Intelligence by Design

Joanna J. Bryson

University of Bath
Department of Computer Science
Artificial models of natural Intelligence (AmonI)

jjb@cs.bath.ac.uk
http://www.cs.bath.ac.uk/~jjb
The Goals

Engineering: Building complete, complex agents which behave, learn and plan.

Science: Testing models of animal cognition.

- Complete agents: artifacts with identity
  e.g. robots, VR characters, tutors, monitors, artificial life.

- Complex agents:
  - multiple, potentially conflicting goals, and
  - multiple, mutually-exclusive means of achieving those goals.
Outline

- Introduction: Artificial Intelligence
- Components of Agent Intelligence
  - Behaviours and Modularity
  - Action Selection and Reactive Planning
  - Behaviour Oriented Design
- Modelling Natural Intelligence
- The Semantic Web?
- Conclusions
Artificial and Natural Intelligence

Artificial Intelligence has two goals (Winston):

- **Engineering** — Trying to improve computer science by emulating natural intelligence.
- **Science** — Trying to understand natural intelligence by designing and testing models.
Artificial and Natural Intelligence

Both fields share some challenges, therefore they may also share some solutions.

- Limits in storage capacity, processing time.
- Characteristics of information environment e.g. light, sound, gravity.
- Combinatorial complexity of planning, learning.
Combinatorics and Search

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

• Then . . .
  – Finding a one-step plan may take 100 tries.
  – Finding a two-step plan, may take $100^2$ (10,000).
  – For a plan of unknown length, agent may search forever, missing a critical step or sequence.
Combinatorics is the Problem

- Search is the only solution.
  - Planning and Learning
  - Design and Evolution

- The task of intelligence is to focus search (provide bias, constraint).
  - Develop good search techniques.
  - Limit search space to likely solutions.
Outline

• Introduction: Artificial Intelligence

• Components of Agent Intelligence
  – Behaviours and Modularity
  – Action Selection and Reactive Planning
  – Behaviour Oriented Design

• Modelling Natural Intelligence

• The Semantic Web?

• Conclusions
Modularity

• Modularity simplifies design.
  – Decomposes the problem into simpler units.
  – Focuses search by using locally optimal representations (planning, learning, perception).

• It also generates design issues.
  – Decomposition
  – Coordination
  – Learning
Modularity in Behaviour-Oriented Design

BOD (Bryson 2001) exploits modularity to limit search while addressing modularity’s problems:

- Learning is done within modules.
- Modular decomposition is dictated by variable state.
- Coordination between modules is done by hierarchical reactive plans.
What is a Behavior? (in BOD)

- A module in a complete, complex agent.
- Control for agent’s actions (expressed and/or internal).
- Perception required for that control.
- Memory required for perception or control.

BOD extends Behavior-Based AI (Brooks 1986, 1991) to be more like Object-Oriented Design. BBAI is a little like Fodor (1983) but modules are actors, no common representation.
A Simple Behavior

screeching
A Behavior with State

- screeching
- screeching-now?
- pulse-duration
Behaviors with Perception

- screeching
- screeching-now?
- pulse-duration

Known, liked

- recognize
- familiarity-levels
- affinity-levels
Behaviors that Aren’t Objects

face recognizer

identity

screeching
screeching-now?
pulse-duration

recognize
familiarity-levels
affinity-levels

known, liked
Behaviors with Processes and/or Triggers

- **Action Selection**
  - inhibit
    - screeching
      - screeching-now?
    - pulse-duration
    - inhibit-STM

- **Face Recognizer**
  - identity
    - recognize
      - familiarity-levels
    - affinity-levels

- known, liked
Example: A Mobile Robot
(Bryson ATAL97)
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

E-Memory  
* directions  
* times

direction, time

csense, odometry

Robot  
(and C-Sense)

DP-Land  
\( x, y \)  
in-dir
out-dir

Robot (and C-Sense)
Outline

• Introduction: Artificial Intelligence
• Components of Agent Intelligence
  – Behaviours and Modularity
  – Action Selection and Reactive Planning
  – Behaviour Oriented Design
• Modelling Natural Intelligence
• The Semantic Web?
• Conclusions
Action Selection and Reactive Plans

