USC Institute for Creative Technologies

University of Southern California

# Σ The Sigma Cognitive Architecture/System

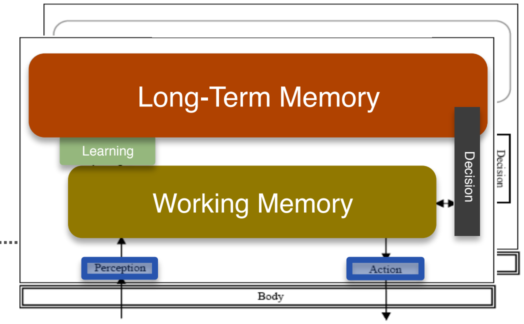

Paul S. Rosenbloom
& the Sigma Group


4.19.2016

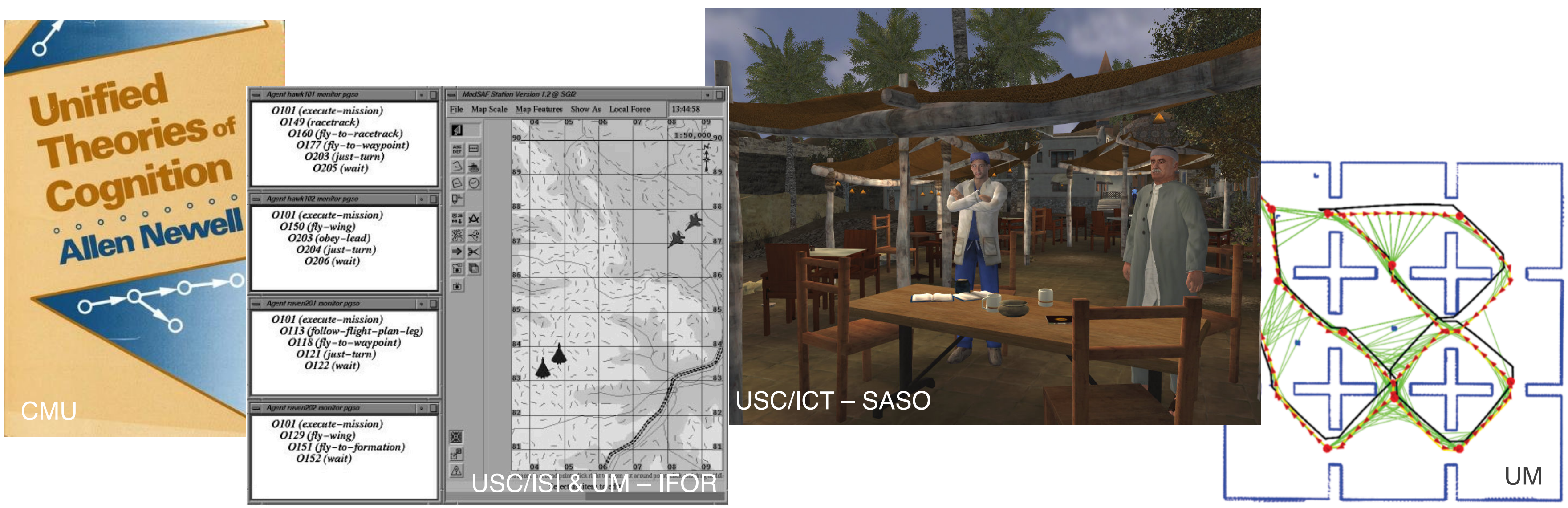

The work depicted here was sponsored by the U.S. Army, the Air Force Office of Scientific Research and the Office of Naval Research. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.


ARL

# Cognitive Architecture




- Model of the fixed structure of a/the mind
  - Memory, reasoning, learning, interaction, ...
  - Integration across these capabilities
- Supports knowledge and skills above the architecture



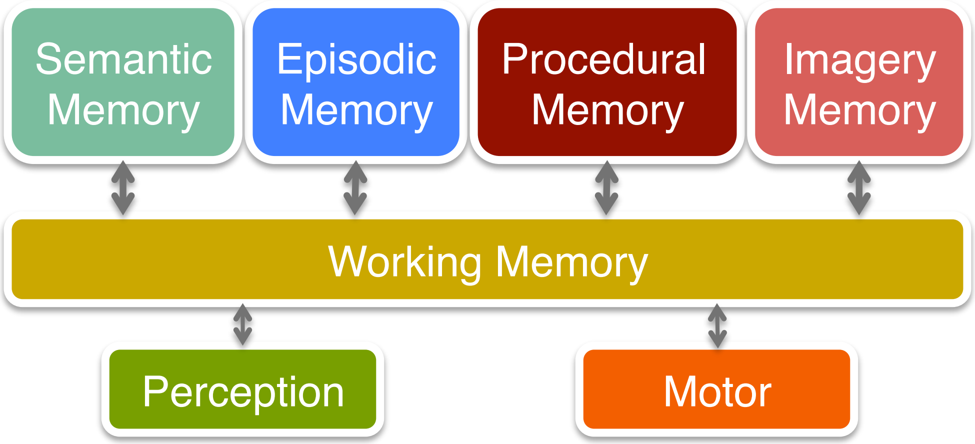

**Examples are from Soar**

# Overall Desiderata for the Sigma (Σ) Architecture

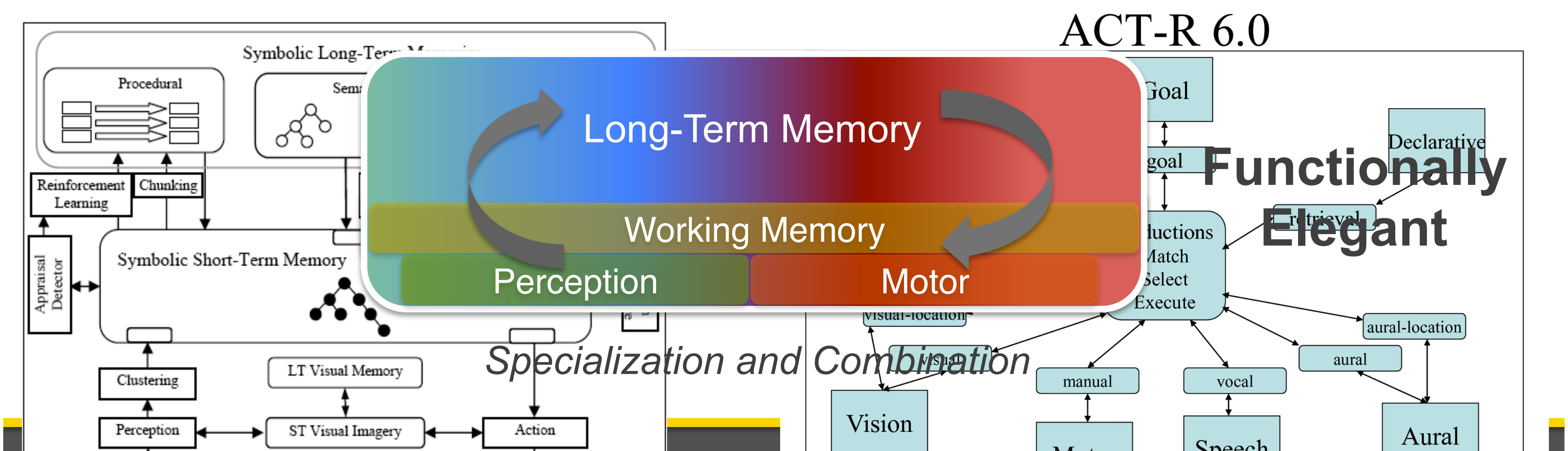
- A new breed of cognitive architecture that is
  - *Grand unified*
    - Cognitive + key non-cognitive (perceptuomotor, affective, attentive, …)
  - *Generically cognitive*
    - Spanning both natural and artificial cognition
  - *Functionally elegant*
    - Broadly capable yet simple and theoretically elegant
      - "cognitive Newton's laws"
  - *Sufficiently efficient*
    - Fast enough for anticipated applications
- For virtual humans & intelligent agents/robots that can
  - Think – Broadly, deeply and robustly *cognitive*
  - Behave – *Interactive* with their physical and social worlds
  - Learn – *Adaptive* given their interactions and experience

*Hybrid*: Discrete + Continuous
*Mixed*: Symbolic + Probabilistic

# Modular versus Functionally Elegant




**Modular**




**Functionally Elegant**

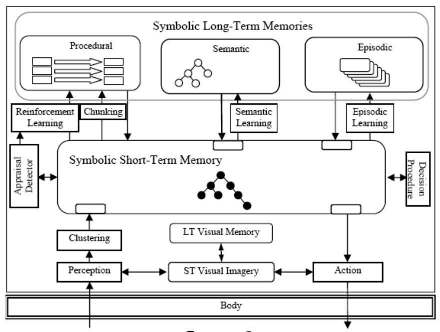
*Specialization and Combination*

**Goal: Advancing *elegance, depth* and *breadth***

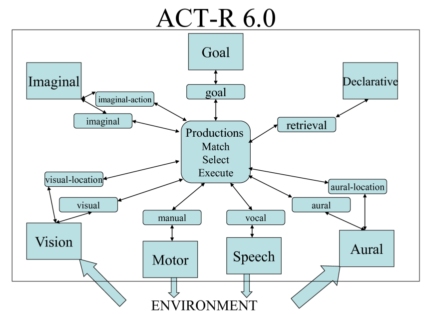
# Approach: Graphical Architecture Hypothesis

Key to success is *blending what has been learned from over three decades of independent work* in **cognitive architectures** and **graphical models**

**Cognitive Architectures**



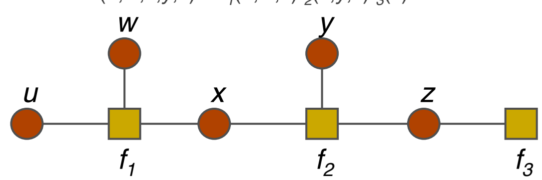

**Soar 9**




**+**

**Graphical Models**

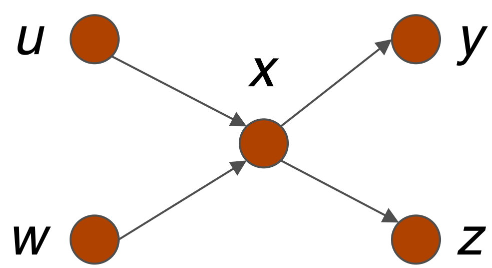
$f(u,w,x,y,z) = f_1(u,w,x)f_2(x,y,z)f_3(z)$



