The Role of Cognition in Cognitive Systems
From Robots to Primatology

Joanna J. Bryson
Artificial Models of Natural Intelligence
University of Bath
Institute of Cognitive and Evolutionary Anthropology
University of Oxford
http://www.cs.bath.ac.uk/ai/AmonI.html
Outline

• Introduction to Intelligence & Cognition
• Where do you put it in a Cognitive System? *Behavior Oriented Design*
• Primates & Cognition
Intelligence

• What matters is expressing the right behavior at the right time: action selection.

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.

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.

• 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

Intelligence & Design

• Combinatorics is the problem, search is the only solution.

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.

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 **natural** intelligence, most **focus** evolves.

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 natural intelligence, most focus evolves.
- For artificial intelligence, most focus designed.

Cognition
Cognition

Definition:

Cognition is on-line (real-time) search.
Cognition

**Definition:**

Cognition is on-line (real-time) search.

**Consequence:**

Cognition is bad.
Cognition
Cognition

- Why is cognition / individual search bad?
  - Slow
  - Uncertain
Cognition

- Why is cognition / individual search bad?
  - Slow
  - Uncertain
  - Unpopular in most species.
  - Plants
  - Protozoa
Only think when you don’t know what’s going on

- Cognition is costly.
- Time, errors, metabolism.
- Value of investment can be estimated from own experience or mother’s (maternal effects).

Deary et al. (2004) (Schaie et al. 2004; Kotrschal & Taborsky in prep.)

Outline

• Introduction to Intelligence & Cognition
• Where do you put it in a Cognitive System? **Behavior Oriented Design**
• Primates & Cognition
Architecture

• Where do you put the cognition?

• Really: How do you bias / constrain / focus cognition (learning, search) so it works?
Behavior Oriented Design

- All search (learning, planning) is done within modules with specialised representations.
- Specialized representations promote reliability of search; also determine decomposition.
- Modules provide perception, action, memory. Arbitration via hierarchical dynamic plans.
- Iterative / agile test & development cycle.

(Bryson 2001, 2003)
BOD Action Selection

Parallel-rooted, Ordered, Slip-stack Hierarchical (POSH) action selection:

• Some things need to be checked at all times: drive collection.

• Some things only need considering in particular context: competences.

• Some things reliably follow from others: action patterns.
BOD Robot Example

(ATAL 1997, PhD 2001)

- Behaviour Library — per platform.
- POSH plan — per “species” / goal set.
- Memory — per individual.

DP-Map

*landmarks

untried_near_neighbor?, untried_far_neighbor?
pick_near_neighbor, pick_further_neighbor

Action Selection

in_dp, entered_dp

continue_untried
keep_going

done-that

direction, time

csense, odometry

Robot (and C-Sense)

E-Memory
*direc_tions
*times

DP-Land
x,y
in-dir
out-dir
life (D)

- talk [1/120 Hz]
  (worth_talking \top)

- sense (C) [7 Hz]
  - bump (bumped \top)
  - look

- walk (C)
  - halt (has_direction \top)
    (move_view 'blocked)
  - start (has_direction \perp)
  - continue

- wait

- speak

- yelp reg.bump back.off clear.bump lose.direction

- compound.sense

- lose.direction

- pick.open.dir

- move.narrow (move_view 'clear) correct_dir

- snore sleep
halt  (has_direction ⊤)  lose_direction
(move_view 'blocked)

cogitate_route (C)

ten_ (in_dp ⊥)  lose_direction greet_dp
(entered_dp ⊥)

leave_dp (in_dp ⊤)  dismiss_dp
(entered_dp ⊤)

look_up
(untried_near_neighbor ⊤)

pick_near_neighbor

pick_previous_direction

pick_far_neighbor

start (has_direction ⊥)  ask_directions

continue

move narrow (move_view 'clear) correct_dir
Statistical Testing of BOD Action Selection

Tests performed in Tyrell’s (1993) “Simulated Environment”

Combinatorics vs Culture

• If each agent has a 1% chance of discovering a skill (e.g. making cheese) in its lifetime and there are 4000 agents, probably some agents will know the skill.

• If it is easier to learn the skill from a knowledgeable agent than by discovery, then selective pressure for culture.

