Hierarchical organization of intelligence

Ethology and AI perspectives

Joanna J. Bryson

Department of Computer Science, University of Bath

Konrad Lorenz Institute
for Evolution & Cognition Research

KLI

Outline

- History: Ethology and AI
- Parallel-rooted, Ordered, Slip-stack Hierarchical (POSH) Action Selection
  - Controlling artificial life
  - Explaining NI task learning
- Hierarchies in brains and other evolved intelligence

Ethology: Controversy

- McDougall, "Hierarchy of Instincts" 1923
- Lorenz, "denied the existence of superimposed mechanisms controlling the elements of groups... particular activity was only dependent on the external stimulation and on the threshold for release of that activity."

(Horst Hendriks-Jansen 1996)

Ethology: Evidence

- Lashley (1951) showed action sequencing happens too fast to wait for sense/act cycle (c.f. Davelaar 2007).
- Dawkins (1975) argued for hierarchy based on ethological parsimony (in a tribute to Tinbergen).
  - Proposed "parsing" structure.

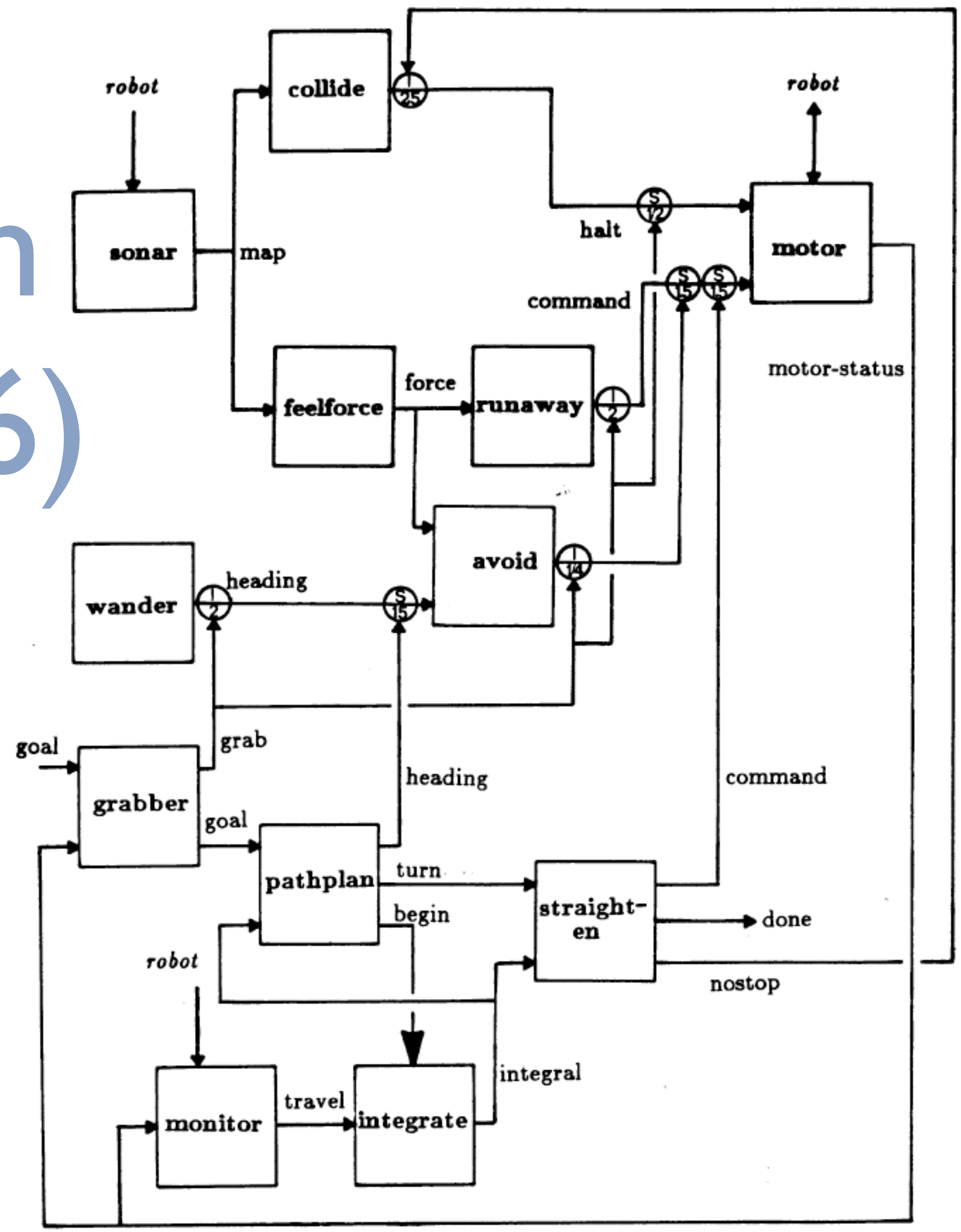
(Bryson 2000)

AI: GOFAI

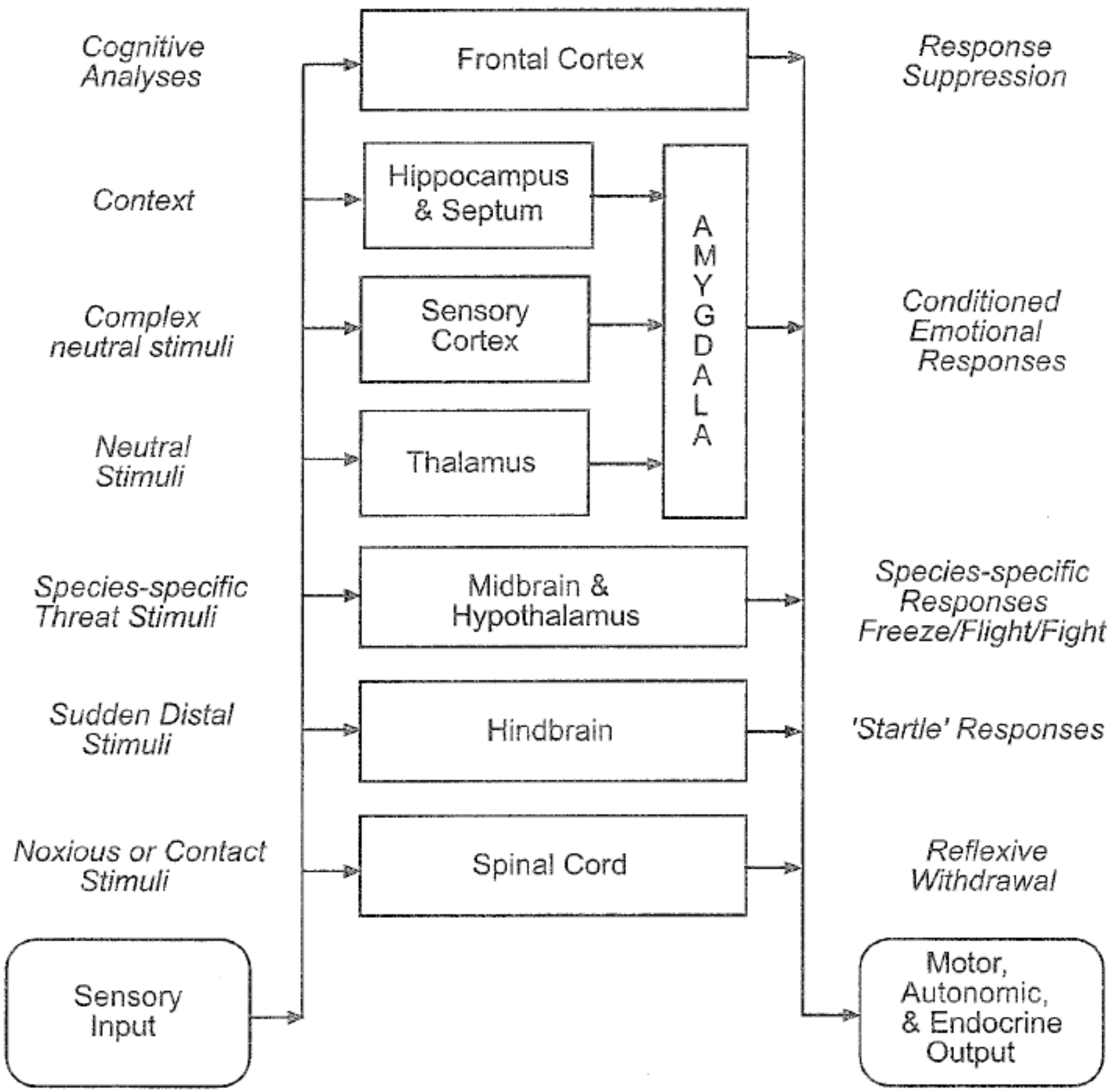
- GPS: You can’t get anywhere without problem spaces (combinatorics.)
- SOAR inherited these; ACT-R tried to do without --- briefly!
- Also: notion of subplans.