- Modularity leads to coordination problems.
  - Behavior arbitration
  - Multi-agent coordination
  - Action selection
- Reactive plans are an engineered solution.
  - Planning
  - Reactive Planning
  - Reactive Plans
The Place of Reactive Planning in an Agent
Reactive Plans in BOD

- Use hierarchy to limit search.
- Support three types of action selection problems:
  - Some things need to be checked at all times.
  - Some things only need considering in particular contexts.
  - Some things reliably follow from others.
Some Things Follow: *Action Patterns*

⟨get a banana → peel a banana → eat a banana⟩
Are Production Rules Better than Sequences?

(have hunger) ⇒ get a banana
(have a banana) ⇒ peel a banana
(have a peeled banana) ⇒ eat a banana
Are Production Rules Better than Sequences?

(have hunger) $\Rightarrow$ get a banana
(have a banana) $\Rightarrow$ peel a banana
(have a peeled banana) $\Rightarrow$ eat a banana
No — A Sequence is State

\langle \text{get a banana from left } \rightarrow \text{ pass a banana to right} \rangle

(left neighbor offers banana) \Rightarrow \text{ get a banana from left}

(have a banana) \Rightarrow \text{ pass a banana to right}
Basic Reactive Plans: State + Flexibility

(hungry) \Rightarrow

\begin{align*}
&\text{(full)} \Rightarrow \text{goal} \\
&\text{(have peeled banana)} \Rightarrow \text{eat banana} \\
&\text{(have a banana)} \Rightarrow \text{peel a banana} \\
&\Rightarrow \text{get a banana}
\end{align*}

Many different *expressed* plans (sequences of behavior) are determined by one *reactive* plan.
Parallel-rooted, Ordered Slip-stack
Hierarchical (POSH) Action Selection

- **Action Pattern:** \( \iota_1, \iota_2, \ldots, \iota_n \)

- Basic Reactive Plans: set of steps \( \{ \langle \pi_i, \rho_i, \alpha_i \rangle \}^* \)
  - **Competence:** competence step \( \langle \pi, \rho, \alpha, \eta \rangle \)
  - **Drive Collection:** drive \( \langle \pi, \rho, \alpha, A, \nu \rangle \)
    * No stack (3,000Hz on a 486)
    * Action scheduler (256Hz on a PentiumII)
Drive Collections:
BRPs for Environment Monitoring

\[ \text{life} \Rightarrow \left\langle \begin{array}{l}
\text{(something looming)} \Rightarrow \text{avoid} \\
\text{(something loud)} \Rightarrow \text{attend to threat} \\
\text{(hungry)} \Rightarrow \text{forage} \\
\Rightarrow \text{lounge around}
\end{array} \right\rangle \]
Action Selection in BOD

- Some things reliably follow from others — Action Patterns.
- Some things only need considering in particular contexts — Basic Reactive Plans.
- Some things need to be checked at all times — Drive Collections.
life (D)

<table>
<thead>
<tr>
<th>talk [1/120 \text{ Hz}] (worth_talking \top)</th>
<th>speak</th>
</tr>
</thead>
<tbody>
<tr>
<td>bump (bumped \top)</td>
<td>yelp reg.bump back_off clear.bump lose.direction</td>
</tr>
<tr>
<td>sense (C) [7 \text{ Hz}]</td>
<td>look</td>
</tr>
<tr>
<td></td>
<td>compound.sense</td>
</tr>
<tr>
<td>halt (has_direction \top) (move_view 'blocked)</td>
<td>lose.direction</td>
</tr>
<tr>
<td>start (has_direction \bot)</td>
<td>pick_open_dir</td>
</tr>
<tr>
<td>continue</td>
<td>move narrow (move_view 'clear) correct_dir</td>
</tr>
<tr>
<td>wait</td>
<td>snore sleep</td>
</tr>
</tbody>
</table>