• Inclusive fitness  \( c < b \times r \)
  (Hamilton 1964; West et al 2007).
BOD Experiments in Social Learning in VR

Basic Result: Still intractable without an enormous amount of prior information.

Extension of Roy 1999 (PhD) to realtime planning.

Fortunately, Priors Easy to Insert with BOD


IDEs for Dynamic Plans

- **Advanced BOD Environment.**

- **Ubiquitous robotics requires AI (or servicing AI) by graduates with second class honours.**

IDEs for Dynamic Plans

This work was & is again being funded by aerospace.

- **Advanced BOD Environment.**
- **Ubiquitous robotics requires AI (or servicing AI) by graduates with second class honours.**

Outline

• Introduction to Intelligence & Cognition
• Where do you put it in a Cognitive System?
  Behavior Oriented Design
• Primates & Cognition
Why BOD Works

• **Modularity**: problem spaces, combat combinatorics, allow locally-optimal representations.

• **Should use ordinary (OO) code** (arbitrarily powerful but also access to primitives.)

• Hierarchical **action selection** for arbitration.

• Dedicated, high-frequency **goal / attention switching**, keeps hierarchical AS responsive.

• **Agile development, refactoring** (Beck 2000).
Subsumption (Brooks 1986)
Subsumption (Brooks 1986)

- Emphasis on sensing to action (via Augmented FSM).
Subsumption (Brooks 1986)

- Emphasis on sensing to action (via Augmented FSM).
- Very complicated, distributed arbitration.
Subsumption (Brooks 1986)

- Emphasis on sensing to action (via Augmented FSM).
- Very complicated, distributed arbitration.
- No learning.
it is nor hand, nor foot, nor arm, nor face, nor any other part belonging to a man.
Dennett (2008)

“Contents arise, get revised, contribute to... the modulation of behavior, and in the process leave their traces in memory...”

“Only [commonality is] the historical property of having won a temporally local competition with sufficient decisiveness... to enable recollection...”
“Contents arise, get revised, contribute to... the modulation of behavior, and in the process leave their traces in memory...”

“Only [commonality is] the historical property of having won a temporally local competition with sufficient decisiveness... to enable recollection...”

• Characteristics:
  1. Selection from concurrent options.
  2. Indicated by episodic memory.
Function-Based Theory

- **Consciousness** is holding one stimulus in mind while searching options primed by it for a better response.

- Only triggered when next action isn’t obvious (reflexive or trained).

- Side effect: special types of learning.

- Side effect: **long reaction times**. Focus attention longer when less certain.

Monkeys Learning New Rewards (or not)

- Monkeys that learn chained pairs of values (A>B; B>C; C>D; D>E; E>F) normally are faster at assessing stimuli the further they are on the chain (B>E faster than B>D).

- Elderly monkeys are always fast.

- Elderly monkeys also don’t learn when you change the reward scheme -- not aware!

What’s Consciousness?

• A module for learning new action selection.
• Triggered by uncertainty.
• Detectable due to episodic memories & reaction time.
• A finite resource constantly directed towards what seems most surprising.
“If the best the roboticists can hope for is the creation of some crude, cheesy, second-rate artificial consciousness, they still win.”

Outline

• Introduction to Intelligence & Cognition
• Where do you put it in a Cognitive System? Behavior Oriented Design
• Primates & Cognition
• Conclusions
Cognitive Robots

Bad News:
Bad News:

Cognition doesn’t require robots (only rich, dynamic, real-time environments).
Cognitive Robots

Bad News:

Cognition doesn’t require robots (only rich, dynamic, real-time environments).

Robots don’t require cognition.
Cognitive Robots

Bad News:

Cognition doesn’t require robots (only rich, dynamic, real-time environments).

Robots don’t require cognition.

Good News:
Cognitive Robots

Bad News:
Cognition doesn’t require robots (only rich, dynamic, real-time environments).
Robots don’t require cognition.

Good News:
Cognitive robots are still pretty interesting.
What I Learned from Robots
What I Learned from Robots

1. Perception is hard -- which explains the brain.
   
   • Lead to specialized representations encapsulated in modules; my method of behavior-module decomposition.
What I Learned from Robots

1. Perception is hard -- which explains the brain.
   • Lead to specialized representations encapsulated in modules; my method of behavior-module decomposition.