Subsumption (Brooks 1986)

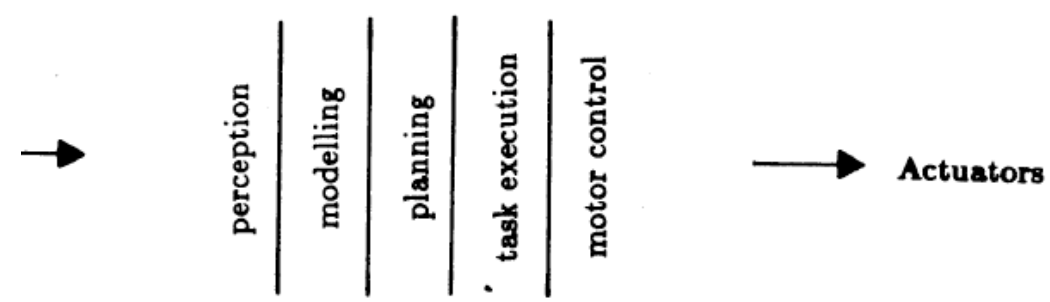
- Intelligence derives only from constantly concurrent behavior modules.
- And inhibition, suppression, timers...



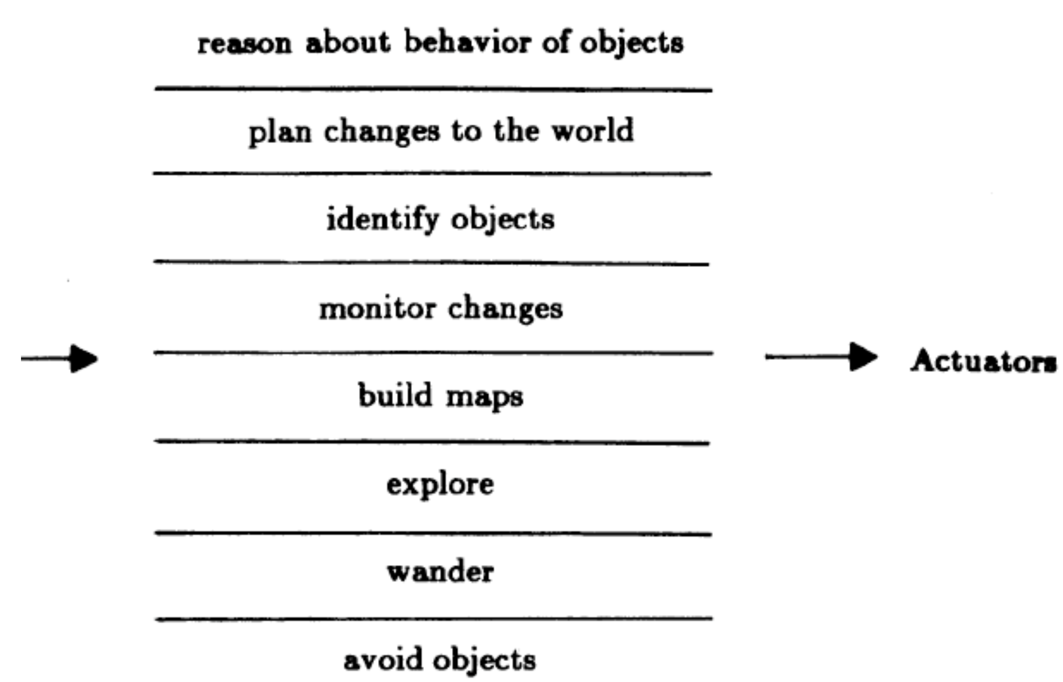
Layers



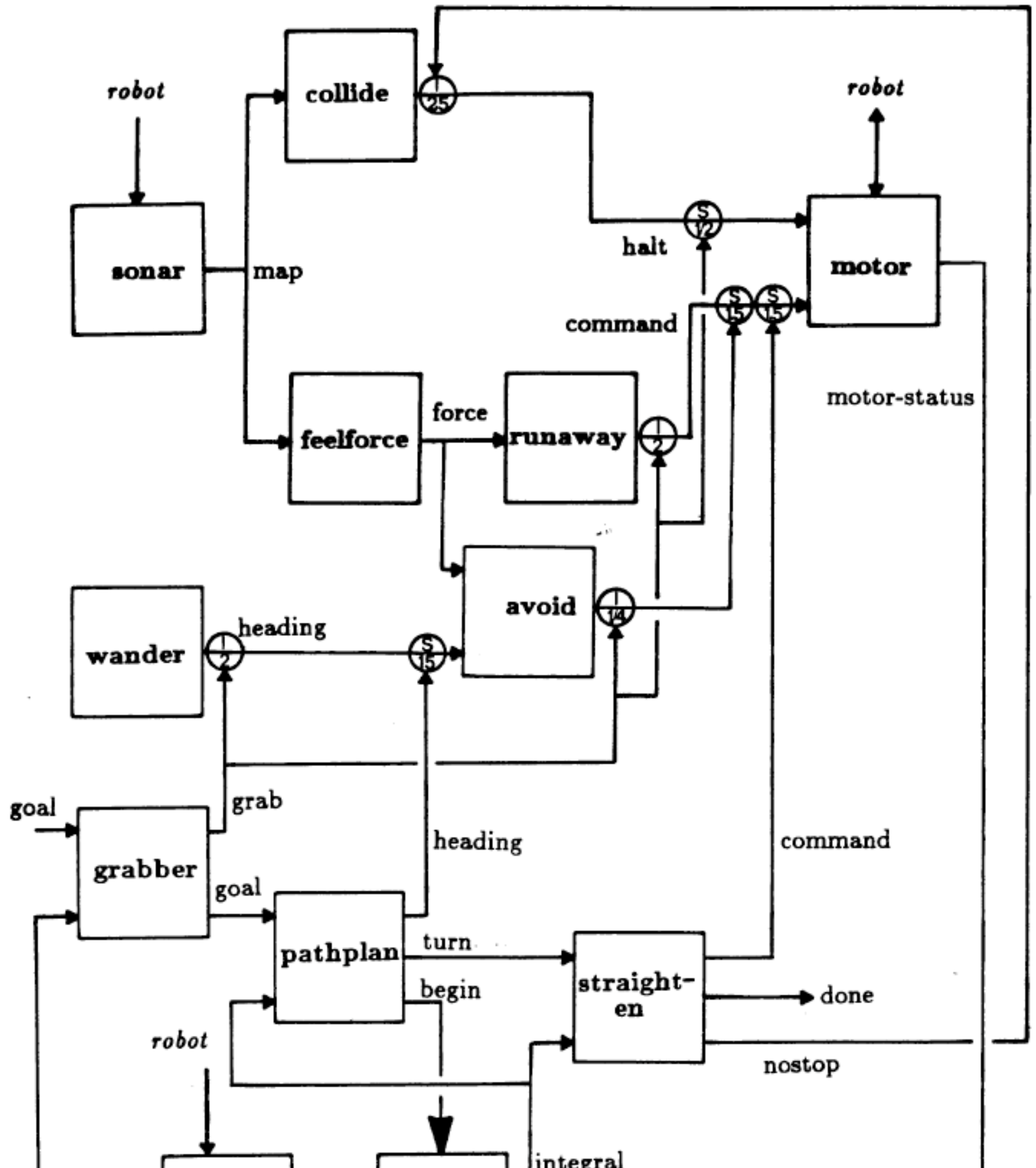
l Control System for a Mobile Robot

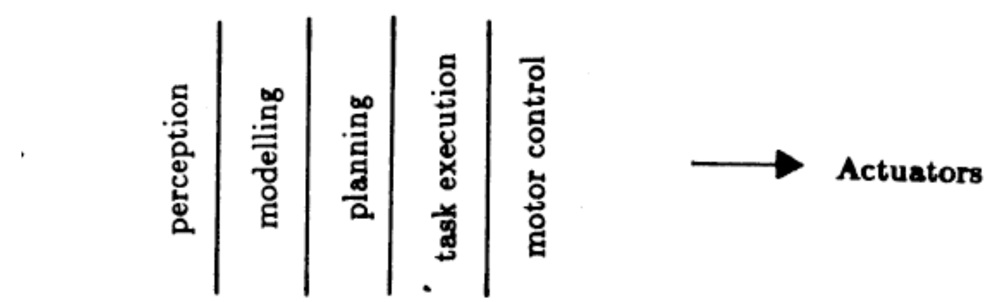


traditional decomposition of a mobile robot control system into functional