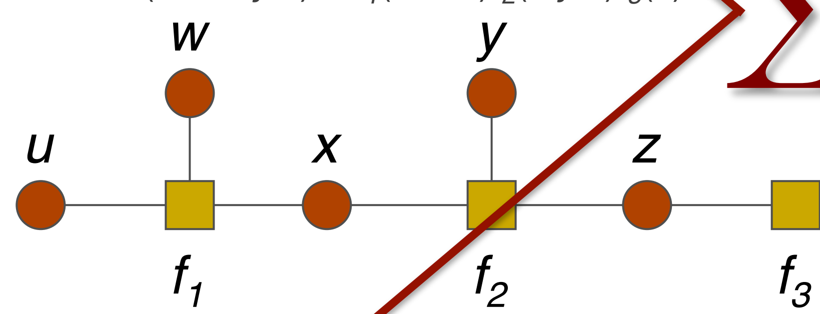
# Graphical Models

- Efficient computation over multivariate functions by leveraging forms of independence to decompose them into products of simpler subfunctions
  - Bayesian/Markov networks, Markov/conditional random fields, factor graphs

$p(u,w,x,y,z) = p(u)p(w)p(x|u,w)p(y|x)p(z|x)$




$f(u,w,x,y,z) = f_1(u,w,x)f_2(x,y,z)f_3(z)$



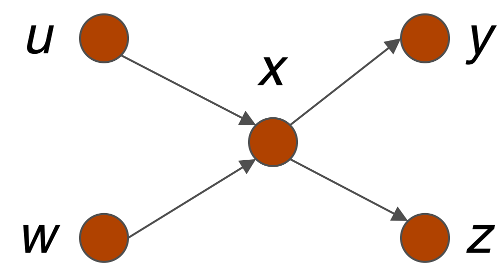

$\Sigma$

- Solve typically via some form of message passing or sampling
- State of the art performance across *symbols*, *probabilities* and *signals* from uniform representation and reasoning algorithm
  - (Loopy) belief propagation, forward-backward algorithm, Kalman filters, Viterbi algorithm, FFT, turbo decoding, arc-consistency, production match, …
- Can support mixed and hybrid processing
- Several neural network models map directly onto them

# Bayesian Network vs. Factor Graph

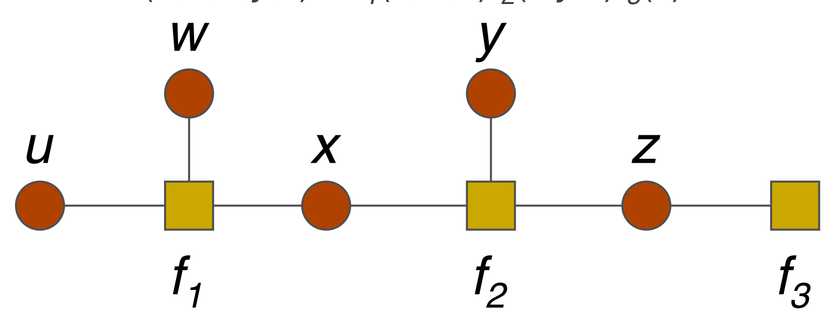
- Bayesian network
  - Directed graph
  - Only variable nodes
  - A distribution at each node *n*
    - $p(n \mid parents_n)$
  - Decompose probabilities

$p(u,w,x,y,z) = p(u)p(w)p(x|u,w)p(y|x)p(z|x)$




- Factor graph
  - Undirected graph
  - Variable and factor nodes
  - A function at each factor node *n*
    - $f_n(vs_n)$
  - Decompose arbitrary functions

$f(u,w,x,y,z) = f_1(u,w,x)f_2(x,y,z)f_3(z)$



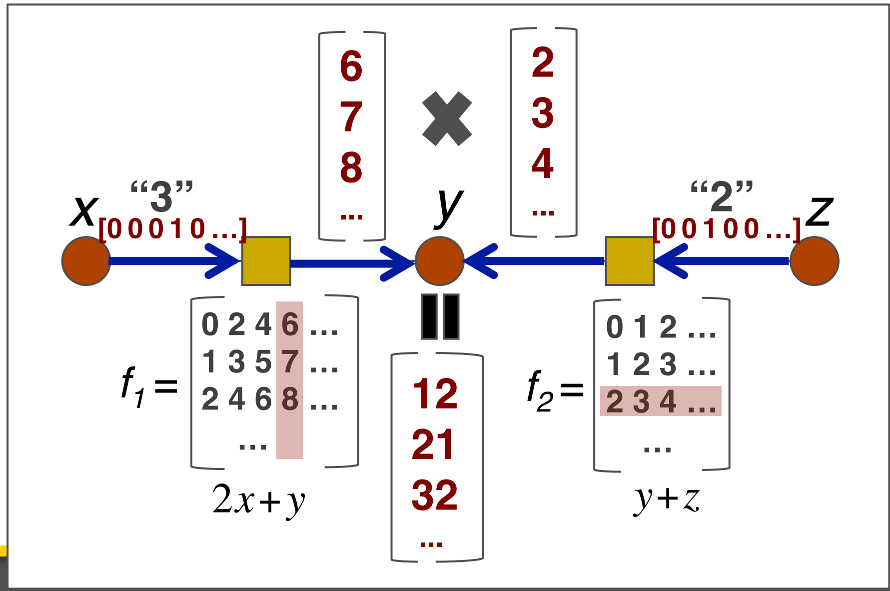
# Summary Product Algorithm

- Compute variable marginals (*sum-product*/*integral-product*) or mode of entire graph (*max-product*)
- Pass messages on links and process at nodes
  - Messages are distributions over link variables (starting w/ *evidence*)
  - At variable nodes messages are combined via *pointwise product*
  - At factor nodes do products, and summarize out unneeded variables:

$$m(y) = \int_x m(x) \times f_1(x,y)$$

$$f(x,y,z) = y^2 + yz + 2yx + 2xz$$
$$= (2x+y)(y+z) = f_1(x,y) f_2(y,z)$$

In Sigma, both functions and messages are piecewise linear



# Piecewise Linear Functions

- Unified representation for *continuous*, *discrete* and *symbolic* data
- At base have multidimensional continuous functions
  - One dimension per variable, with multiple dimensions providing *relations*
  - Approximated as *piecewise linear* over *arrays/tensors of regions*
- *Discretize domain* for discrete distributions (& symbols)
- *Booleanize range* (and add symbol table) for symbols

  Color($O_1$, Brown) & Alive($O_1$, T)
- Dimensions/variables are *typed*

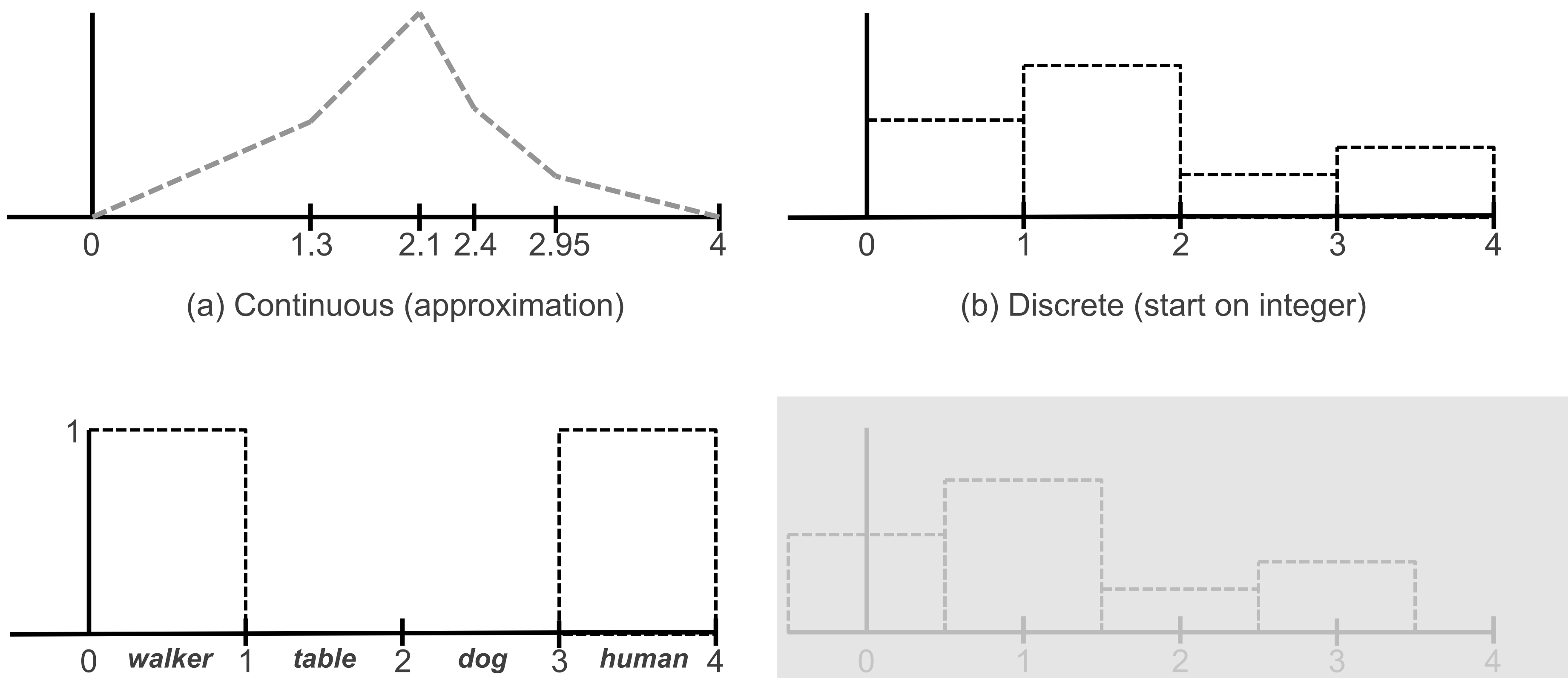
| $O_1$ | Brown | Silver | White |
|---|---|---|---|
| T | 1 | 0 | |
| F | 0 | | |

| P(*legs* \| *concept*) | Walker | Table | … |
|---|---|---|---|
| 1 | 0 | 0 | … |
| 2 | 0 | 0 | … |
| 3 | 0 | .1 | … |
| 4 | 1 | .9 | … |

| P(*weight* \| *concept*) | Walker | Table | … |
|---|---|---|---|
| [1,10> | .01*w* | .001*w* | … |
| [10,20> | .2-.01*w* | " | … |
| [20,50> | 0 | .025-.00025*w* | … |
| [50,100> | " | " | … |

**Analogous to implementing digital circuits by restricting an inherently continuous underlying substrate**

# Piecewise Linear Functions




(a) Continuous (approximation)