2. Discrete action selection is compatible with continuous acting, provided the primitive `acts’ alter ongoing behaviour supported by modules.
   • e.g. motor act sends target velocity, not vector;
   • multiple || devices/modules e.g. speech, motion.
The Brain

Higher mammals separate sense & action (Central Sulcus).

Chance for Cognition?
(images: Carlson)
When Your Robot Must Think...

- **Modularity**: problem spaces, combat combinatorics, allow locally-optimal representations.

- Hierarchical *action selection* for real-time arbitration between modules.

- Dedicated, high-frequency *goal / attention switching*, compensates for hierarchical AS.

Thanks!

Mark Wood

Cyril Brom (et al)

Tristan Caulfield

Jan Drugowitsch

Jon Leong

Sam Partington
The Role of Cognition in Cognitive Systems
From Robots to Primatology

Joanna J. Bryson
Artificial Models of Natural Intelligence
University of Bath
Institute of Cognitive and Evolutionary Anthropology
University of Oxford
http://www.cs.bath.ac.uk/ai/AmonI.html
**Action Selection**

**Functionalist Assumption:** All we care about is producing intelligent behaviour.


- Thinking, consciousness as epiphenomena (Churchland 1988, Brooks & Stein 1993).

**Science:** We’ll build it to see if we need it.
• Old theories of limits: altruism, rate of environmental change.

• Concurrency can accelerate behaviour change (Bryson 2008).

• Čače & Bryson (2005, 2007) show altruistic communication about food is selected for due to niche creation.

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 $\Rightarrow$ 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)
References
References

References


References


“History as Evolution”
Hypothesis

Joanna J. Bryson, “Cross-Paradigm Analysis of Autonomous Agent Architecture”,
“History as Evolution” Hypothesis

• If an architecture is around for a while, and it changes, the change was probably selected, adaptive.

“History as Evolution”
Hypothesis

• If an architecture is around for a while, and it changes, the change was probably selected, adaptive.

• This is particularly likely if the change goes against the stated theories of the architecture’s makers.

“History as Evolution”
Hypothesis & Correlary
“History as Evolution”
Hypothesis & Correlary

• If an architecture is around for a while, and it changes, the change was probably selected, adaptive.
“History as Evolution”

Hypothesis & Correlary

• If an architecture is around for a while, and it changes, the change was probably selected, adaptive.

• If similar features occur in a lot of architectures with different phylogenies, those features are probably adaptive.
“History as Evolution”
Hypothesis & Correlary

• If an architecture is around for a while, and it changes, the change was probably selected, adaptive.

• If similar features occur in a lot of architectures with different phylogenies, those features are probably adaptive.

• If you want to make a contribution to a field, describe your best innovations in terms of well-known systems.
Productions

• From sensing to action (c.f. Skinner; conditioning; Witkowski 2007.)

• **These work** -- basic component of intelligence.

• The problem is choice (**search**).

• Requires an **arbitration mechanism**.
Production-Based Architectures

- **Expert Systems**: allow choice of policies, e.g. recency, utility, random.
- **SOAR**: problem spaces (from GPS), impasses, chunk learning.
- **ACT-R**: (Bayesian) utility, problem spaces (reluctantly, from SOAR/GPS.)
- Productions operate on predicate database.
- If conflict, declare impasse, reason (search).
- Remember resolution: chunk
### Soar

<table>
<thead>
<tr>
<th>Contributing Ideas</th>
<th>Soar Version</th>
<th>Major Results</th>
<th>Example Systems</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal Dependency</td>
<td>Soar8 - 1999</td>
<td>Substate Coherence</td>
<td>MOUTBOT QuakeBot</td>
<td>SGIO</td>
</tr>
<tr>
<td></td>
<td>Soar7 - 1996</td>
<td>Improved Interfaces</td>
<td>TacAir-Soar RWA-Soar</td>
<td>TCL/Tk Wrapper</td>
</tr>
<tr>
<td></td>
<td>Soar6 - 1992</td>
<td>High Efficiency</td>
<td>Air-Soar Instructo-Soar</td>
<td>C</td>
</tr>
<tr>
<td>Single State</td>
<td>Soar5 - 1989</td>
<td>External Tasks</td>
<td>Air-Soar Hero-Soar</td>
<td>External Release</td>
</tr>
<tr>
<td></td>
<td>Soar4 - 1986</td>
<td>UTC</td>
<td>ET-Soar NL-Soar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Soar3 - 1984</td>
<td>General Learning</td>
<td>R1-Soar</td>
<td></td>
</tr>
<tr>
<td>Preferences</td>
<td>Soar2 - 1983</td>
<td>Universal Subgoaling</td>
<td>R1-Soar Dypar-Soar</td>
<td>OPS5 Lisp</td>
</tr>
<tr>
<td>Weak Methods</td>
<td>Soar1 - 1982</td>
<td>Universal Weak Method</td>
<td>Toy Tasks</td>
<td>XAPS 2 Lisp</td>
</tr>
<tr>
<td>Symbol Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heuristic Search</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem Spaces</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- Soar has serious engineering.
• Soar has serious engineering.