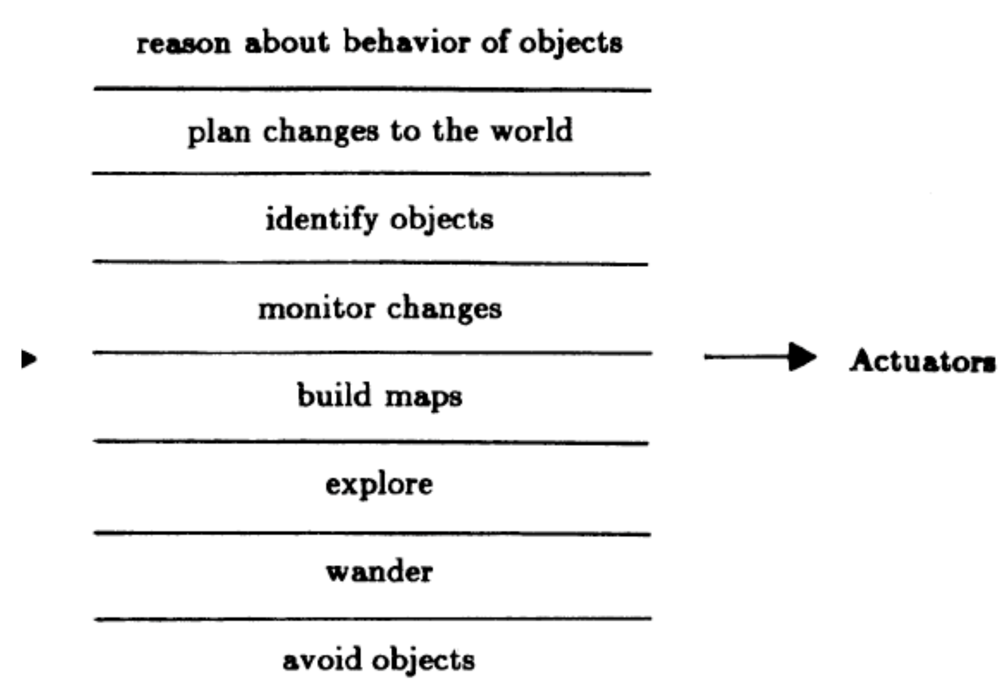


SA: Layers vs. Behaviours



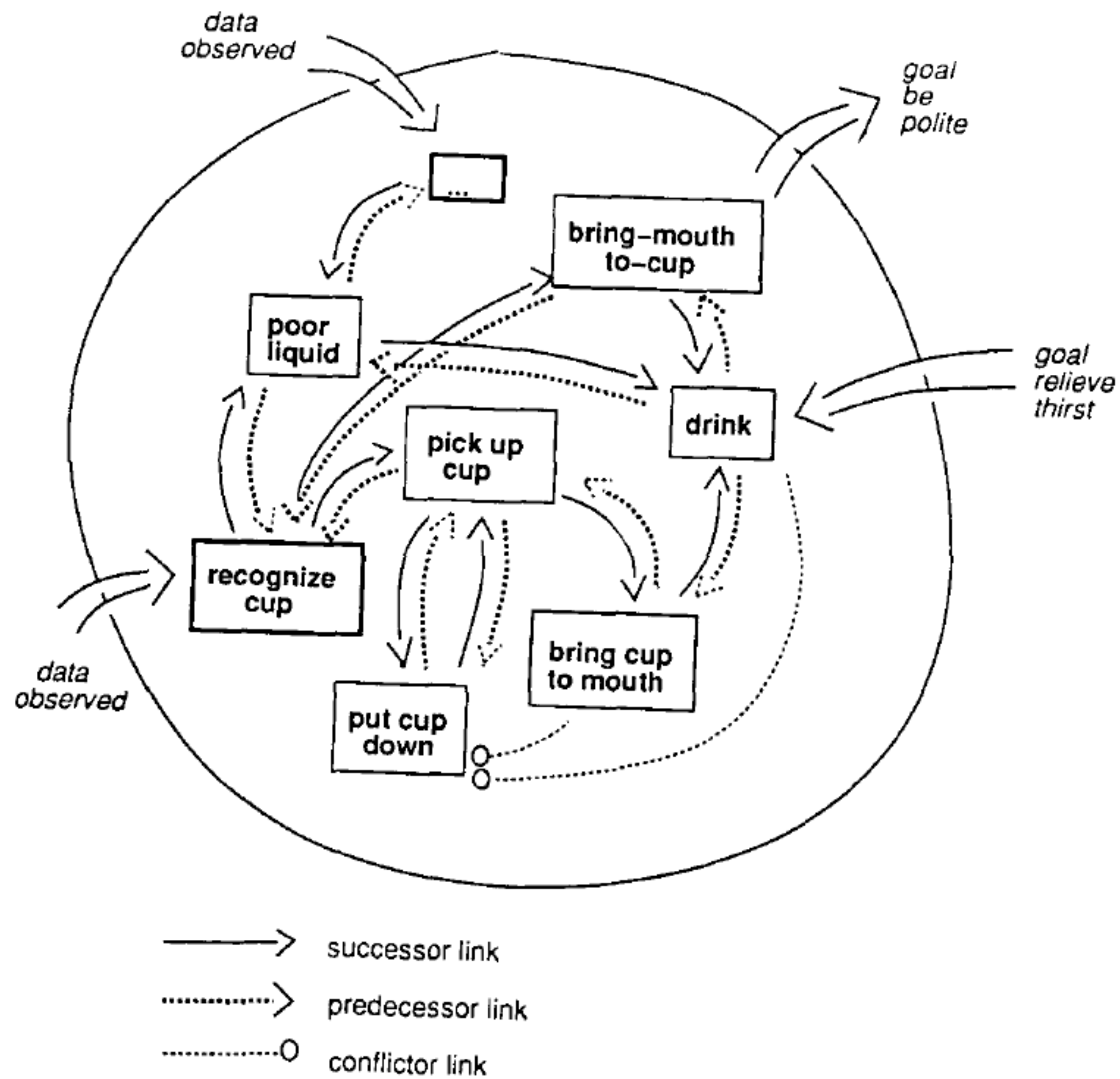


itional decomposition of a mobile robot control system into functional

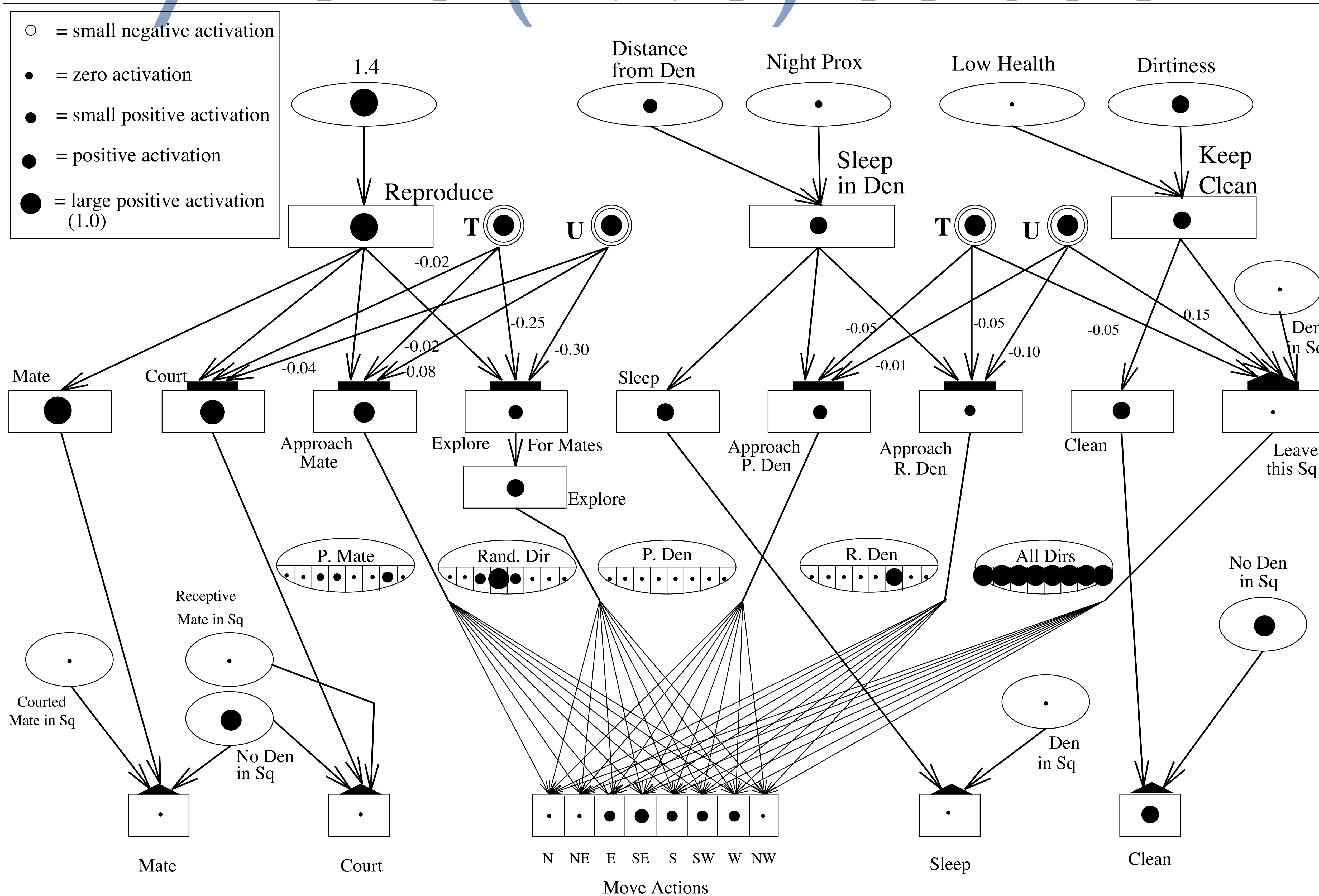


Maes' Spreading Activation Networks

- "Maes Nets" (Adaptive Neural Arch.; Maes 1989)
- "Hierarchies lead to bottlenecks"
- ANA fail to scale (Tyrrell 1994).