walk (C)
walk (C)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>halt</td>
<td>(has_direction (\top)) (move_view 'blocked)</td>
</tr>
<tr>
<td>cogitate_route (C)</td>
<td>enter_dp (in_dp (\bot)) (entered_dp (\bot))</td>
</tr>
<tr>
<td></td>
<td>lose_direction greet_dp</td>
</tr>
<tr>
<td></td>
<td>lose_direction</td>
</tr>
<tr>
<td></td>
<td>enter_dp (in_dp (\top)) (entered_dp (\top))</td>
</tr>
<tr>
<td></td>
<td>lose_direction greet_dp</td>
</tr>
<tr>
<td></td>
<td>dismiss_dp</td>
</tr>
<tr>
<td></td>
<td>look_up (untried_near_neighbor (\top))</td>
</tr>
<tr>
<td></td>
<td>pick_near_neighbor</td>
</tr>
<tr>
<td></td>
<td>pick_previous_direction</td>
</tr>
<tr>
<td></td>
<td>keep_going (continue_untried (\top))</td>
</tr>
<tr>
<td></td>
<td>pick_previous_direction</td>
</tr>
<tr>
<td></td>
<td>desperate_look_up (untried_far_neighbor (\top))</td>
</tr>
<tr>
<td></td>
<td>pick_further_neighbor</td>
</tr>
<tr>
<td>start (has_direction (\bot))</td>
<td>ask_directions</td>
</tr>
<tr>
<td>continue</td>
<td>move narrow (move_view 'clear) correct_dir</td>
</tr>
</tbody>
</table>
Outline

• Introduction: Artificial Intelligence
• Components of Agent Intelligence
  – Behaviours and Modularity
  – Action Selection and Reactive Planning
  – Behaviour Oriented Design
• Modelling Natural Intelligence
• The Semantic Web?
• Conclusions
BOD: Initial Decomposition

1. Specify (high-level) what the agent will do.
2. Describe activities as sequences of actions. reactive plans
3. Identify sensory and action primitives from these sequences.
4. Identify the state necessary to enable the primitives, cluster primitives by shared state. behaviors
5. Identify and prioritize goals or drives. drive collection
6. Select a first behavior to implement.
BOD: Cyclic Development

• Scale the system.
  – Code behaviors and / or plans.
  – Test and debug code.

• Simplify the design.
  – Revise the specifications.
Simplifying the Design

Exploit trade-offs between representations.

- Behavior Modules
- Reactive Plans
BOD: Example
Control State Only

walk $\Rightarrow$ \[
\begin{align*}
\text{(left-feeler-hit)} & \Rightarrow \text{avoid-obstacle-left} \\
\text{(right-feeler-hit)} & \Rightarrow \text{avoid-obstacle-right} \\
& \Rightarrow \text{walk-straight}
\end{align*}
\]

avoid-obstacle-left $\Rightarrow$ \{walk backwards $\rightarrow$ walk right $\rightarrow$ walk left\}

avoid-obstacle-right $\Rightarrow$ \{walk backwards $\rightarrow$ walk left $\rightarrow$ walk right\}
Deictic State as Well

avoid-hit, feeler-hit, compensate-avoid  \[\text{deictic-avoid} \]
\[\text{hit-left} \rightarrow \text{feeler info}\]

walk  \[\Rightarrow \]
\[\langle \text{(feeler-hit)} \Rightarrow \text{avoid-obstacle} \]
\[\Rightarrow \text{walk-straight} \rangle\]

avoid-obstacle  \[\Rightarrow \langle \text{walk backwards} \rightarrow \text{avoid hit} \rightarrow \text{compensate avoid} \rangle\]
Specialized State (rather than Deictic)

\[
\text{walk} \Rightarrow \left\langle \begin{array}{c}
(feeler\text{-}hit) \Rightarrow \text{store\text{-}obstacle back\text{-}up} \\
\Rightarrow \text{find\text{-}way}
\end{array} \right\rangle
\]
Revising the Specification: State