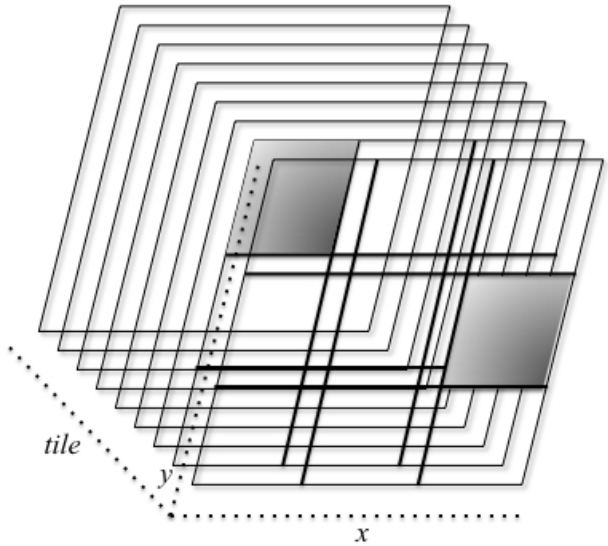
(b) Discrete (start on integer)

(d) Symbolic

(c) Discrete (center on integer)

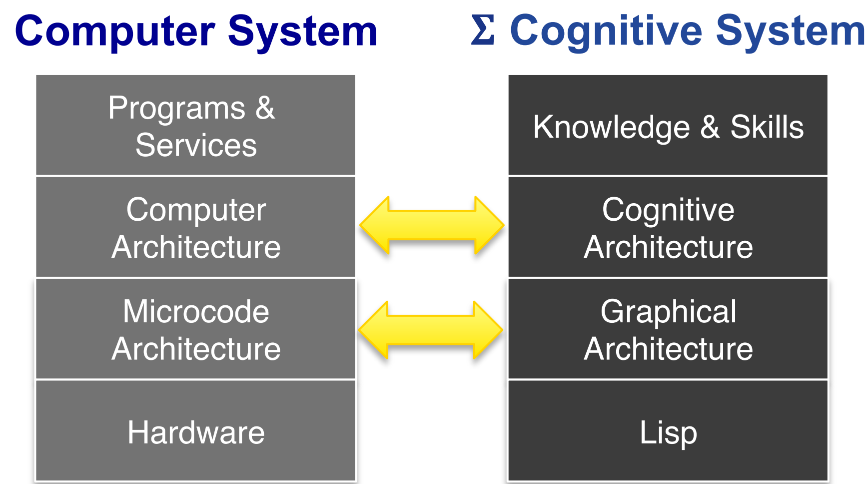
# The Eight Puzzle


- Classic sliding tile puzzle
- Represented symbolically in standard AI systems
  - **LeftOf**(*cell*$_{11}$, *cell*$_{21}$), **At**(*tile*$_1$, *cell*$_{11}$), etc.
- Instead represent as a hybrid 3D function
  - Continuous spatial *x* & *y* dimensions
    - `dimension[0-3)`
  - Discrete *tile* dimension (an *xy* plane)
    - `tile[0:9)`
  - Region of plane with tile has value 1
    - All other regions have value 0

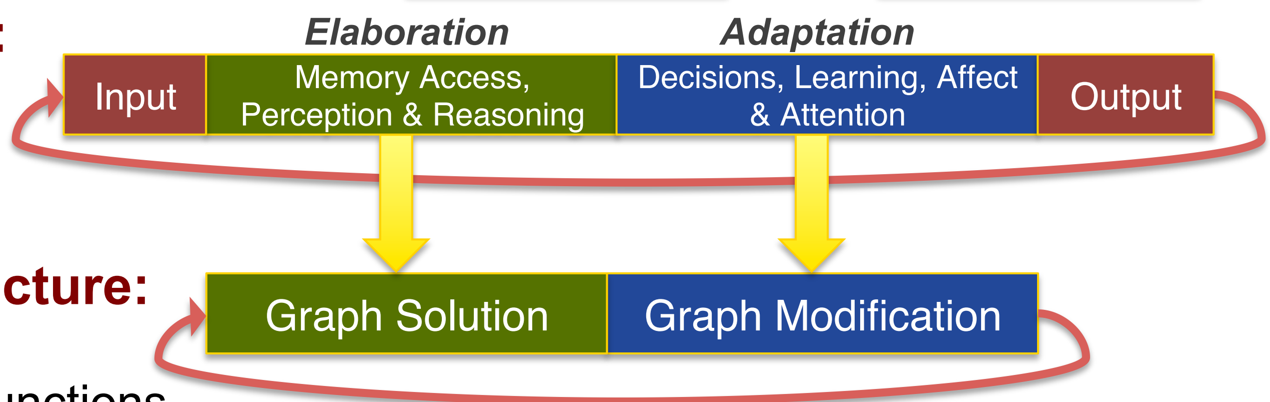


# The Structure of Sigma

Σ




**Cognitive Architecture:**
- Predicates
- Conditionals
- Nested tri-level control

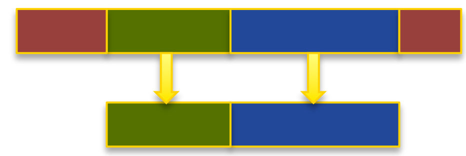



**Graphical Architecture:**
- Graphical models
- Piecewise linear functions
- Gradient-descent learning

# (Soar-like) Nested Tri-Level Control

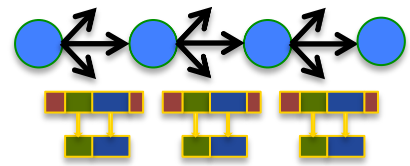
- **A (parallel) *reactive* layer**
  - Single graph/cognitive cycle

*Which acts as the inner loop for*

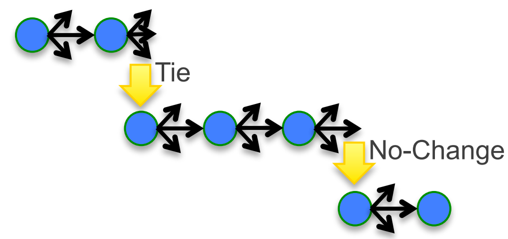


- **A (serial/iterative) *deliberative* layer**
  - Repeated operator selection & application

*Which acts as the inner loop for*



- **A (recursive) *reflective* layer**
  - Impasse-driven meta-level processing




- Maps onto bi-/tri-level models in
  - Cognitive Psychology (automatic vs. controlled, System 1 vs. 2, …)
  - Robotics (3T, …)
  - Emotion modeling

# Fundamental Questions about Sigma

- Can full range of capabilities be provided in this manner?
- Can it all be sufficiently efficient for real time behavior?
- What are the functional gains?
- Can the human mind (and brain) be modeled?

Winner of **Kurzweil Award** at AGI 2011 & 2012

# Overall Progress on Sigma

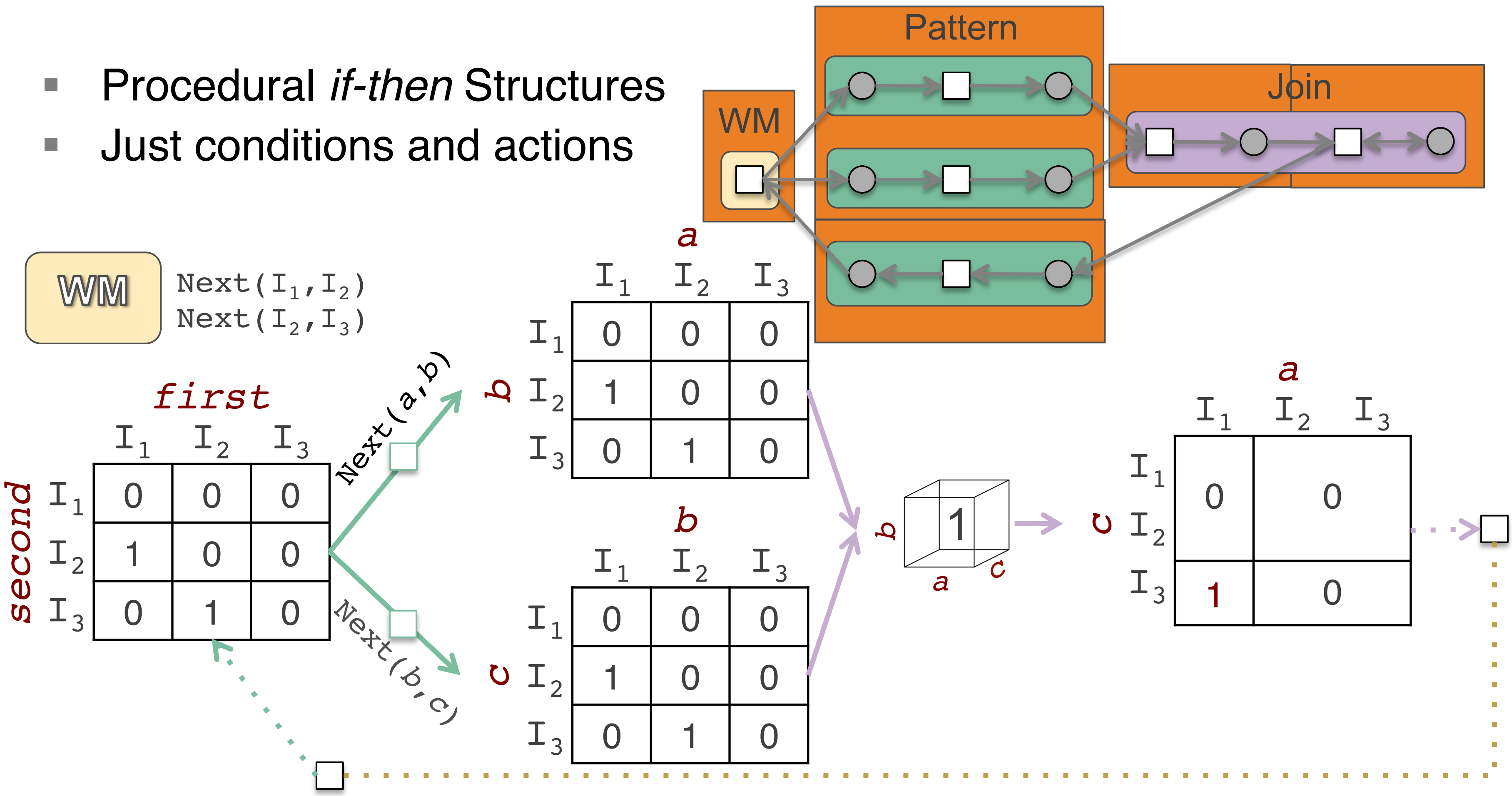
- Memory [ICCM 10]
  - Procedural (rule)
  - Declarative (semantic/episodic) [CogSci 14]
  - Constraint
  - Distributed vectors [AGI 14a]
  - Perceptual [BICA 14a, AGI 15]
  - Neural network
- Problem solving
  - Preference based decisions [AGI 11]
  - Impasse-driven reflection [AGI 13]
  - Decision-theoretic (POMDP) [BICA 11b]
  - Theory of Mind [AGI 13, AGI 14b]
- Learning [ICCM 13]
  - Concept (supervised/unsupervised)
  - Episodic [CogSci 14]
  - Reinforcement [AGI 12a, AGI 14b]
  - Action/transition models [AGI 12a]
  - Models of other agents [AGI 14b]
  - Perceptual (including maps in SLAM)
- Efficiency [ICCM 12, BICA 14b]
- Mental imagery [BICA 11a, AGI 12b]
  - 1-3D continuous imagery buffer
  - Object transformation
  - Feature & relationship detection
- Perception
  - Object recognition (CRFs) [BICA 11b]
  - Spoken word recognition (HMMs) [BICA 14a]
  - Localization [BICA 11b]
- Natural language
  - Word sense disambiguation [ICCM 13]
  - Part of speech tagging [ICCM 13]
  - Sentence identification [WS 15]
  - Dialogue [WS 15]
- Affect [AGI 15]
  - Appraisal (expectedness, desirability)
  - Attention (perceptual, cognitive)
- Integration
  - CRF+Localization+POMDP [BICA 11b]
  - Rules+SLAM+RL+ToM+VH [IVA 15, WS 15]
  - SentenceID+Dialogue [WS 15]

