Speculations on Leveraging Graphical Models for Architectural Integration of Visual Representation and Reasoning

Paul Rosenbloom
University of Southern California

7/11/2010
Outline

- Cognitive architecture
- The diversity dilemma
- Graphical models
- A graphical memory architecture
- Visual representation and reasoning
Cognitive Architecture

- **Hypothesis about the 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 and/or act as core of a virtual human or intelligent agent**
  - Strong overlap with *intelligent agent architecture*
Some Cognitive-Architecture-Based Virtual Humans

Improving in *mind* and *body* over the years

These all based on *Soar architecture*
Soar Architecture (Versions 3-8)

- Symbolic working memory
- Long-term memory of rules
- Decide what to do next based on preferences generated by rules
- Reflect when can’t decide
- Learn results of reflection
- Interact with world

Pursued both as a *unified theory of cognition* and as an architecture for virtual humans and intelligent agents
Systems Levels in Cognition

- In the large, architecture is a theory about one or more systems levels in an intelligent entity
  - Usually part of Cognitive Band

- At each level, a combination of structures and processes implements basic elements at the next higher level

(Newell, 1990)
Hierarchical View of Soar (Versions 3-8)

<table>
<thead>
<tr>
<th>Scale</th>
<th>Functionality</th>
<th>Mechanism</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 sec</td>
<td>Reflective</td>
<td>Problem Space Search</td>
<td>Impasse/Subgoal If Can’t Decide</td>
</tr>
<tr>
<td>100 ms</td>
<td>Deliberative</td>
<td>Decision Cycle</td>
<td>Preference-based Decisions upon Quiescence</td>
</tr>
<tr>
<td>10 ms</td>
<td>Reactive</td>
<td>Elaboration Cycle</td>
<td>Parallel Rule Match &amp; Firing</td>
</tr>
</tbody>
</table>

 Learning by Chunking
Diversity Dilemma

- The *girth* at a level is its range of structures & processes
- **Key issue:** *uniformity* versus *diversity*
- **Uniformity:** Minimal mechanisms combining in general ways
  - Appeals to simplicity and elegance
  - The “physicist’s approach”
  - Challenge: Efficiency and full range of required functionality/coverage
- **Diversity:** Large variety of specialized mechanisms
  - Appeals to functionality and optimization
  - The “biologist’s approach”
  - Challenge: Integrability, extensibility and maintainability
- **Want benefits of both!**
  - Can this be achieved by varying girth across levels?
- **Across a hierarchy, level girth may stay comparable or vary**
  - Physicists and biologists likely assume uniform
  - Network researchers assume *hourglass*
What About Cognition?

- **Top (applications) is clearly diverse**
  - Key part of what architectures try to explain

- **Bottom is likely diverse as well**
  - Physicalism: Grounded in diversity of biology
  - Strong AI: Also groundable in other technologies

- **Is the waist uniform or diverse?**
  - Hourglass or rectangle
  - Traditionally question about the architecture
Architectural Uniformity vs. Diversity

- Soar (3-8) is a traditional uniform architecture
- ACT-R is a traditional diverse architecture
Examples

- Soar (3-8) is a traditional uniform architecture
- ACT-R is a traditional diverse architecture
- Recently Soar 9 has become highly diverse
Towards Reconciling Uniformity and Diversity

- Accept diversity at the architectural level
- Move search for uniformity down to *implementation* level
  - *Biological Band* in humans
    - Locus of neural modeling
  - *Computational Band* in AI
    - Normally just Lisp, C, Java, etc.
      - Impacts efficiency and robustness but usually not part of theory
- Base on *graphical models*
  - Pragmatic *and* theoretic impact

(Newell, 1990)
Graphical Models

- Efficient computation with multivariate functions by decomposition into products of subfunctions
  - For constraints, probabilities, speech, etc.
  - Based on independence assumptions