• Prefer the simplest.
  1. Control State
  2. Deictic State
  3. Specialized State (learning)
  4. Meta-State (learning to learn)

• Exceptions:
  – Eliminate Plan Redundancy
  – Reduce Plan Complexity
Revising the Specification: Control

- Prefer the simplest.
  - Single Primitive > Sequence
  - Sequence > BRP
  - Control State > Variable State

- Exceptions:
  - Want part of primitive ⇒ sequence.
  - Sequence elements repeated, skipped ⇒ BRP.
  - Use variable state to:
    - Replace lots of triggers.
    - Generalize control state.
Outline

- Introduction: Artificial Intelligence
- Components of Agent Intelligence
  - Behaviours and Modularity
  - Action Selection and Reactive Planning
  - Behaviour Oriented Design
- Modelling Natural Intelligence
- The Semantic Web?
- Conclusions
Natural Intelligence Models

- Hierarchical Action Selection (Bryson 2000)
- Transitive Inference in Monkeys & Children (Bryson 2001)
- Competing Learning Systems in Primate Task Learning (in progress)
- Social Organization in Different Species of Macaques (Bryson 2002/3 and in progress)
Macaque Social Order

- Some (e.g. Rhesus) show strict, hierarchical order, also violent but infrequent conflict.

- Some (e.g. Stumptail) show egalitarian social order, more frequent but less violent conflict.
Hypotheses of Macaque Social Order

- Less resources (e.g. food) $\Rightarrow$ more violence $\Rightarrow$ selective pressure for social structure (Hemelrijk 2001, 2002)
- New conflict resolution behavior $\Rightarrow$ less violence $\Rightarrow$ less pressure for social structure (de Waal 2001, Flack *in prep.*)
Modelling Tolerance

- A very basic conflict resolution behaviour.
- Modelling emotions and other chemical drives as modules.
- Model ran in 3 conditions:
  - No tolerance (ignore peers.)
  - Tolerate if they groom you.
  - Tolerate if they want to groom you.
No Communication  Wait on Groom  Wait on Approach

Percentage Waking Time in Activity

[2*se, p< .05]
[mean, N=16]
grooming
trying to groom
untangling
## Basic Social Behaviors

<table>
<thead>
<tr>
<th></th>
<th>Navigate</th>
<th>Groom</th>
<th>Explore</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>state</strong></td>
<td>x, y, size, name focus-of-attn</td>
<td>drive-level, partner groomed-when, being-groomed?</td>
<td>drive-level direction-of-interest</td>
</tr>
<tr>
<td><strong>actions</strong></td>
<td>approach, wait, align, untangle</td>
<td>groom, choose-partner partner-chosen? tolerant, notify</td>
<td>choose-new-location lose-target, explore want-novel-loc?</td>
</tr>
</tbody>
</table>
No Communication  Wait on Groom  Wait on Approach

Percentage Waking Time in Activity

[2*se, p< .05]
[mean, N=16]
grooming
trying to groom
untangling
Dynamic Emotion Representation
Emmanuel Tanguy (w/ Willis, Bryson 2003)

Disposition to anxiety \[\rightarrow\] Emotional stability

Mood Layer

Emotional Stimuli

Mood state

Energy \[\uparrow\downarrow\] Tension \[\uparrow\downarrow\]

Mood emotional stimuli filter

Secondary Emotion Layer

Emotional Stimuli

Secondary Emotion State

Anger \[\uparrow\downarrow\] Joy \[\uparrow\downarrow\] Sadness \[\uparrow\downarrow\]

Surprise \[\uparrow\downarrow\] Disgust \[\uparrow\downarrow\] Fear \[\uparrow\downarrow\]

Facial Mood

Tension Levels

Facial expression representing the tension

Composition Expression

Combine the tension & emotional expressions

Expression

Facial Emotion

Emotion Levels

Facial expression representing the emotion

Expression
Emotions as Behaviour Modules
Emmanuel Tanguy (w/ Willis, Bryson in prep.)
BOD and the Semantic Web
(Bryson, Martin, McIlraith and Stein 2002)