# MEMORY AND DECISIONS

```
CONDITIONAL Transitive
   Conditions: Next(a,b)
               Next(b,c)
   Actions: Next(a,c)
```

# Procedural Memory (Rules)

- Procedural *if-then* Structures
- Just conditions and actions

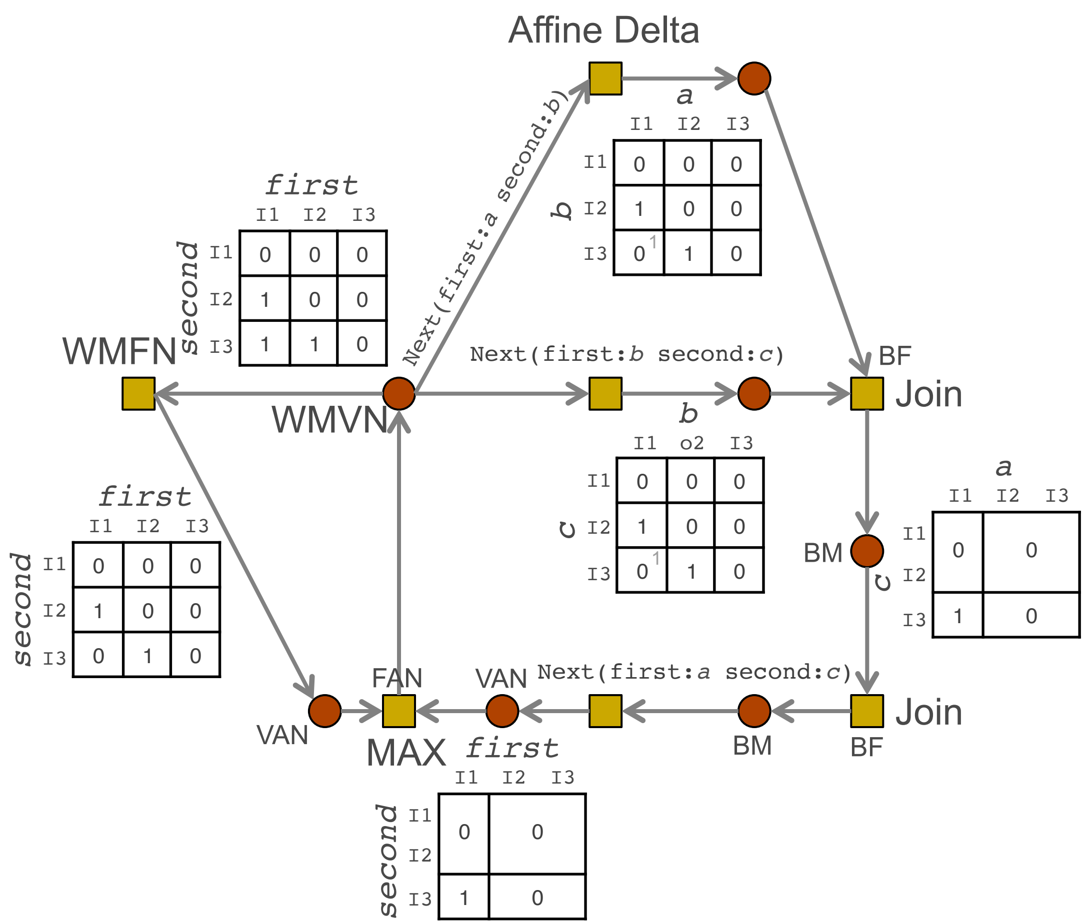


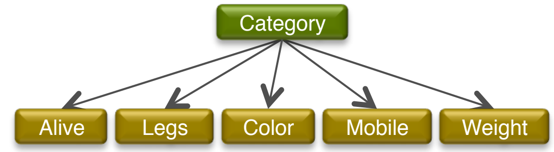

```
(type 'ID :constants '(I1 I2 I3))
(predicate 'Next '((first ID) (second ID)) :world 'closed)
```

# Procedural Memory (Rules)

## In More Detail

```
CONDITIONAL Transitive
   Conditions: Next(a,b)
               Next(b,c)
   Actions: Next(a,c)
```



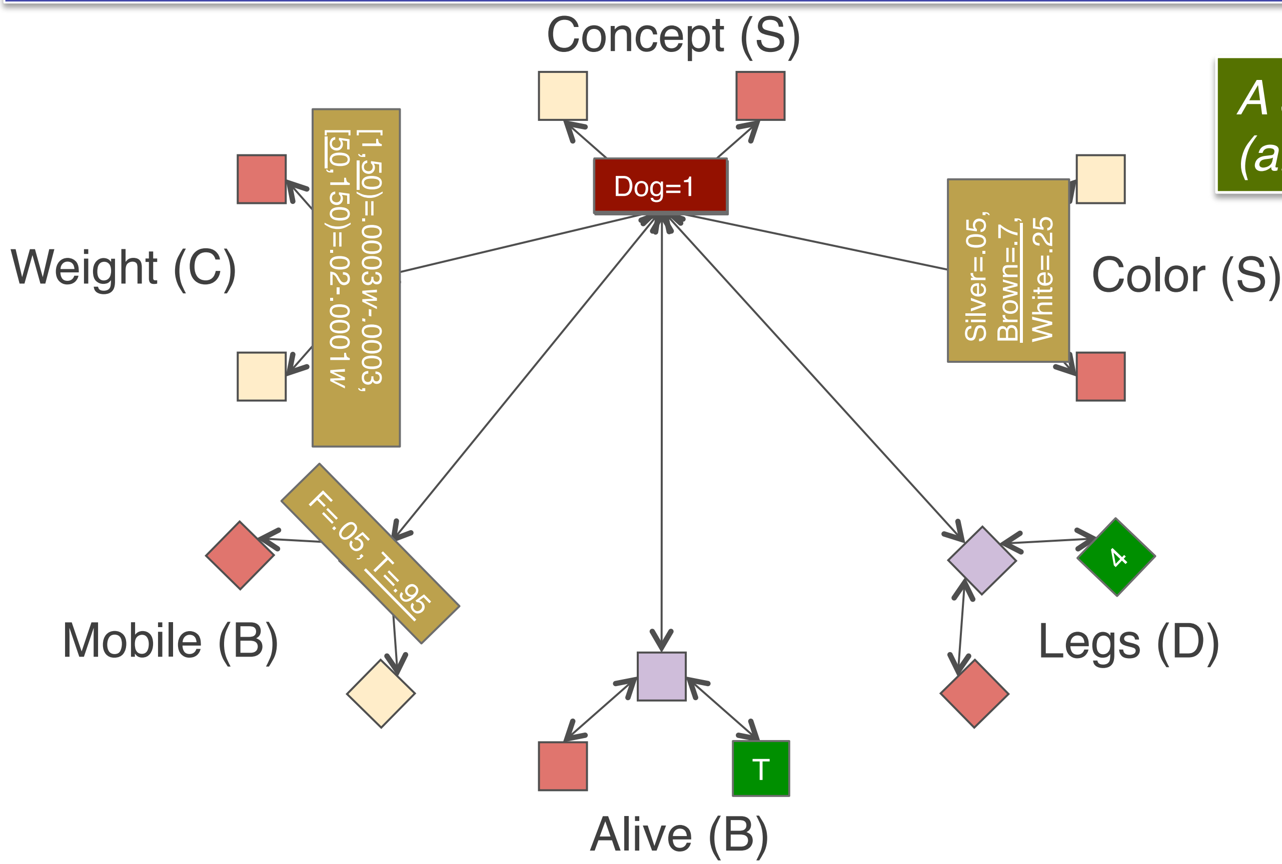


Naïve Bayes classifier

# Semantic Memory (Classifier)

**Given** cues, **retrieve/predict** object category and missing attributes

E.g., **Given** *Alive*=T & *Legs*=4 **Retrieve** *Category*=Dog, *Color*=Brown, *Mobile*=T, *Weight*=50



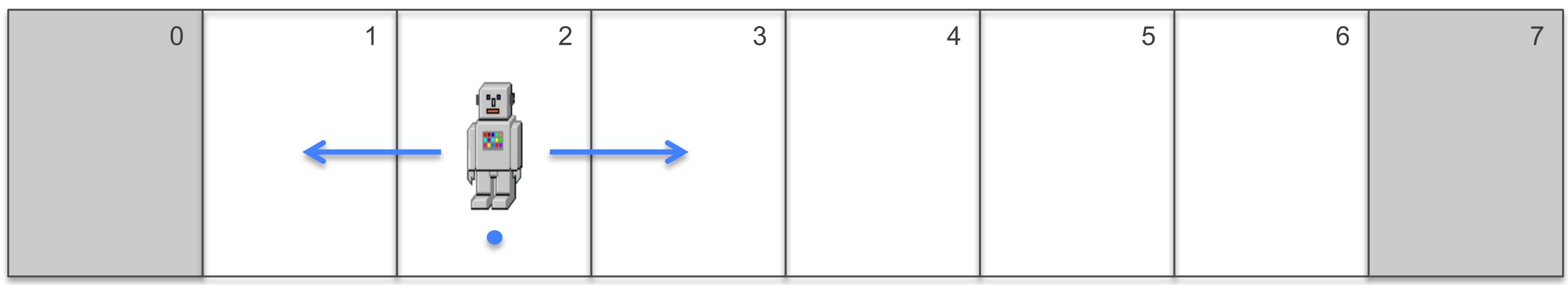

*A subset of factor nodes (and no variable nodes)*

Function
Perception
Join

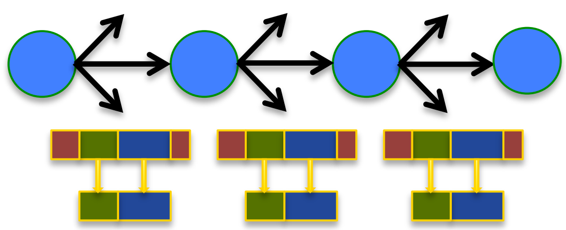
B: Boolean
S: Symbolic
D: Discrete
C: Continuous