- Come in a variety of related flavors
  - Bayesian networks: Directed, variable nodes
    - E.g., \( p(u,w,x,y,z) = p(u)p(w)p(x|u,w)p(y|x)p(z|x) \)
  - Markov networks: Undirected, variable nodes & clique potentials
    - Basis for Markov logic and Alchemy
  - Factor graphs: Undirected, variable & factor nodes
    - E.g., \( f(u,w,x,y,z) = f_1(u,w,x)f_2(x,y,z)f_3(z) \)

- Compute marginals via variants of
  - Sum-product (message passing)
  - Monte Carlo (sampling)
Properties of Graphical Models

- **Significant potential for resolving diversity dilemma**
  - State of the art performance across *symbols, probabilities* and *signals* via uniform representation and reasoning algorithm
    - Representation: My focus has been on *factor graphs*
    - Reasoning: *Sum-product* passes messages about elements of variables’ domains

- **Sum-product algorithm yields**
  - (Loopy) belief propagation in Bayesian networks (BNs)
  - Forward-backward algorithm in hidden Markov models (HMMs)
  - Kalman filters, Viterbi algorithm, FFT, turbo decoding
  - Arc-consistency and production match

- **Many neural network models map onto them**
- **Support both mixed (symbolic & probabilistic) and hybrid (discrete & continuous) processing**
Mixed and Hybrid Processing

- **Symbol processing is norm in cognitive architectures**
  - Generally approached by rules or comparable techniques
- **But much knowledge is uncertain**
  - Architectures may use neural nets or activation-based approaches
  - *Bayesian networks* are state of the art
    - Only recently been combined effectively with general symbol processing, to yield *mixed processing*, and has had little impact so far on cognitive architectures
- **Intelligent systems also need to perceive their worlds**
  - Speech, vision, etc. require extended *signal processing*
  - Also significant progress via graphical *models*
    - *Hidden Markov models* for speech and *Markov random fields* for vision
  - Architectures view this as preprocessing that creates symbols
    - Need *hybrid processing* to use reasoning in perception and perception in reasoning
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
Progress To-Date: Memory Architecture

- Nature of memories used within decision cycle
- Short-term/working and long-term memories
  - Soar 1-8: working memory + production memory
  - ACT-R: buffers + production memory, semantic memory
  - Soar 9: working memory, ST visual imagery + production memory, semantic memory, episodic memory, LT visual memory
- Focus here is on representation and access
  - Haven’t yet got to learning
Goals

- Broadly functional memory architecture
  - Both procedural and declarative knowledge
  - Hybrid: Continuous/signal + discrete-symbolic
  - Mixed: Probabilistic/uncertain + discrete-symbolic

- Uniform implementation

- Provide core for development of full hybrid mixed architecture
  - Melding scope and efficiency with simplicity and elegance
Approach

- Base roughly on Soar 9 and ACT-R
  - Working memory
  - Procedural LT Memory
    - Productions
  - Declarative LT Memory
    - Semantic: Predict unspecified attributes of objects based on specified ones (cues)
    - Episodic: Retrieve best episode based on recency and match to cues
  - Eventually imagery as well, but not yet

- Implement via *graphical models*
  - Layered approach: *graph* and *memory* layers
Graph Layer: Factor Graphs w/ Summary Product

- **Factor graphs** are undirected bipartite graphs
  - Decompose functions: e.g., \( f(u,w,x,y,z) = f_1(u,w,x)f_2(x,y,z)f_3(z) \)
  - Map to variable & factor nodes (with functions in factor nodes)

- **Summary product algorithm** does message passing
  - Output of variable node is *pointwise product* of inputs
  - Output of factor node is pointwise product times factor function
    - And then *summarize out* all variables not on link (by *sum* or *max*)
    - Sum: Marginals of variables (distributions over individual variable domains)
    - Max: Maximum a posteriori estimation (best combination across all variables)
Representation of Functions and Messages

- **N dimensional continuous functions**
  - Approximated as piecewise linear functions over rectilinear regions