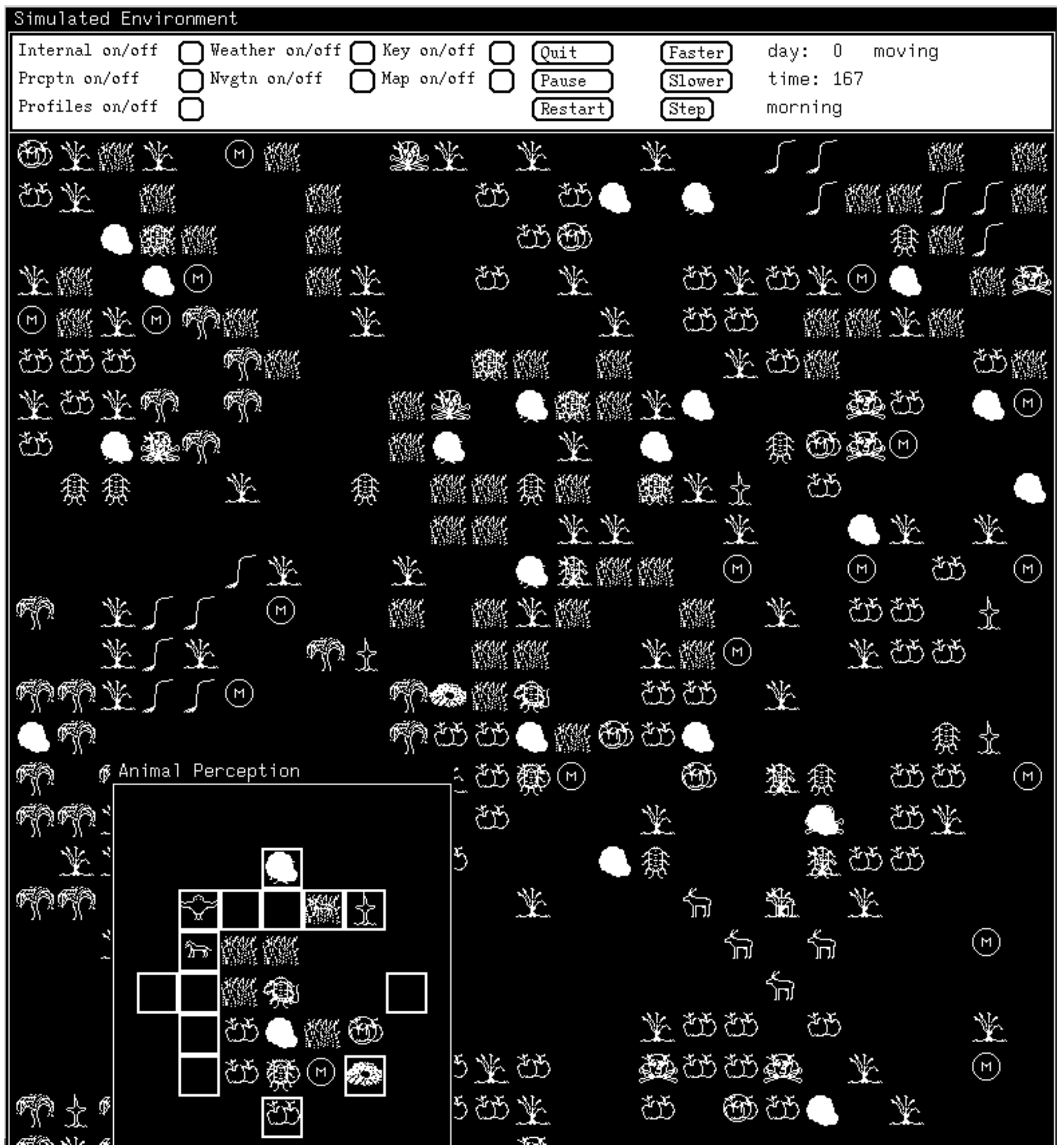


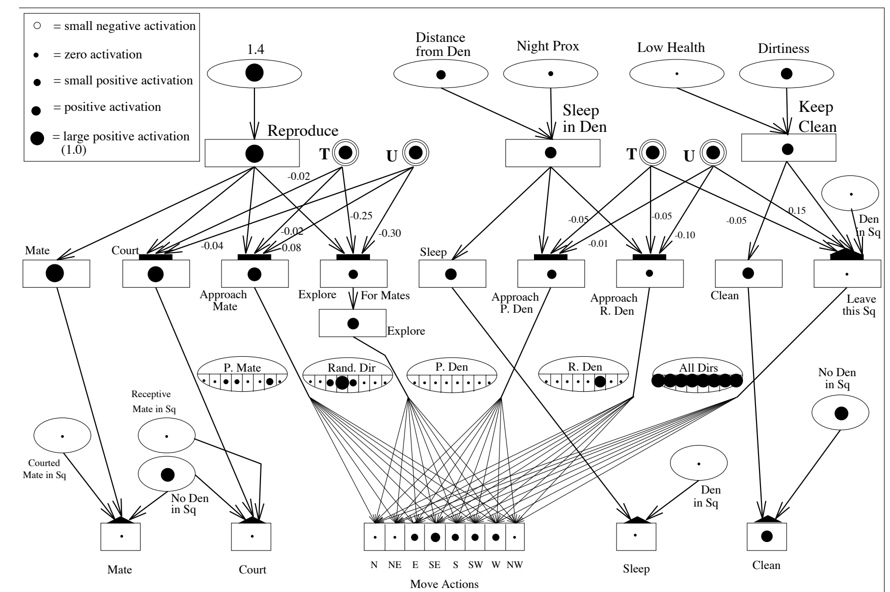
Tyrrell's (1993) Solution



Extended Rosenblatt and Payton Free-Flow Hierarchy

Tyrell's (1993) PhD



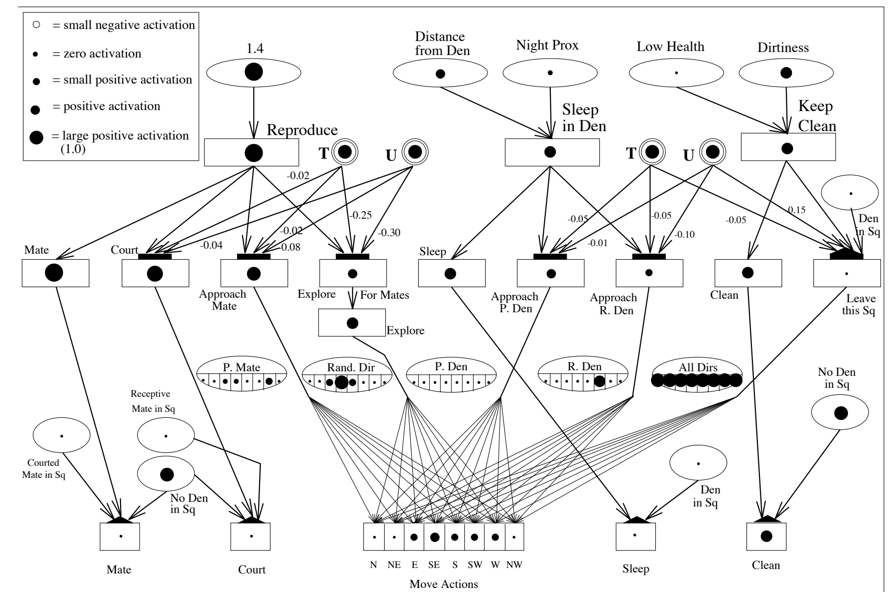


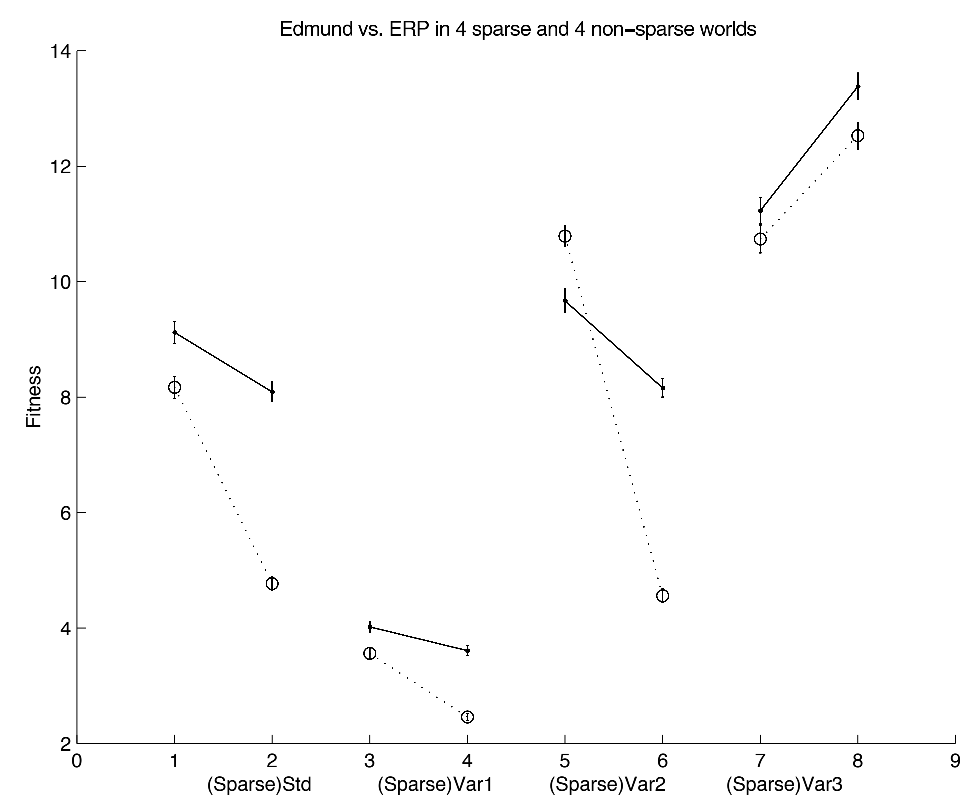
- Compared Hull, Lorenz, Maes & Free-flow Hierarchy.
- (Hull and Lorenz hard to beat!)

Tests performed in Tyrell's "Simulated Environment"

Stricter hierarchy, more probably correct

life (D)	flee (C) (sniff_predator t)	freeze (see_predator t) (covered t) (hawk t)	hold_still
		run_away (see_predator t)	pick_safe_dir go_fast
		look	observe_predator
	mate (C) (sniff_mate t)	inseminate (courted_mate_here t)	copulate
		court (mate_here t)	strut
		pursue	pick_dir_mate go
	triangulate (getting_lost t)	pick_dir_home go	
	home *1::5* (late t) (at_home ⊥)	pick_dir_home go	
	check *1::5*	look_around	
	exploit (C) (day_time t)	use_resource (needed_res_avail t)	exploit_resource
		leave	pick_dir go
	sleep_at_home (at_home t) (day_time ⊥)	sleep	





(Bryson SAB 2000)

Parallel-rooted, Ordered Slip-stack Hierarchical (POSH) Action Selection

- Part of Behavior Oriented Design (Bryson 1997, 2000-2007).
- Use hierarchy to describe priorities for behaviour-arbitration in modular, proactive, real-time intelligent systems with multiple conflicting goals.

Parallel-rooted, Ordered Slip-stack Hierarchical (POSH) Action Selection

- Part of Behavior Oriented Design (Bryson 1997, 2000-2007).
- Use hierarchy to describe priorities for behaviour-arbitration in modular, proactive, real-time intelligent systems with multiple conflicting goals.

What caused cognitivism?