# Random Walk on 1D Grid

```
CONDITIONAL RW-1D
   Actions: Selected(left)
            Selected(right)
            Selected(none)
```




- 1D Grid of discrete cells
- Agent can move one cell to left or right, or stay where is



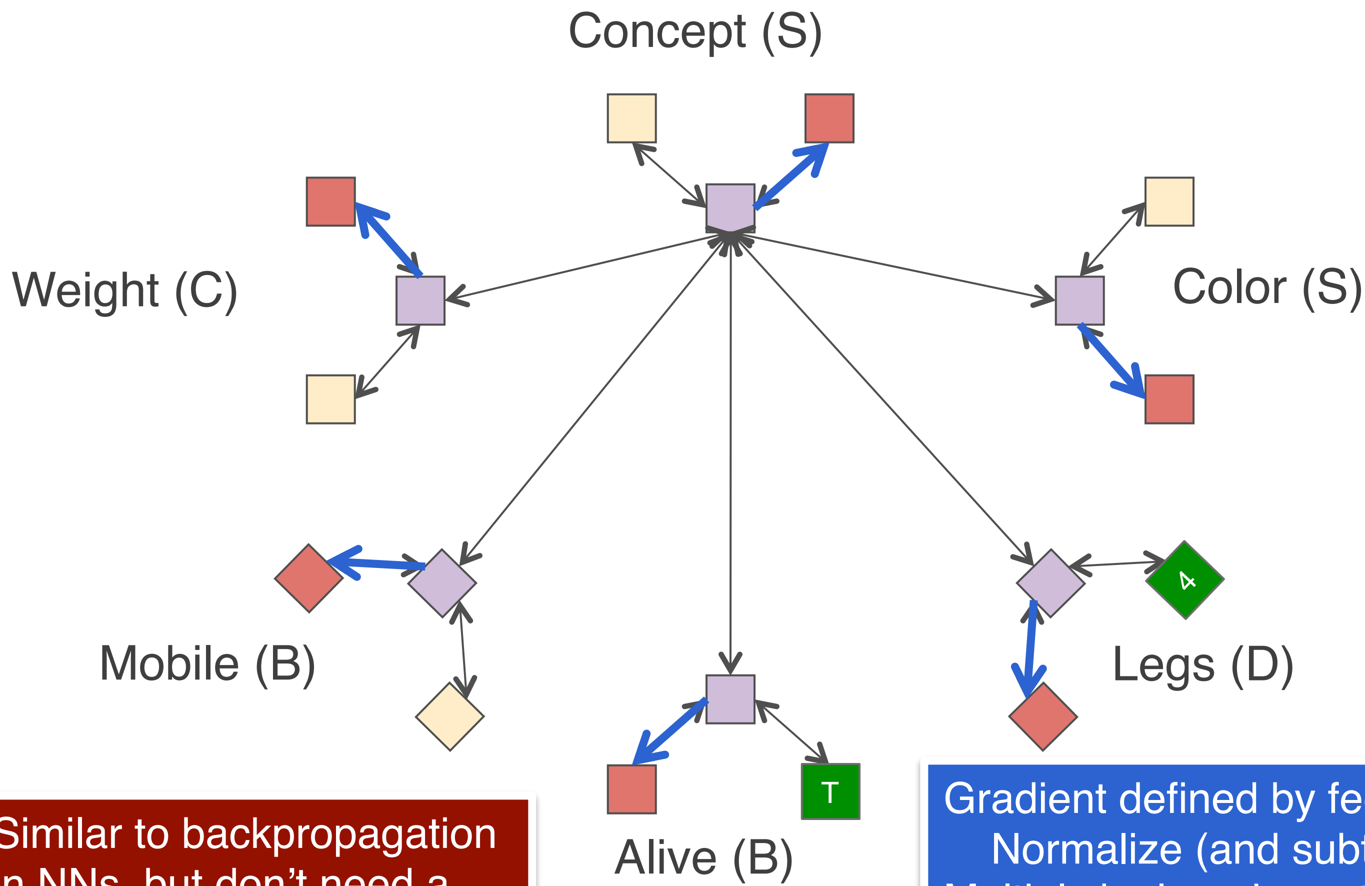
# Replicating a Virtual Human "Mind"

- Immersive Naval Officer Training System (INOTS)
  - Targets leadership and basic counseling for junior Navy leaders
  - Trained over 5000 sailors since 2012


- INOTS "mind" based on two tools
  - Statistical query-answering tool (NPCEditor)
  - Transition diagram for dialogue management
- Both aspects reimplemented and integrated together in Sigma
  - Query answering via semantic memory (*reactive*)
  - Dialogue management by sequences of operators (*deliberative*)

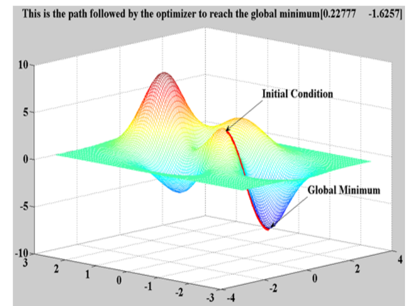
# LEARNING

# Parameter Learning via Gradient Descent




Gradient descent

Local, incremental search for optimal weights




Similar to backpropagation in NNs, but don't need a separate backprop phase

Gradient defined by feedback to function node
- Normalize (and subtract out average)

Multiply by learning rate

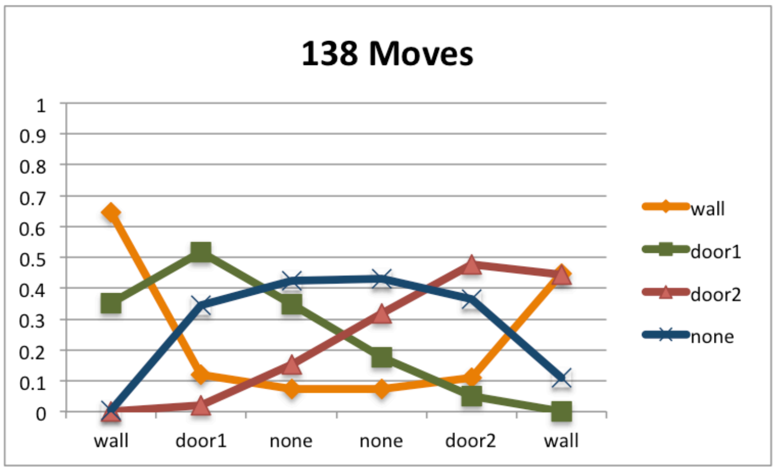
Add to function, smooth and normalize

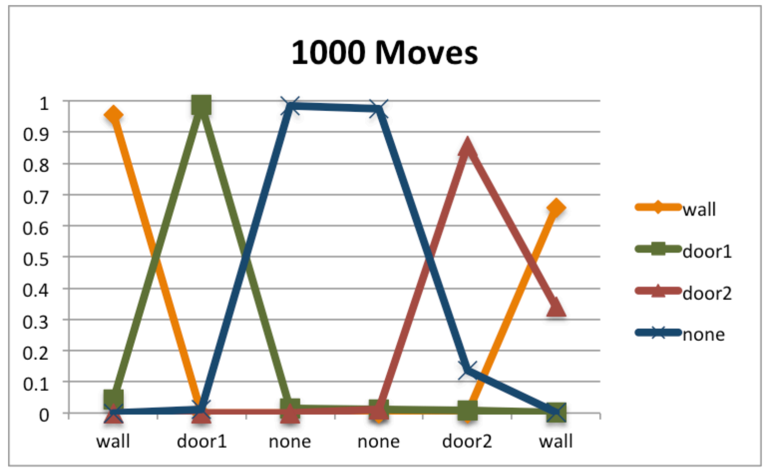
# Learning Maps

- Map: **P**(Objects | Locations)

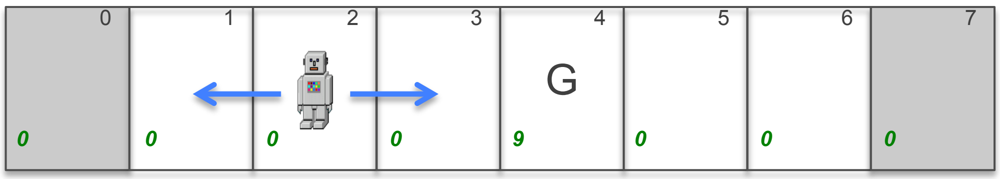
```
CONDITIONAL Object-Location-Map
    Conditions: Object(value:o)
    Condacts:   Location(x:x)
    Function(x,o): .25
```

| wall | door1 | | | door2 | wall |
|---|---|---|---|---|---|



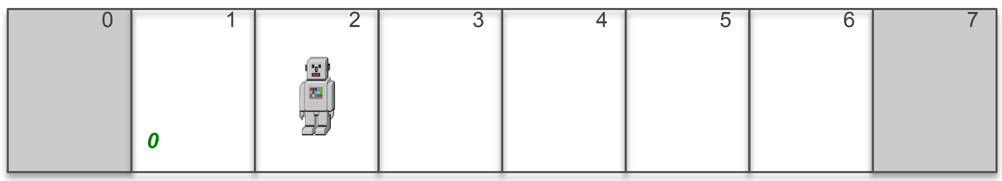
# Reinforcement Learning




Learn values of actions for states by backwards propagation of rewards received during exploration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