- **Span (continuous) signals, (continuous and discrete) probability distributions, symbols**
  - *Discretize domain* for discrete distributions & symbolic
    - [0,1>, [1,2>, [2,3>, …
  - *Booleanize range* (and add symbol table) for symbolic
    - E.g., [0,1]=1 ➔ RED true; [1,2]=0 ➔ GREEN false

<table>
<thead>
<tr>
<th>y\x</th>
<th>[0,10&gt;]</th>
<th>[10,25&gt;]</th>
<th>[25,50&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,5&gt;]</td>
<td>0</td>
<td>.2y</td>
<td>0</td>
</tr>
<tr>
<td>[5,15&gt;]</td>
<td>.5x</td>
<td>1</td>
<td>.1+.2x+.4y</td>
</tr>
</tbody>
</table>
Memory Layer: Distinguish WM and LTM

- **Representation is predicate based**
  - E.g., Object$(s, O1)$, Concept$(O1, c)$
  - Arguments may be constants, or variables (in LTM)

- **Long-term memories compile into graphs**
  - LTM is composed of *conditionals* (generalized rules)
  - Each conditional is a set of predicate patterns and a function

- **WM compiles to evidence at peripheral factor nodes**
  - It is just an N dimensional continuous function where normal symbolic wmes correspond to unit regions with Boolean values

- **Elaboration phase is one settling of graph**
Conditionals

- **Patterns can be conditions, actions or conducts**
  - Conditions and actions embody normal rule semantics
    - Conditions: Messages flow from WM
    - Actions: Messages flow towards WM
  - Conducts embody (bidirectional) constraint/probability semantics
    - Messages flow in both directions: local match + global influence
  - Encoded as (generalized) linear *alpha networks*

- **Pattern networks joined via bidirectional *beta network***
- **Functions are defined over conduct variables**
Additional Details

- **Link directionality** is set independently for each link
  - Determines which messages are sent
- Whether to use *sum* or *max* is specified on an individual variable/node basis
  - Overall algorithm thus mixes sum-product and max-product
- **Variables can be specified as** *unique* or *multiple*
  - Unique variables sum to 1 and use *sum* for marginals: [.1 .5 .4]
  - Multiple variables can have any or all elements valued at 1 and use *max* for marginals: [1 1 0 0 1]
- **Predicates can be declared as** *open world* or *closed world* with respect to matching WM
- **Pattern variables** cause sharing of graph structure
  - May be within a single conditional or across multiple conditionals
Memories

Production Memory

- Just conditions and actions
  - Although may also have a function
- CWA and multiple variables

Semantic Memory

- Just conducts (in pure form)
- OWA and unique variables
- Naïve Bayes (prior on concept + conditionals on attributes)

**CONDITIONAL Transitive**

Condition: \( \text{Next}(a, b) \land \text{Next}(b, c) \)

Action: \( \text{Next}(a, c) \)

**CONDITIONAL ConceptPrior**

Condition: \( \text{Object}(s, O_1) \land \text{Concept}(O_1, c)[\alpha_1] \)

Weight: \( W_{O_1, w} \)

<table>
<thead>
<tr>
<th>( w )</th>
<th>Walker</th>
<th>Table</th>
<th>Dog</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>( [1,10] )</td>
<td>.01w</td>
<td>.001w</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>( [10,20] )</td>
<td>.2-.01w</td>
<td>“</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>( [20,50] )</td>
<td>0</td>
<td>.025-.00025w</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>( [50,100] )</td>
<td>“</td>
<td>“</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Memories (cont.)

Episodic Memory
- Just conducts (in pure form)
- OWA and unique variables
- Exponential prior on time + conditionals on episode attributes