• “Evolution of Soar” is my favorite paper (Laird & Rosenbloom 1996)
• Soar has serious engineering.

• “Evolution of Soar” is my favorite paper (Laird & Rosenbloom 1996)

• Admits problems!
• Soar has serious engineering.

• "Evolution of Soar" is my favorite paper (Laird & Rosenbloom 1996)

• Admits problems!

• Not enough applications for human-like AI
Architecture Lessons (from CMU)

- An architecture needs:
  - action from perception, and
  - further structure to combat combinatorics.
- Dealing with time is hard.
ACT-R

- Learns (& executes) productions.
- For arbitration, rely on (Bayesian probabilistic) utility.
- Call it implicit knowledge.
• Replicate lots of Cognitive Science results.

• See if the brain does what you think it needs to.

• Win Rumelhart Prize (John Anderson, 2000).

ACT-R Research Programme

Intentional Module (not identified)

Goal Buffer (DLPFC)

Visual Buffer (Parietal)

Visual Module (Occipital/Parietal)

Declorative Module (Temporal / Hippocampus)

Retrieval Buffer (VLPFC)

Manual Motor (Motor)

Manual Module (Motor/Cerebellum)

External World

Productions (Basal Ganglia)

Matching (Striatum)

Selection (Pallidum)

Execution (Thalamus)
Architecture Lessons (from CMU)

- Architectures need *productions* and *problem spaces*.
- Real-time is hard.
- Being *easy to use* can be a win.
Spreading Activation Networks

- “Maes Nets” (Adaptive Neural Arch.; Maes 1989)
- Activation spreads from senses and from goals through net of actions.
- Highest activated
Spreading Activation Networks
Spreading Activation Networks

- Sound good:
  - easy
  - brain-like (priming, action potential).
  - Still influential (Franklin 2000, Shanahanhan 2006).
Spreading Activation Networks

• **Sound good:**
  - easy
  - brain-like (*priming*, action potential).
  - Still influential (Franklin 2000, Shanahan 2006).

• **Can’t do full action selection:**
  - Don’t *scale*; don’t *converge* on consummatory acts (Tyrrell 1993).
Tyrrell (1993)

Extended Rosenblatt and Payton Free-Flow Hierarchy
Subsumption (Brooks 1986)
Subsumption (Brooks 1986)

- Emphasis on sensing to action (via Augmented FSM).
Subsumption (Brooks 1986)

- Emphasis on sensing to action (via Augmented FSM).
- Very complicated, distributed arbitration.
Subsumption (Brooks 1986)

- Emphasis on sensing to action (via Augmented FSM).
- Very complicated, distributed arbitration.
- No learning.
Subsumption
(Brooks 1986)

• Emphasis on sensing to action (via Augmented FSM).

• Very complicated, distributed arbitration.

• No learning.

• Worked.
Architecture Lessons (Subsumption)
Architecture Lessons
(Subsumption)

• Action from perception can provide the further structure -- modules (behaviors).

• Modules also support iterative development / continuous integration.
Architecture Lessons (Subsumption)

- Action from perception can provide the further structure -- modules (behaviors).
- Modules also support iterative development / continuous integration.
- Real time should be a core organizing principle -- start in the real world.
Architecture Lessons (Subsumption)

- Action from perception can provide the further structure -- modules (behaviors).
- Modules also support iterative development / continuous integration.
- Real time should be a core organizing principle -- start in the real world.
- Good ideas can carry bad ideas a long way (no learning, hard action selection).
• Goals ordering needs to be flexible.
• Goals ordering needs to be flexible.