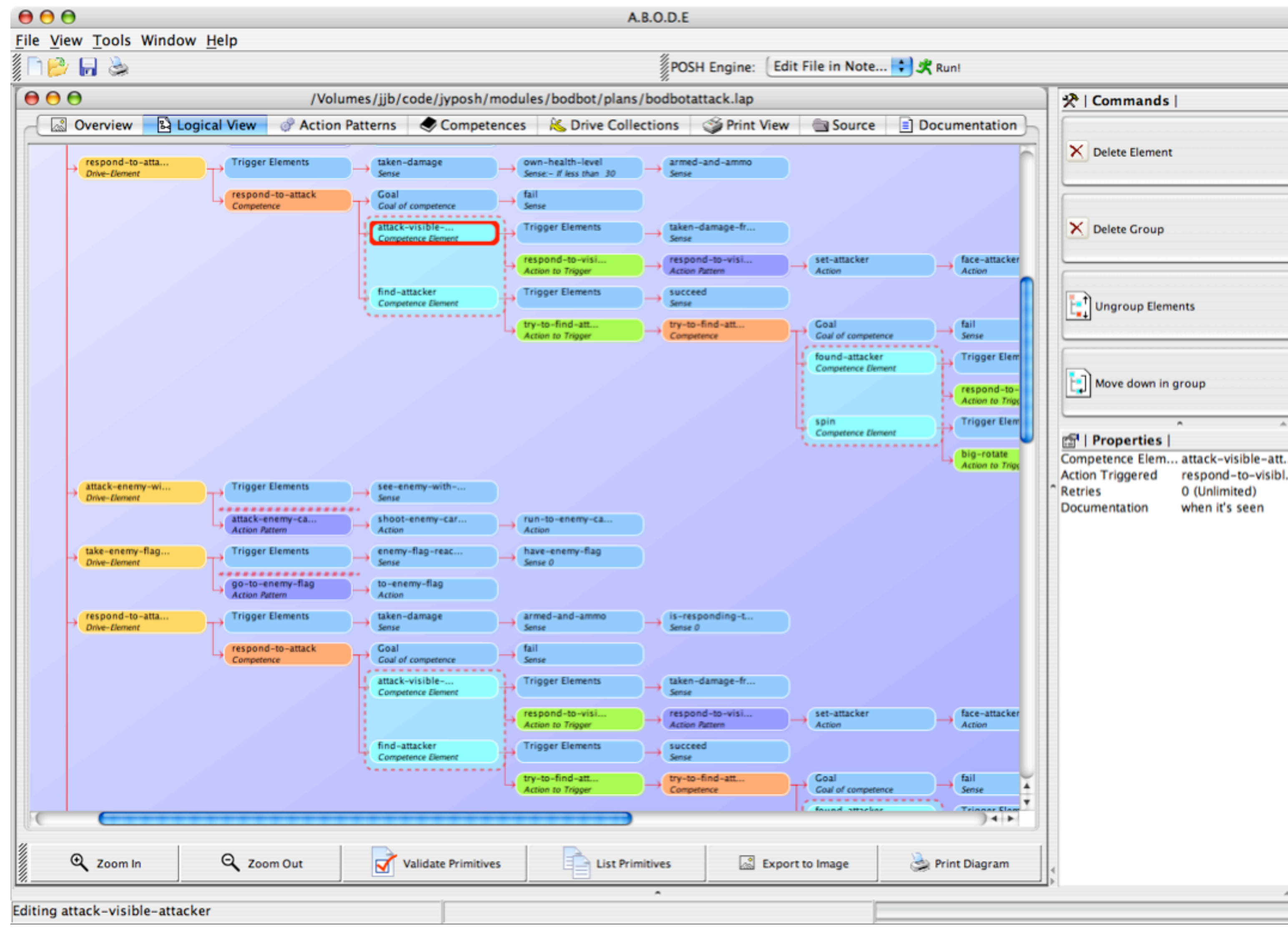
- Latent Learning (Tolman 1930)
- Specialized Learning (Gallistel *et al*. 1991 for review)
  - Pigeons: learn pecking for food, flapping for shock, not vv.
  - Rats: learn smell cues for poison, visual or sound for shock, not vv.

BOD

- Modules for perception, action, memory, learning.
  - Each can “run in own thread” / continuous parallel.
- POSH dynamic plans express sequences, competences (subplans), drives (focus of attention.)
  - Action selection necessary where resource contention (Blumberg 1996).

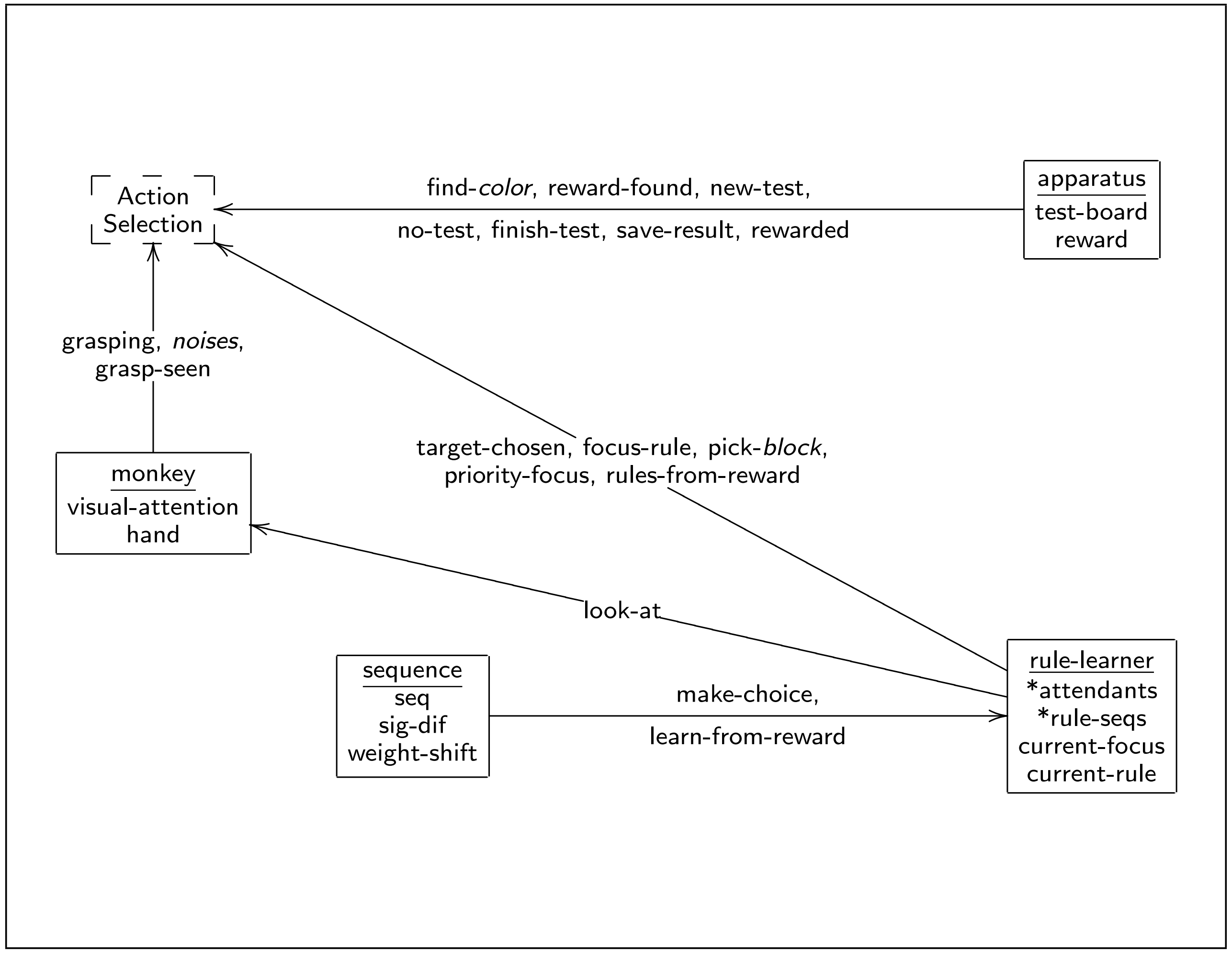
POSH plan in ABODE (for UT: Capture the Flag)

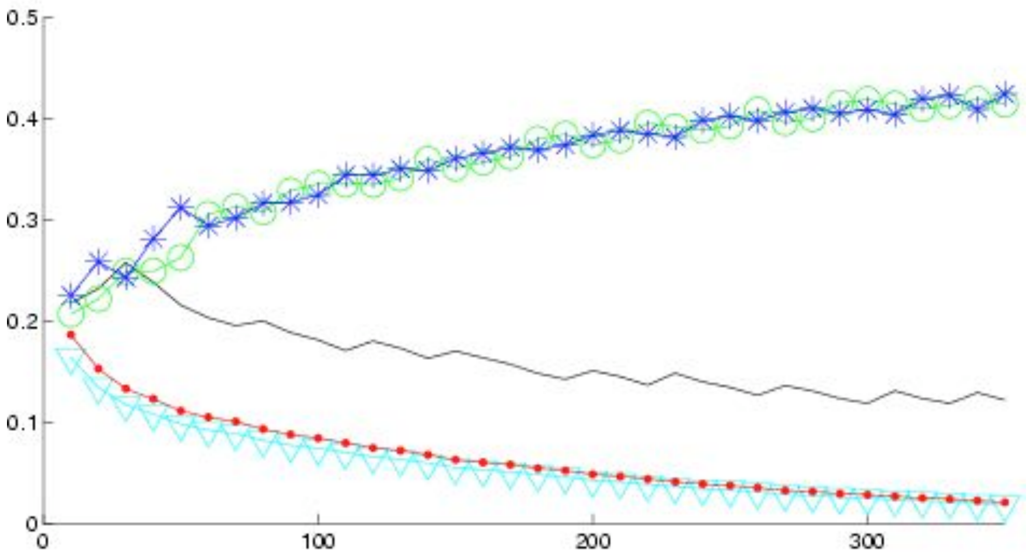
(Partington & Bryson 2005)