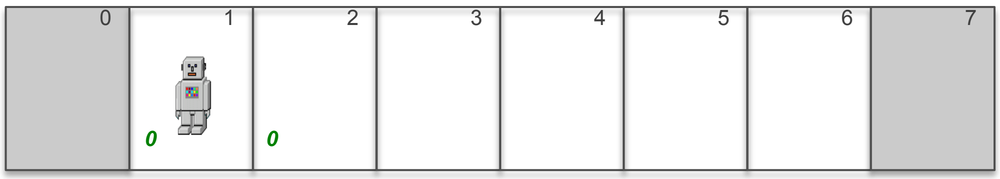
# Reinforcement Learning




Learn values of actions for states by backwards propagation of rewards received during exploration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

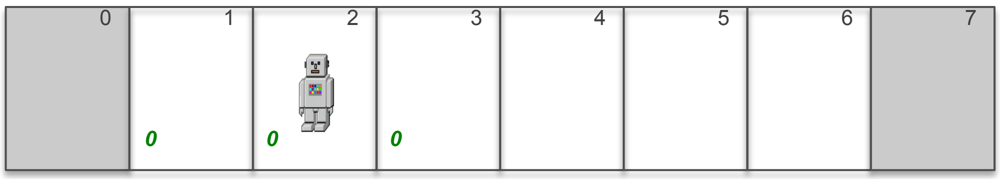
# Reinforcement Learning




Learn values of actions for states by backwards propagation of rewards received during exploration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

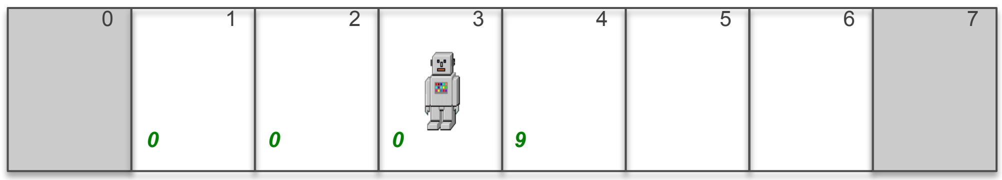
# Reinforcement Learning




Learn values of actions for states by backwards propagation of rewards received during exploration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

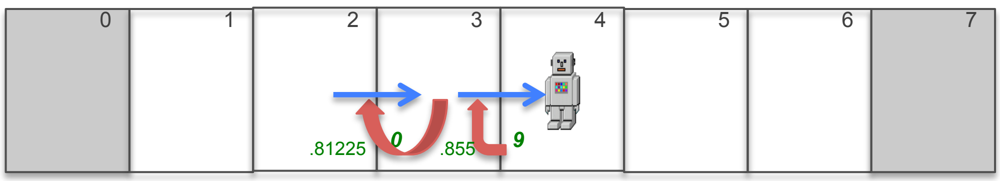
# Reinforcement Learning




Learn values of actions for states by backwards propagation of rewards received during exploration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

# Reinforcement Learning

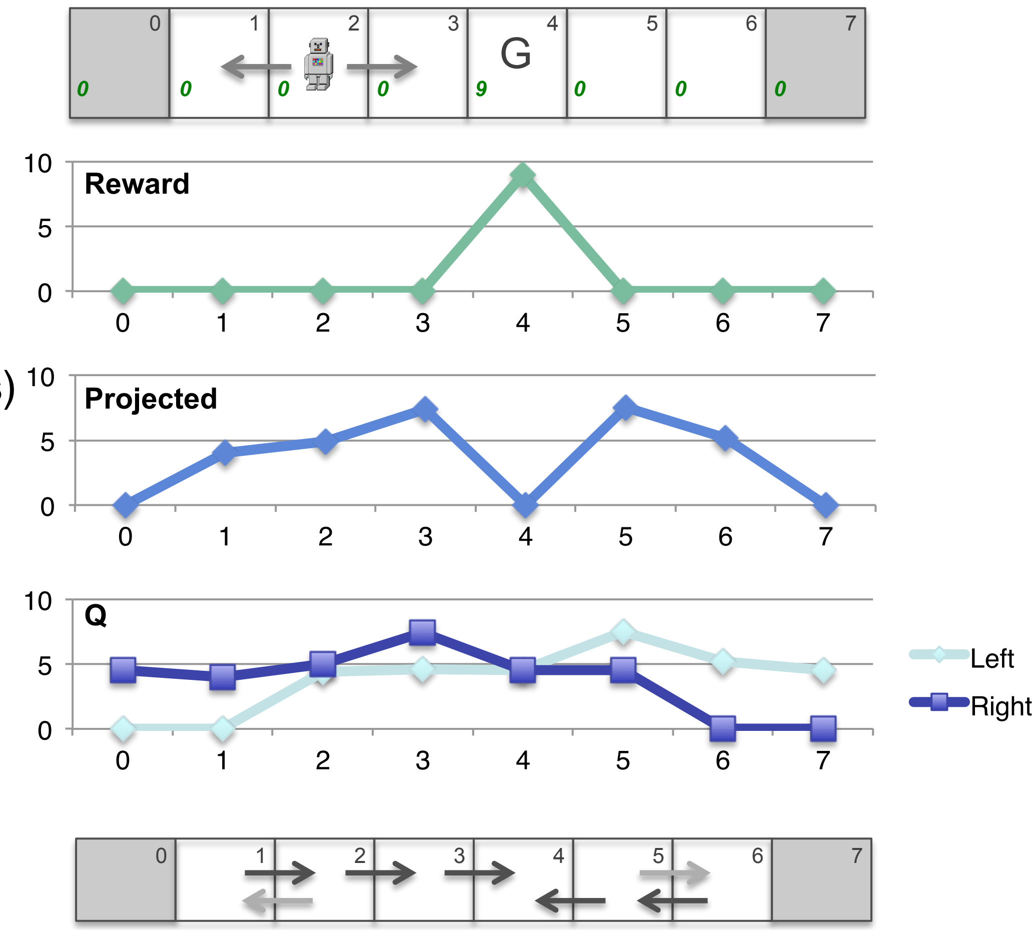



Learn values of actions for states by backwards propagation of rewards received during exploration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

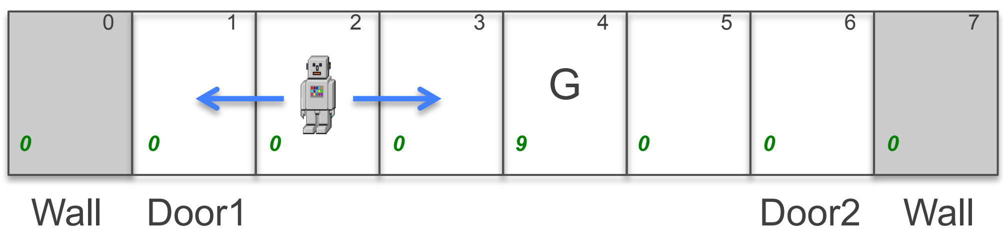
# Deconstructing RL in Sigma

- Knowledge:
  - Initial uniform predictors for:
    - Current reward (*R*)
    - Projected future reward (*P*)
    - Action preferences (*Q*)
  - Regression (backup) knowledge
  - Action models (predict next states)
- Supervised learning of:
  - Current reward (*R)*
  - Projected future reward (*P)*
  - Action preferences (*Q*)
- Add *Diachronic cycles* to also learn action models




Graphs are of expected values, but learning is actually of full distributions

# Integration




- Determine location in corridor
- Map corridor
- Learn to go to goal location in corridor
- Learn to model action effects

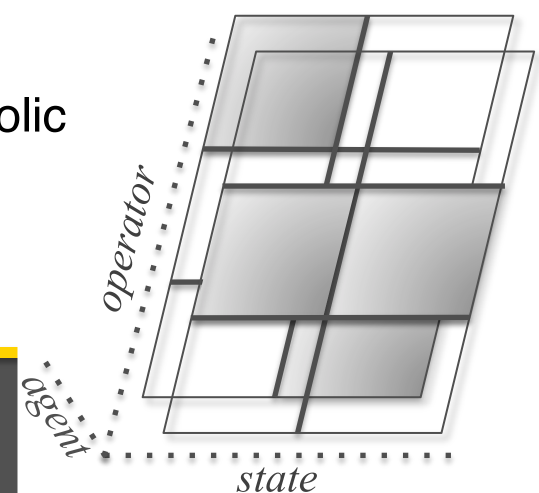
SLAM

RL

# THEORY OF MIND
# (& MULTIAGENT SYSTEMS)

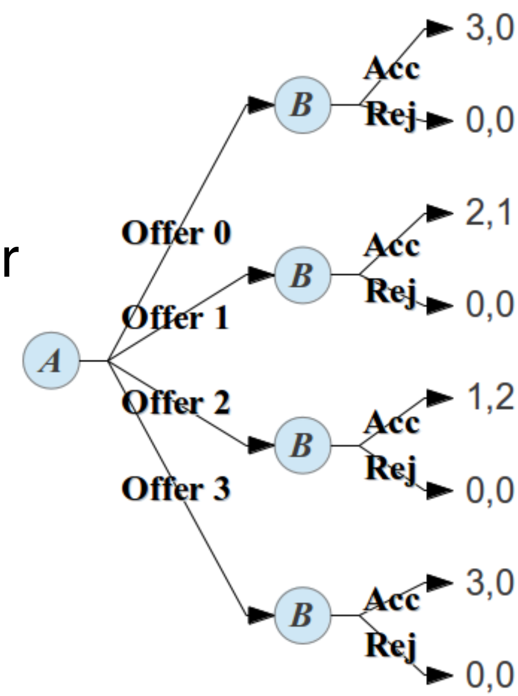
# Theory of Mind (ToM) in Sigma

- ToM models the minds of others, to enable for example:
  - Understanding multiagent situations
  - Participating in social interactions
- ToM approach based on *PsychSim* (Marsella & Pynadath)
  - Decision theoretic problem solving based on POMDPs
  - Recursive agent modeling
- Multiagent Sigma: Add agent argument to predicates
  - E.g., Selected(*agent*, *operator*, *state*)
  - A discrete dimension, but may be numeric or symbolic



# Sequential Games

- Players (*A*, *B*) alternate moves
  - E.g., *Ultimatum*, *centipede* and *negotiation*
- Ultimatum game
  - *A* starts with a fixed amount of money (3)
  - *A* decides how much (in 0-3) to offer *B*
  - *B* decides whether or not to accept the offer
    - If *B* accepts, each gets the resulting amount
    - If *B* rejects, both get 0
  - Each has a utility function over money
    - E.g., <.1, .4, .7, 1>



# Solving Sequential Games in Sigma

- Decision-theoretic approach with *softmax combination*
  - Use expected value at each level of search
  - Action **P**s assumed exponential in their utilities (à la Boltzmann)
- There may be many *Nash equilibria*
- Instead seek stricter concept of *subgame perfection*
  - Overall strategy is an equilibrium strategy over any subgame
- **Key result:** Games solvable in two modes:
  - Reactive/automatic/system-1
  - Reflective/controlled/system-2

