
- Implies a three phase decision cycle
  1. Compile/match to generate a ground/instantiated network
  2. Perform probabilistic inference in the ground network
  3. Decide

Alchemy Implication

- Alchemy, and systems like it, get stuck in local minima
  - Generally considered a problem, but is it actually appropriate?
- If Alchemy maps onto Soar's decision cycle then should only perform *Knowledge Search*, not *Problem Space Search*
  - K-Search: Fast (~100ms), parallel, bounded, closed
  - PS-Search: Slow (>1s), serial, combinatoric, open
- Idea
  - K-Search only yields local minima in general
  - PS-Search is required, in general, for global minima
    - Through a sequence of local minima
- Implication would be that Alchemy should just yield local minima, but it also needs PS-Search on top of it
  - The same may then also be true for all one-level, logical and/or probabilistic inference systems

Soar Implication

- Taking this a step further, we can hypothesize functionally that:
  - Problem Space Search (> 1 sec): Global minima
  - Decision Cycle (100 ms): Global propagation but only local minima
  - Elaboration Cycle (10 ms): Local propagation of information
- Implies productions shouldn't globally propagate information
  - Explicit global: Creating unique identifiers
  - Implicit global: Non-monotonic (negated conditions, operator applications)
  - Interacting with a global working memory?
- Can Soar function if global propagation is limited to the decision cycle?

Summary and Future

- A new kind of application for graphical models
  - Build *architectures* instead of more specific domain solutions or more general toolkits
- Rethink cognitive architecture via a graphical implementation level
  - Resolve the diversity dilemma
  - Improve elegance, functionality, extensibility, integrability and maintainability
    - Combine symbolic with probabilistic reasoning and cognition with perception
    - Bridge from symbolic to neural processing
- Exploring a graphical reimplementation and enhancement of Soar
  - Focus here is on the cognitive inner loop (the decision cycle)
  - Raises interesting issues that need resolution
    - Moving nonlocal processing from the elaboration to the decision cycle
    - Combining symbolic and probabilistic processing in message passing
  - Need to cover full decision cycle and then extend to reflection and learning
    - Looking to include probabilities and signals (for vision & speech) in the inner loop
    - Add Soar 9's capabilities for semantic/episodic memory & reinforcement learning
- Longer term, need to
  - Reexamine other cognitive architectures and hybrids among them
  - Experiment with radically new architectures enabled by graphical models
  - Evaluate feasibility & utility of uniform implementation level for architectures