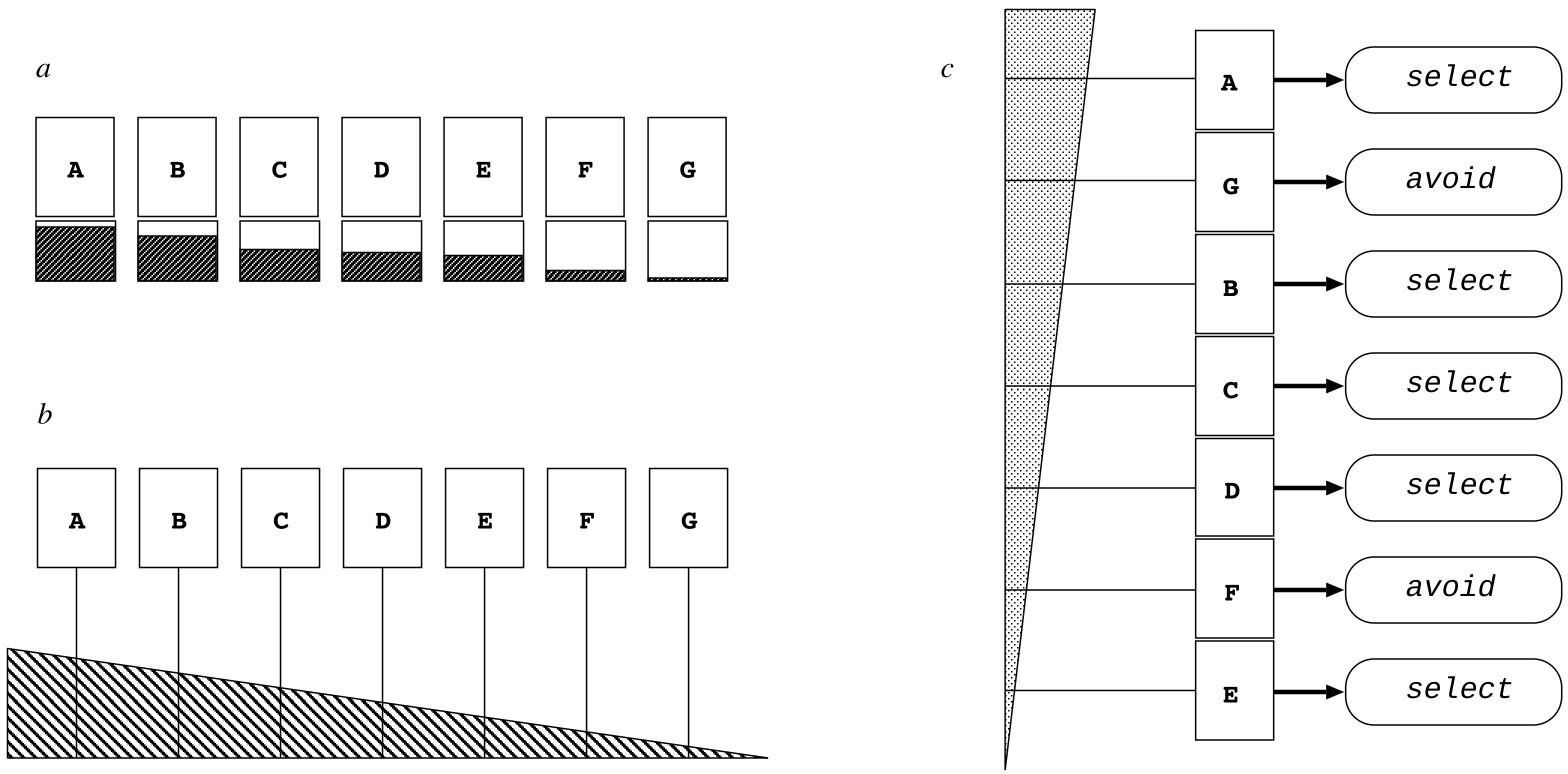
- Advanced BOD Environment.
- Works for realtime or cycle-based systems.
- Games, robots, intelligent environments, alife.

Animals Sacrifice Correctness for Tractability (TI)



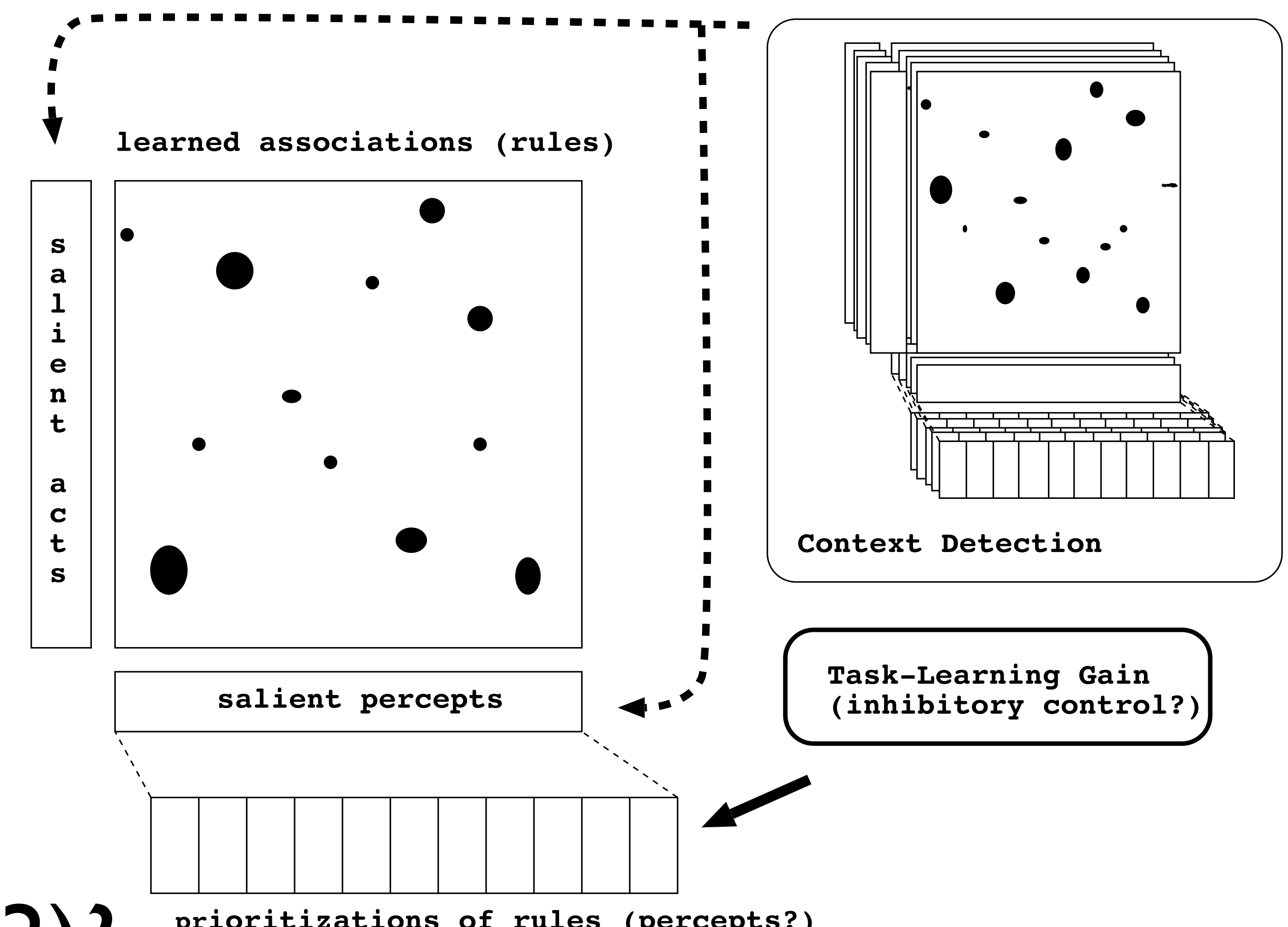


(Bryson & Leong 2007)