  *Both modes well documented in humans for general processing*

  *Combination not found previously in ToM models*

# Reactive/Automatic Approach

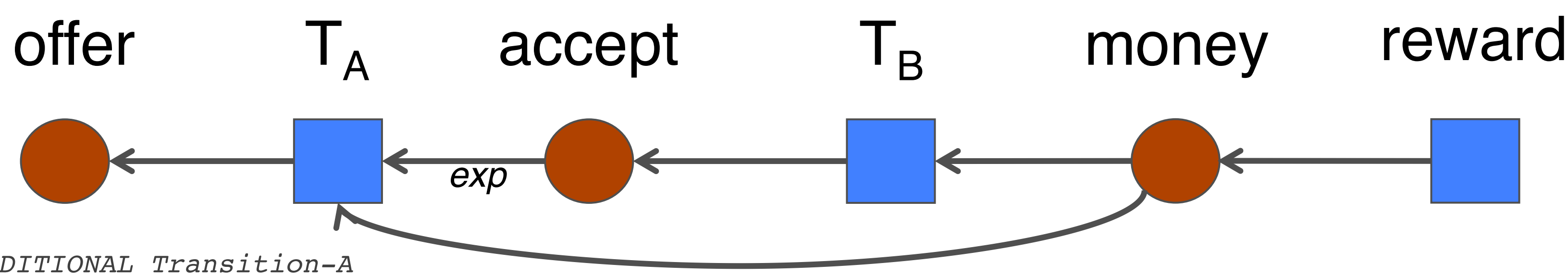
- A trellis (factor) graph in LTM with one stage per move
- Focus on backwards messages from reward(s)

```
CONDITIONAL Transition-B
  Conditions: Money(agent:B quantity:moneyb)
  Condacts: Accept(offer:offer acceptance:choice)
  Function(choice,offer,moneyb): 1<T,0,0>, 1<T,1,1>, 1<T,2,2>,
                                 1<T,3,3>, 1<F,*,0>
```

$$V_B(\mathit{offer},\mathit{choice}) = \mathrm{Reward}_B(\mathrm{Money}_B(\mathit{offer},\mathit{choice}))$$

$$V_A(\mathit{offer}) = \sum_{\mathit{choice}} \mathrm{Reward}_A(\mathrm{Money}_A(\mathit{offer},\mathit{choice})) * e^{K * V_B(\mathit{offer},\mathit{choice})}$$

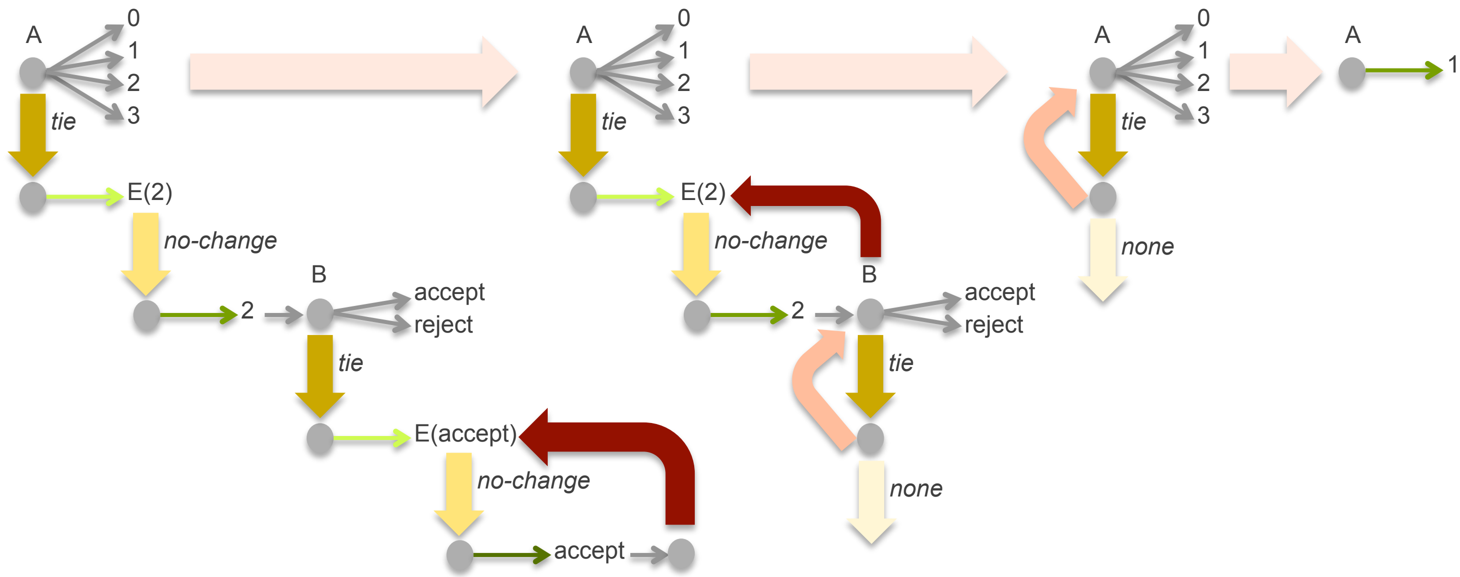
```
CONDITIONAL Reward
  Condacts: Money(agent:agent quantity:money)
  Function(agent,money): .1<*,0>, .4<*,1>,
                         .7<*,2>, 1<*,3>
```




```
CONDITIONAL Transition-A
  Conditions: Money(agent:A quantity:moneya)
              Accept-E(offer:offer acceptance:choice)
  Condacts: Offer(agent:A quantity:offer)
  Function(choice,offer,moneya): 1<T,0,3>, 1<T,1,2>, 1<T,2,1>,
                                 1<T,3,0>, 1<F,*,0>
```

# Reflective/Controlled Approach

- Decision-theoretic problem-space search across metalevels
  - Very Soar-like, but with softmax combination
    - Depends on *summary product* and Sigma's *mixed* aspect
  - Corresponds to PsychSim's online reasoning



# Comments on the Ultimatum Game

- Reactive version (5 conditionals)
  - A's normalized distribution over offers: <.315, .399, .229, .057>
  - 1 decision (94 messages) and **.02 s** (on a MacBook Air)
- Reflective version (19 conditionals)
  - A's normalized distribution over offers: <.314, .400, .229, .057>
  - 72 decisions (868 messages/decision) and **126.69 s**

Distributions Comparable

Speed Ratio >6000

- Same result, with distinct computational properties
  - Reactive is fast and occurs in parallel with other memory processing, but is not (easily) penetrable by new bits of other knowledge
  - Reflective is slow, sequential, but can (easily) integrate new knowledge
  - Distinction also maps onto *expert vs. novice* behavior in general

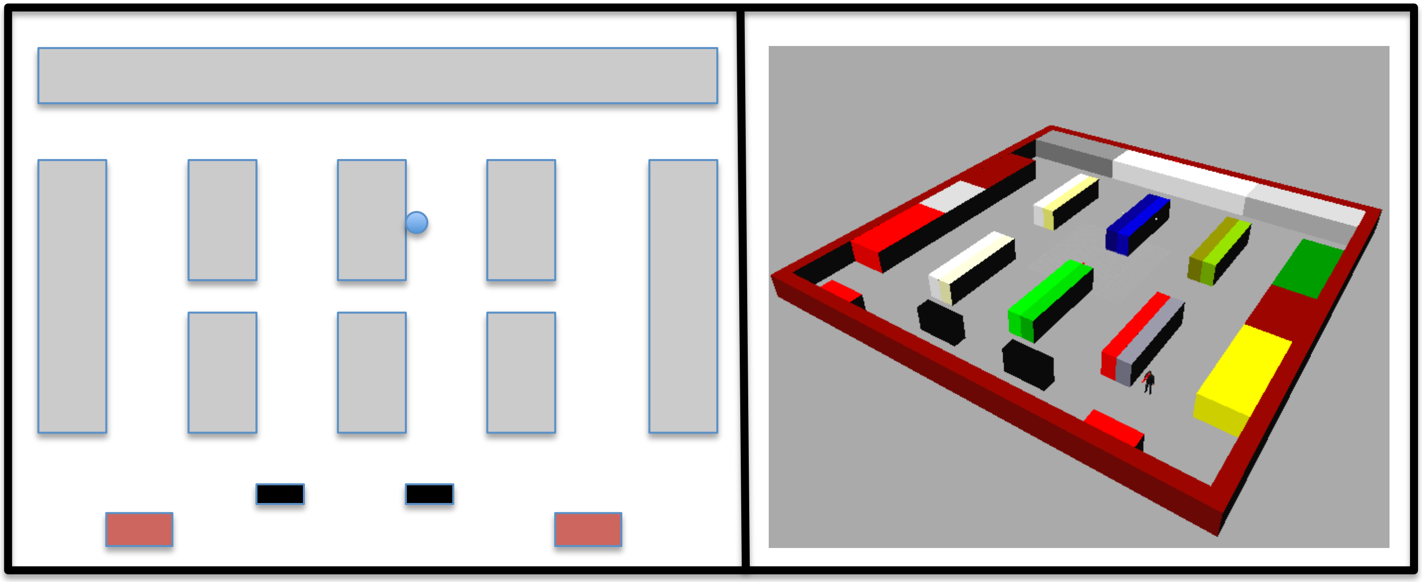
*Raises possibility of a generalization of Soar's chunking mechanism*

  - Compile/learn reactive trellises from reflective problem solving
  - Finer grained, mixed(/hybrid) learning mechanism