Constraint Memory
- Just conducts (in pure form)
- OWA and multiple variables

\[ \text{Conditioned TimeConcept} \]
\[ \text{Conduction: Time}(t)[\alpha_3] \]
\[ \text{Concept}(O_1, c) \]

\[
\begin{array}{|c|c|c|c|c|}
\hline
\&c & Walker & Table & Dog & Human \\
\hline
1 & 1 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 \\
3 & 0 & 0 & 0 & 1 \\
4 & 0 & 0 & 1 & 0 \\
\hline
\end{array}
\]

\[ \text{CONDITIONAL TwoColorConstraint12} \]
\[ \text{Conduction: Color}(R_1, c_1)[\alpha_7] \]
\[ \text{Color}(R_2, c_2)[\alpha_8] \]

\[
\begin{array}{|c|c|c|}
\hline
\&c2 & Red & Blue \\
\hline
\text{Red} & 0 & 1 \\
\text{Blue} & 1 & 0 \\
\hline
\end{array}
\]

\[ \text{CONDITIONAL TimePrior} \]
\[ \text{Conduction: Time}(t)[\alpha_3] \]

\[
\begin{array}{|c|c|c|c|c|}
\hline
0 & 1 & 2 & 3 & 4 \\
\hline
0 & .032 & .087 & .237 & .644 \\
\hline
\end{array}
\]
Key Similarities and Differences

**Similarities**
- All based on WM and LTM
- All LTM based on conditionals
- All conditionals map to graph
- Processing by summary product

**Differences**
- Procedural vs. declarative
  - Conditions/actions vs. conducts
  - Directionality of message flow
  - Closed vs. open world
  - Multiple vs. unique variables
- Semantic vs. episodic
  - Marginal/sum vs. MAP/max
  - Condition on concept vs. time
  - General probs. vs. instances

**Constraints are actually hybrid: conducts, OWA, multiple**

*Other variations and hybrids are also possible*
Goals for Visual Representation and Reasoning

- **Represent and reason about 3D imagery**
  - Represent structure and location of objects
  - Deal with uncertainty about location
  - Implement basic transformations (translation, rotation, etc.)

- **Act as an intermediary between “peripheral” signal processing and “central” symbol processing**
  - Integrate tightly with perception and cognition
  - Imagery assists perception and cognition, and vice versa

- **Implement uniformly with perception and cognition**
  - Base imagery on factor graphs and summary product

- **Limit initial focus to symbol-level issues**
  - How imagery works and interacts with perception and cognition
  - Not how content is translated across modalities
Core: 3D Imagery Buffer

- Build on function/message representation
- Currently nD continuous functions approximated as piecewise linear functions over rectilinear regions
  - Generalization of standard voxel 3D representation
- Other variations also conceivable
  - Generalize rectilinear regions to convex polytopes
  - Shift from linear to (sum of) Gaussian functions

<table>
<thead>
<tr>
<th>$y \land x$</th>
<th>[0,10&gt;</th>
<th>[10,25&gt;</th>
<th>[25,50&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,5&gt;</td>
<td>0</td>
<td>.2y</td>
<td>0</td>
</tr>
<tr>
<td>[5,15&gt;</td>
<td>.5x</td>
<td>1</td>
<td>.1+.2x+.4y</td>
</tr>
</tbody>
</table>

http://www.drubu.com/tutorial/voxels.html
Performing Imagery Transformations Graphically

- Much state-of-the-art perceptual processing occurs via Markov (and conditional) random fields
- Can these be adapted for graphical implementation of imagery transformations?
  - E.g., with ideas from stereo vision and sequence prediction?
- Can this be efficient via a uniform implementation of factor graphs with summary product?
- Will this enable tight coupling and interpenetrability with perception and cognition?
  - Combine perception with imagery-based prediction for Kalman filter-like behavior?
  - Use knowledge to control imagery and imagery to aid cognition?
Summary and Future

- **GMs show potential for resolving diversity dilemma**
  - A uniform approach/implementation for architectural diversity
    - Combining simplicity with broad functionality (and efficiency?)
    - *Mixed* and *hybrid* representation and processing
    - So far yielded a graphical memory architecture
- **Working towards a mixed, hybrid variant of Soar 9**
  - Leverage mixed processing for decisions, reflection, learning, etc.
  - Leverage hybrid processing for imagery (and perception)
- **Much continued room to grow past Soar 9**
  - Understand, simplify and hybridize across existing architectures
  - Build radically new architectures with novel capabilities