a. Value Transfer
b. Sequence
c. Piroritised productions

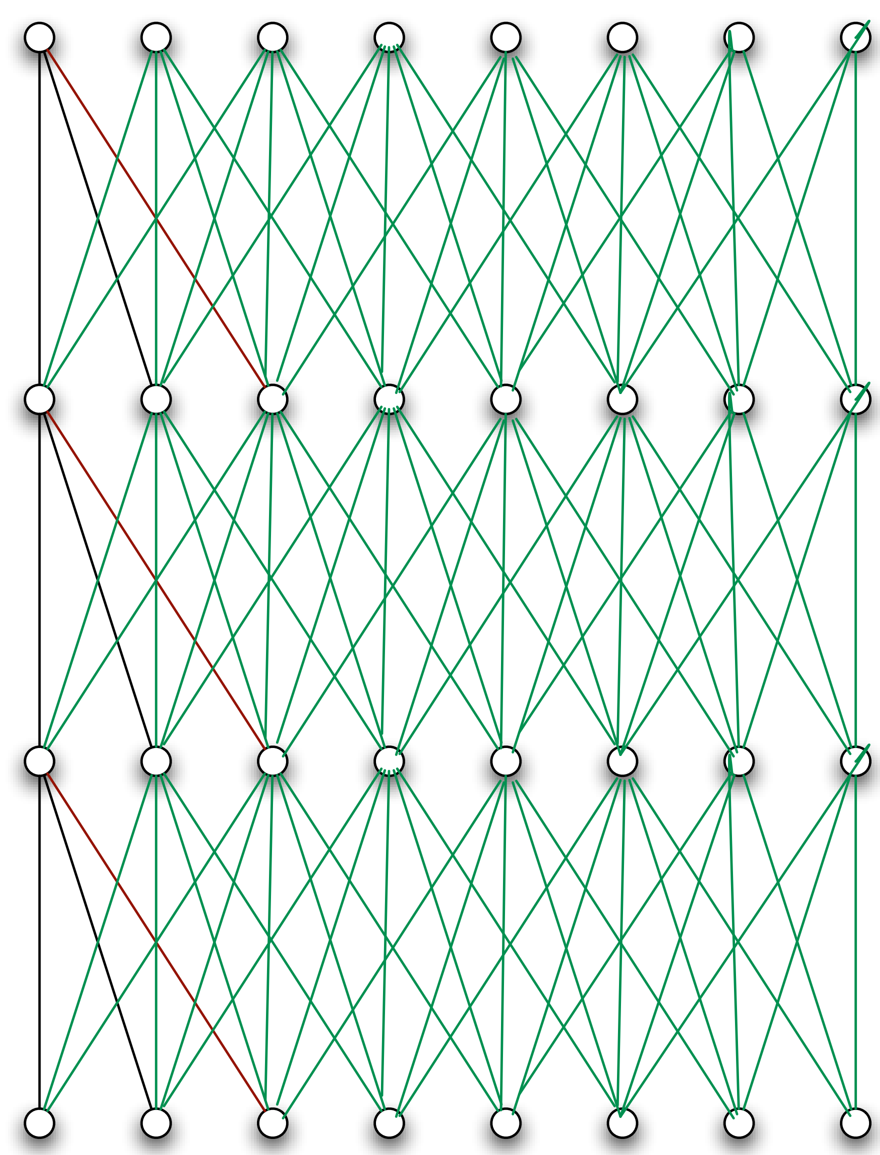
1. Action and perception categories learned cortically.
2. Associations in parahippocampal gyrus (DG)?
3. Priortisations in HP (CA3)?



Outline

- History: Ethology and AI
- Parallel-rooted, Ordered, Slip-stack Hierarchical Action (POSH) Selection
  - Controlling artificial life
  - Explaining NI task learning
- Hierarchies in brains and other evolved intelligence

Much of cortex is a persistent two-way hierarchy



- Each node informs and is informed by a number of nodes in the next level.
- Two inverted columns of overlapping hierarchies.
- Also has lateral excitation and inhibition.
- Culture is like this?

Summary

- Ethology and AI both have “egalitarian” tendencies, but for AS solid evidence that...
  - Ethology: there is more state than just sense/act associations.
  - In AI, that hierarchies can be flexible, efficient, and easier to program.
- Elsewhere, persistent hierarchies are used for recording and developing / evolving information theories.

Hierarchical organization of intelligence

Ethology and AI perspectives

Joanna J. Bryson

Department of Computer Science, University of Bath

Konrad Lorenz Institute
for Evolution & Cognition Research

KLI

POSH Action Selection

- On every cycle, roots / drives are checked to see whether attention should shift.
- For selected drive, execute the next action in sub-plan already attended to (highest-priority production for competence, next item for sequences, primitive...)
- AS must trade off dithering for flexibility.
- Not a strict hierarchy.

Intelligence

- What matters is expressing the right behavior at the right time: action selection.
- Conventional AI planning searches for an action sequence, requires set of primitives.
- Learning searches for the right parameter values, requires primitives *and* parameters.
  - *parameter:* variable state.
  - Evolution and development *are* learning.

Combinatorics

- If . . .

– an agent knows 100 actions (e.g. eat, drink, sleep, step, turn, lift, grasp, poke, flip...), and
– it has a goal (e.g. go to Madagascar)

- Then . . .

– Finding a one-step plan may take 100 acts.
– A two-step plan may take 100^2 (10,000).
– For unknown number of steps, may search forever, missing critical steps or sequence.

Intelligence & Design

- Combinatorics is the problem, search is the only solution.
- The task of intelligence is to focus search.
  - Called bias (learning) or constraint (planning).
  - Most behavior has no or little real-time search.
- For artificial intelligence, most focus comes from design (including physical affordances).

Modularity is not Enough

Get Fuzzy (Conley 2006)

BOD Development Cycle

1. Initial decomposition ⇒ specification.
2. Scale the system.
   i. Code one behavior and/or plan.
   ii. Test and debug code (test earlier plans).
   iii. Simplify the design.
3. Revise the specification.

BOD Development Cycle

1. Initial decomposition ⇒ specification.
2. Scale the system.
   i. Code one behavior and/or plan.
   ii. Test and debug code (test earlier plans).
   iii. Simplify the design.
3. Revise the specification.

1. Specify (high-level) what the agent will do.
2. Describe activities as sequences of actions. competences and action patterns
3. Identify sensory and action primitives from these sequences.
4. Identify the state necessary to enable the primitives, cluster primitives by shared state. behavior modules
5. Identify and prioritize goals / drives. drive collection
6. Select a first (next) behavior to implement.

BOD Development Cycle

1. Initial decomposition ⇒ specification.
2. Scale the system.
   i. Code one behavior and/or plan.
   ii. Test and debug code (test earlier plans).
   iii. Simplify the design.
3. Revise the specification.

BOD Development Cycle

1. Initial decomposition ⇒ specification.
2. Scale the system.
   i. Code one behavior and/or plan.
   ii. Test and debug code (test earlier plans).
   iii. Simplify the design.
3. Revise the specification.

Simplify the Design

Use the simplest representations.

- Plans:
  - primitives, action patterns, competences.
  - drives only if need to always check.
- Behavior modules / memory:
  - none, deictic, specialized, general.

(Bryson, AgeS 2003)

Simplify the Design

Trade off representations: plans vs. behaviors

- Use simplest plan structure unless redundancy (split primitives for sequence, add variable state in modules).
- If competences too complicated, introduce primitives or create more hierarchy.
- Split large behaviors, use plans to unify.
- All variable state in modules (deictic).

(Bryson, AgeS 2003)

life (D)	untangle (tangled?)	untangle	
	groom (C) (want-to-groom?)	(partner-chosen?) (aligned?)	notify groom
		(being-groomed?)	choose-groomer-as-partner
		(partner-chosen?) (touching?)	notify align
		(partner-chosen?)	notify approach
		($\top$)	choose-partner
	receive (being-groomed?)	tolerate-grooming	
	explore (C) (want-novel-loc?)	(place-chosen?) (there-yet?)	lose-target
		(place-chosen?)	explore-that-a-way
		($\top$)	choose-explore-target
	wait ($\top$)	wait	