# INTERACTIVE, ADAPTIVE VIRTUAL HUMANS

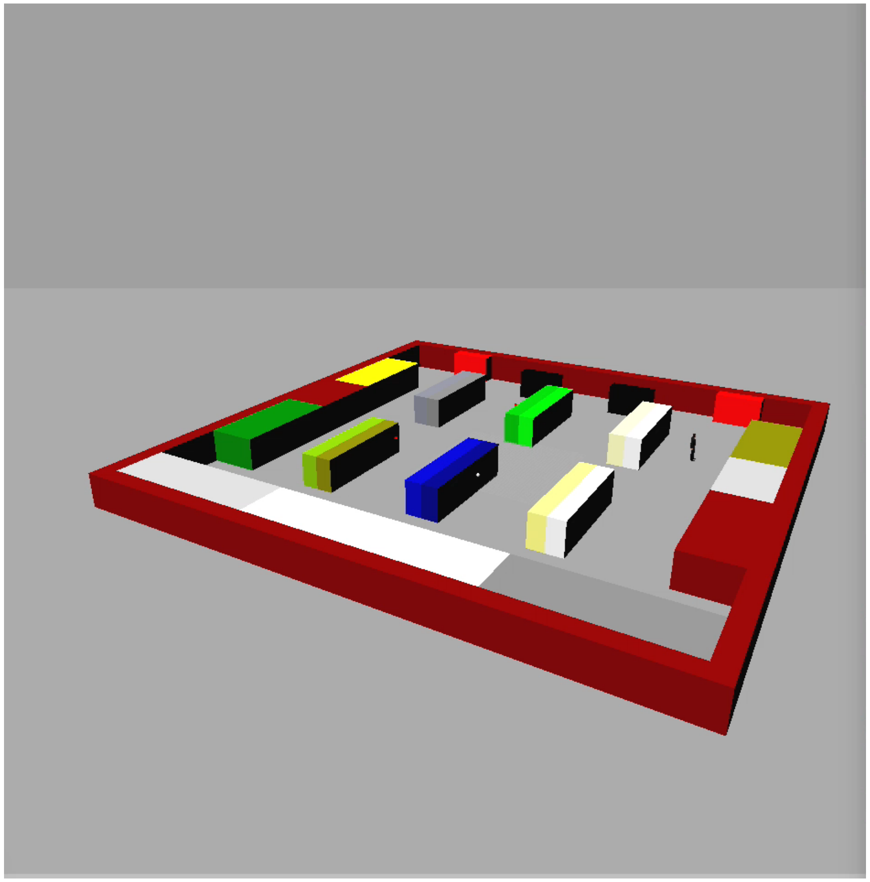
# Interactive, Adaptive Virtual Humans

- Control behavior of SmartBody VH(s) in a retail store scenario
  - A civilian instance of a *physical security system*

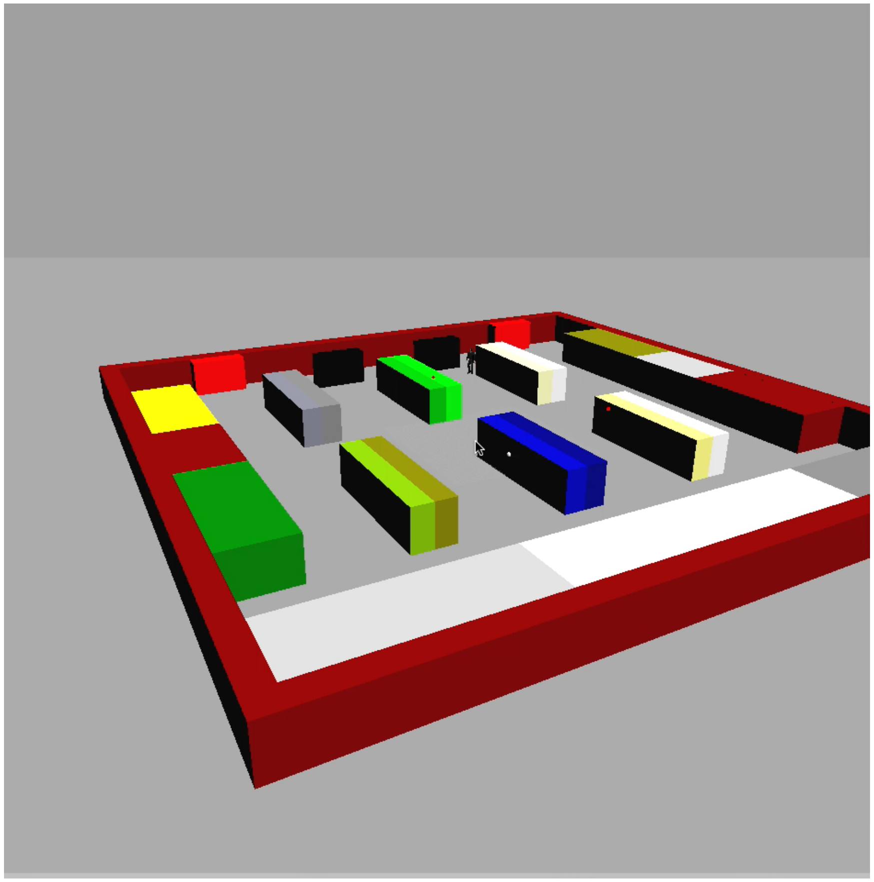


  - Rule-based, probabilistic and social reasoning (ToM)
  - Simultaneous localization and mapping (SLAM)
  - Multiagent reinforcement learning (RL)
  - [Appraisal+attention-based control]

# Simultaneous Localization and Mapping (SLAM)

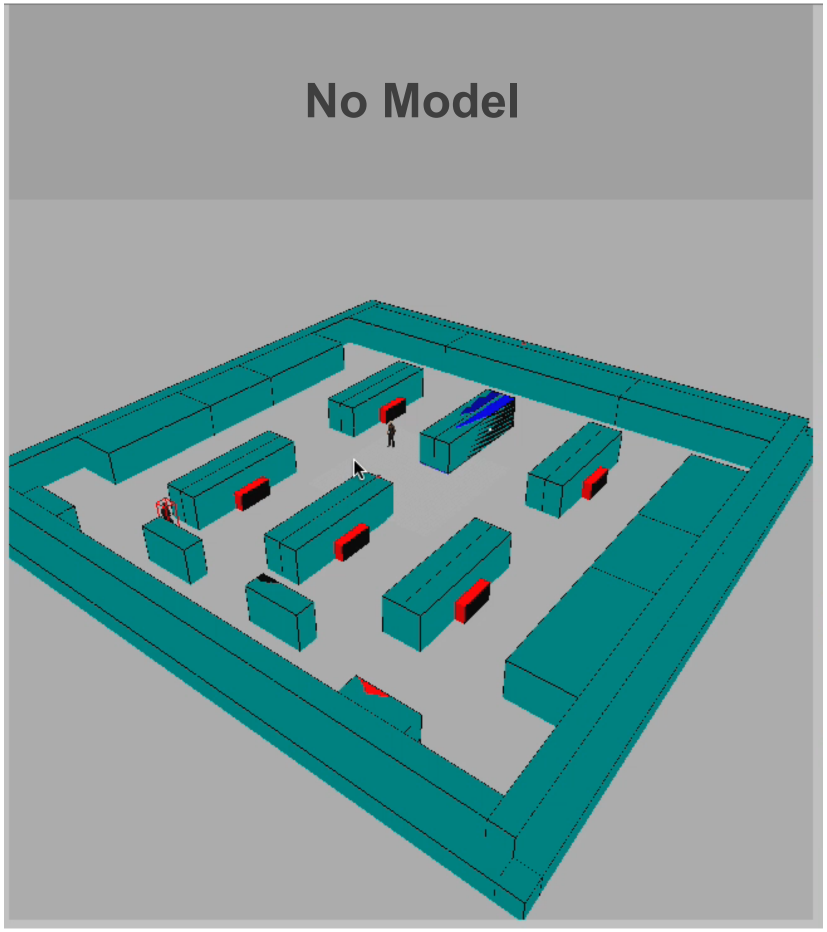


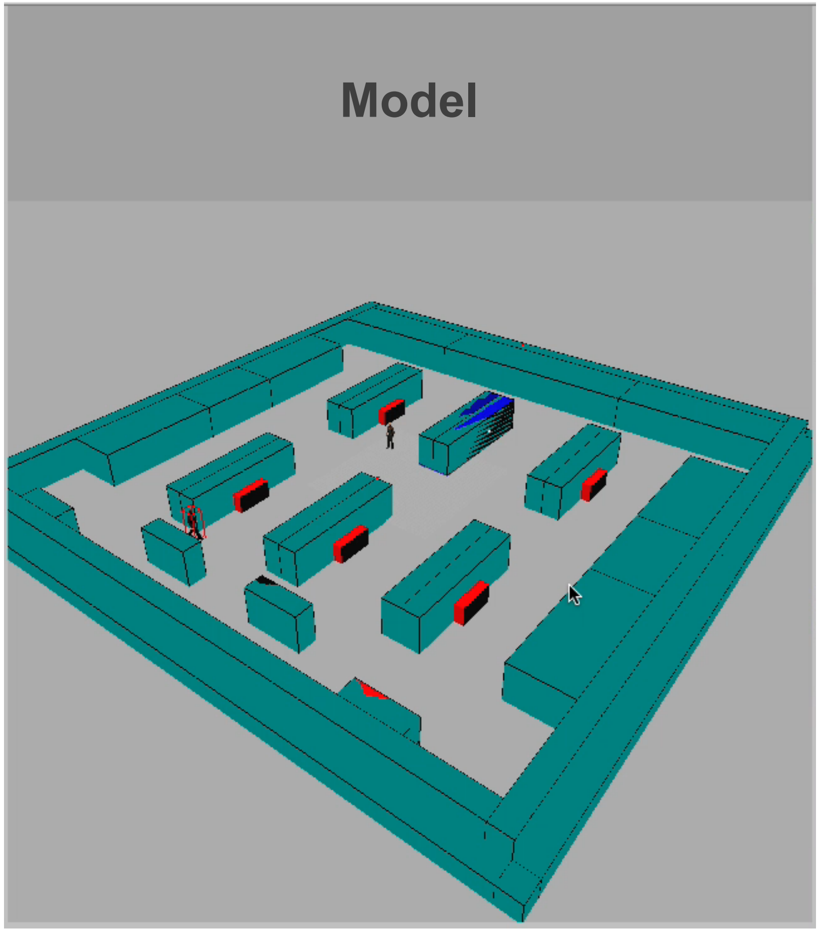
No Map



Map

# Multiagent Reinforcement Learning (RL)



# SUMMARY

## Current and Near Future Topics

- Scaling up memory, reasoning and learning
- Continuous speech understanding, and its integration with language and cognition
- Theory of Mind
- Emotion/affect and its relationship to the architecture
- Distributed vectors/semantics (i.e., word embeddings)
- (Deep) neural networks
- A generalized skill acquisition mechanism (chunking)
- A new level below the graphical architectural
- Exploiting parallelism and GPUs for efficiency
- Interactive, adaptive, intelligent, emotional virtual humans

## Wrapping Up

- Replicating capabilities from existing architectures in a functionally elegant manner
- Extending uniformly to capabilities only possible with existing architectures via interfaces to external modules
- Continuously working on efficiency

- Sigma website is http://cogarch.ict.usc.edu
  - Most papers on Sigma can be found there
  - New papers on which I'm an author usually appear online sooner at http://cs.usc.edu/~rosenblo/pubs.html
- Public release, with online tutorial, available in May