• Maybe spreading activation is good for this.
SA: Layers vs. Behaviours

- Relationship not evident except in development!
SA: Layers vs. Behaviours

- Relationship not evident except in development!

A Robust Layered Control System for a Mobile Robot

Diagram:
- Sensors → perception
- perception → modelling
- modelling → planning
- planning → task execution
- task execution → motor control
- motor control → Actuators

Diagram:
- robot
  - collide
    - map
      - sonar
  - command
    - halt
      - motor
  - force
    - feelforce
      - runaway

Diagram:
- reason about behavior of objects
  - plan changes to the world
    - identify objects
  - monitor changes
    - build maps
  - explore
  - wander
  - avoid objects
- Actuators
SA: Layers vs. Behaviours

A Robust Layered Control System for a Mobile Robot

Diagram showing the relationship between layers and behaviours in a mobile robot control system.
SA: Layers vs. Behaviours

A Robust Layered Control System for a Mobile Robot

reason about behavior of objects

plan changes to the world

identify objects

monitor changes

build maps

explore

wander

avoid objects

Actuators
SA: Layers vs. Behaviours

Cognitive Analyses
- Frontal Cortex
- Hippocampus & Septum
- Sensory Cortex
- Amygdala
- Thalamus

Response Suppression
- Conditioned Emotional Responses
- Species-specific Responses: Freeze/Flight/Fight
- 'Startle' Responses
- Reflexive Withdrawal

Motor, Autonomic, & Endocrine Output

Control System for a Mobile Robot

perception  modelling  planning  task execution  motor control

Actuators

tional decomposition of a mobile robot control system into functional

reason about behavior of objects
- plan changes to the world
- identify objects
- monitor changes
- build maps
- explore
- wander
- avoid objects

Actuators
Layered or Hybrid Architectures

1. Incorporate behaviors/modules (action from sensing) as “smart” primitives.

2. Use hierarchical dynamic plans for behavior sequencing.

3. (Allegedly) some have automated planner to make plans for layer 2.

- Examples: Firby/RAPS/3T (‘97); PRS (1992-2000); Hexmoore ‘95; Gat ‘91-98
Belief, Desires, Intentions (BDI)

- **Beliefs**: Predicates
- **Desires**: goals & related dynamic plans
- **Intentions**: current goal

![PRS-CL Architecture](image)
Procedural Reasoning System

PRS-CL Architecture

Execution Cycle
1. New information arrives that updates facts and goals
2. Acts are triggered by new facts or goals
3. A triggered Act is intended
4. An intended Act is selected
5. That intention is activated
6. An action is performed
7. New facts or goals are posted
8. Intentions are updated
Procedural Reasoning System

- BDI

**Execution Cycle**
1. New information arrives that updates facts and goals
2. Acts are triggered by new facts or goals
3. A triggered Act is intended
4. An intended Act is selected
5. That intention is activated
6. An action is performed
7. New facts or goals are posted
8. Intentions are updated
Procedural Reasoning System

• BDI
• And reactive (responds to emergencies by changing intentions.)
Procedural Reasoning System

- BDI
- And reactive (responds to emergencies by changing intentions.)
- Er... once or twice (Bryson ATAL 2000).
Architecture Lessons

• Structured dynamic plans make it easier to get your robot to do complicated stuff.

• Automated planning (or for Soar, chunking/learning) is seldom actually used.

• To facilitate that automated planning, modularity is often compromised.
Soar as a 3LA

• **Reflection** on Top.

• Sense & Action separated!

• (Davis & Sloman 1995)
- Reflection on Top.
- Sense & Action separated!
- Hierarchy in AS; Goal Swapping (Alarms).
- (Sloman 2000)
CogAff

- Reflection on Top.
- Sense & Action separated!
- Hierarchy in AS, Goal Swapping (now reactive).
- Current Web
Architecture Lessons
(CogAff)
• Maybe you don’t really want productions as your basic representation -- you may want to come between a sense and an act sometimes.
Architecture Lessons (CogAff)

• Maybe you don’t really want productions as your basic representation -- you may want to come between a sense and an act sometimes.

• Aaron Sloman thinks about a lot more human cognitive traits than I do.