- Web Services are a form of behaviour.
- DAML-S, the language for coordinating web services, does action selection.
POSH and DAML-S

- Data: reliability of state in service/modules must be a functional attribute in the service profile.
- Primitives: must specify timeout behaviour, prefer anytime response.
- Sequences: specify duration of elements, how they terminate.
- BRP: need flexible BRP-like plan element.
- Agent-Level Control: Needed, nice if continuous with rest of DAML-S.
Outline

• Introduction: Artificial Intelligence
• Components of Agent Intelligence
  – Behaviours and Modularity
  – Action Selection and Reactive Planning
  – Behaviour Oriented Design
• Modelling Natural Intelligence
• The Semantic Web?
• Conclusions
Related Work: Reactive Control

- Behavior-Based and Production Rule systems, e.g. Subsumption (Brooks 1986), ANA (Maes 1992), Soar (Newell 1990), specialize in emergencies.

- Plan-based hybrids, e.g. PRS (Georgeff & Lansky 1987), JAM (Huber 1999), 3T / RAPs (Bonasso, Firby et. al 1997), specialize in order.

- Only Teleo-Reactive (Nilsson 1994) has BRPs and a user base. No user base: (Fikes 1972) (Correia and Steiger-Garção 1995) (me).
Related Work:
Behaviors, Modularity and Learning

- Behavior-Based AI has modularity and specialized learning, but overly diffuse control. (e.g. Brooks 1991, Horswill 1993)

- Hybrid systems have reactive plans, but reduce behaviors to mere primitives, have overly monolithic representations.
Summary

- Combinatorics and search are the main problems of intelligence.
- Building AI requires good engineering, even if it’s going to learn (or plan) on its own.
- Behavior-Oriented Design: Modularity plus hierarchical reactive plans.
Future Work: Primates

- Hierarchical Action Selection (Bryson 2000)
- Transitive Inference in Squirrel Monkeys (Bryson 2001)
- Competing Learning Systems in Cotton-Top Tamarins (in progress)
- Social Organization in Different Species of Macaques (more! replications, more conflict resolution)
- Language and Memetics
Future Work: Agent Architectures

- Elements in BOD
  - Behaviors: Neocortex
  - Action Selection: Basal Ganglia, Periaqueductal Gray Matter
  - Drive Collection: Amygdalic System

- Other Possible Additions
  - Smoothing and Interpolation (Cerebellum?)
  - Episodic Memory (Hippocampus?)
Future Work: Learning

- Learning in Behaviors
- Learning of Plans
  - Search
  - Evolution
  - Imitation
- Learning Behaviors
  - Existing work is one behavior in BOD.
  - Learning dynamical models (e.g. Hogg, Brand)
Future Work: Tools and Applications

• Tools to facilitate agent construction.
  – Improve proprietary BOD tools.
  – Combine BOD (or at least POSH) with RePast (Collier 2001).
  – Create Primate Social Simulator.

• Applications outside A Life.
  – VR characters for entertainment.
  – Tutoring systems.
  – Intelligent environments.
Is BOD Biologically Plausible?
(Bryson EmerNet00)

- Elements in BOD
  - Behaviors: Neocortex
  - Action Selection: Basal Ganglia, Periaqueductal Gray Matter
  - Drive Collection: Amygdalic System

- Other Possible Additions
  - Smoothing and Interpolation (Cerebellum?)
  - Episodic Memory (Hippocampus?)
Revising the Specification – BRPs

- A BRP is a worst-case scenario sequence backwards.
- A BRP should only have 3-7 elements.
- Too many elements or triggers:
  - Two ways to do same goal $\Rightarrow$ make sibling BRPs
  - Multi-step subgoal $\Rightarrow$ make child sequence or BRP.
- Be careful of termination.
  - Converge to goal.
– Fail if goal is impossible (habituate).
– Manage chaining.
Learning Behaviors