Intelligent Robotics in 30 Minutes

Joanna J. Bryson
Artificial models of natural Intelligence
University of Bath
http://www.cs.bath.ac.uk/ai/AmonI.html
Outline

• Introduction
  • Why is it hard to be smart?
  • What works?
• Behavior Oriented Design
  • Modular Control
  • Action Selection
• Iterative Development
Outline

• Introduction
  • Why is it hard to be smart?
  • What works?
• Behavior Oriented Design
  • Modular Control
  • Action Selection
• Iterative Development
Why is it hard to be smart?

- Pretend you bought a robot, and it came with 100 things it knew how to do without being told.

- For example, eat, sleep, turn right, turn left, step forward, step backward, pick things up, drop them...

- Now pick a goal for your robot.

- For example, flying to Tokyo.
Suppose you can’t be bothered to tell your robot exactly how to get to Tokyo, so you have it guess.

If getting to Tokyo takes one step, the robot may have to try 100 different things.

If it takes two steps, the robot may have to try each thing after each thing:

$$100^2 = 10,000$$
The hardness of smartness (3)

- If the robot doesn’t know how many steps it takes to go to Tokyo, it might get caught in an infinite loop.

- For example, it might eat, sleep, work, eat, sleep, work, eat, sleep, work... and never buy a passport.

- When computer scientists say “hard” they mean “pretty much intractable.”

Sony SDR-4Xs. Pictures from BBC
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 `intelligent’ behavior has no or little **real-time** search (not cognitive).

• For **artificial** intelligence, most focus from **design**.
Outline

• Introduction
  • Why is it hard to be smart?
  • What works?
• Behavior Oriented Design
  • Modular Control
  • Action Selection
• Iterative Development
Intelligence

• What matters is expressing the right behaviour at the right time.

• Decompose the problem:
  • Behaviour module: combination of code and mechanism.
  • Action selection: detect when to express which behaviour.
  • Both require good perception.
What works

• **Modularity**: simplifies design (Brooks 1986), allows locally-optimal representations (Bryson PhD 2001, Bryson IJCAI 2001).

• **Action selection** (sequencing): specifies goal prioritisation and ordering (Bryson JETAI 2000, PhD).

• **Iterative develop & test**: cycle in increasing complexity (object-oriented design; agile development (Beck 2000, Bryson IJCAI, PhD).
Outline

• Introduction

• Why is it hard to be smart?

• What works?

• Behavior Oriented Design

• Modular Control

• Action Selection

• Iterative Development
Behavior Oriented Design

• **Modularity**: simplifies design *(Brooks 1986)*, allows locally-optimal representations *(Bryson PhD 2001, Bryson IJCAI 2001)*.

• **Action selection** *(sequencing)*: specifies goal prioritisation and ordering *(Bryson JETAI 2000, PhD)*.

• **Iterative develop & test**: cycle in increasing complexity *(object-oriented design; agile development* *(Beck 2000, Bryson IJCAI, PhD)*).
Outline

• Introduction
  • Why is it hard to be smart?
  • What works?
• Behavior Oriented Design
  • Modular Control
  • Action Selection
• Iterative Development
Behavior Modules

- Generate & control actions.
- Sense & interpret perception for actions.
- Store & maintain memory for perception.
Behavior Modules

- **Objects** (in an object-oriented language):
  - **Methods** provide interface with other modules, including action selection.
Example for BURST

- **Recognize gate** and **compute trajectory to centre**.

- Do this at 10Hz, adjust velocity slowly, bad frames won’t matter if there aren’t too many.

- **OR**

- Remember previous 3 guesses, don’t report new one if too different.
Issues of Modularity

Get Fuzzy (Conley 2006)
Issues of Modularity

• How do you get ordered behavior for the complete agent?

• How do you decide what goes in which module?
Issues of Modularity

• How do you get ordered behavior for the complete agent?

• How do you decide what goes in which module?
Outline

- Introduction
  - Why is it hard to be smart?
  - What works?
- Behavior Oriented Design
  - Modular Control
  - Action Selection
- Iterative Development
Simple Action Selection

- Sets of productions
- A sense (recognize your context) and an action.
What is an action?

• In robotics / real-time systems, actions must be very brief, because the context shifts very quickly.

  • **Drive through the hoop** is a dangerous atomic action, what if you turn a little?

  • **Increase thrust towards hoop center** is safer, it can be reevaluated frequently.

  • Sometimes open loop is necessary.
Simple Action Selection

• Sets of productions
• A sense (recognize your context) and an action.
• Sadly, robotics isn’t simple.
Context in Action Selection

• Recognize gate and compute trajectory to centre (production).

• What if you are already through gate and were just looking for a floating target?

• Don’t want to be inappropriately “captured” by action in the wrong context.
Contextual Action Selection

- **Sequence:** when one production is finished, move to next one.

- **Hierarchy:** allow several productions to operate within one higher-level context.

- **Priority:** if more than one production could fire, say which is most important.
Example for BURST

• Top of hierarchy is a sequence: through gate, drop weight, surface under balloon.

• Prioritised productions for drop weight:

  1. If stopped over target, drop weight.
  2. If see floor target, halt on top.
  3. [default], go towards beacon.
Issues of Modularity

- How do you get ordered behavior for the complete agent?
- How do you decide what goes in which module?
Issues of Modularity

• How do you get ordered behavior for the complete agent?

• How do you decide what goes in which module?
Outline

- Introduction
  - Why is it hard to be smart?
  - What works?
- Behavior Oriented Design
  - Modular Control
  - Action Selection
- Iterative Development
Behavior Oriented Design

1. Add a behavior module: how to act.
2. Add a bit of plan: when to act.
3. Test it works.
4. Go back to 1.
Behavior Oriented Design

- If a module is getting too big and complicated, take it a part, connect the parts with plans.

- If a plan is getting too complicated, break it up with hierarchy OR build a smarter module to solve some of the problems.

- Often perception is key!
Bryson’s first law of intelligent robotics

No robot works the first time you run it.
Outline

- Introduction
  - Why is it hard to be smart?
  - What works?
- Behavior Oriented Design
  - Modular Control
  - Action Selection
- Iterative Development
Intelligent Robotics in 30 Minutes

Joanna J. Bryson
Artificial models of natural Intelligence
University of Bath
http://www.cs.bath.ac.uk/ai/AmonI.html
Building Intelligence

- AI normally associated with clever algorithms.
- No one algorithm produces working systems from a vision (or even first-cut specs.)
- **Behavior Oriented Design** is a methodology.
- Optimize balance between human and machine search for the right behavior.
Intelligent Systems

- Complete, complex agents:
  - Multiple, potentially conflicting goals.
  - Multiple, mutually exclusive means of achieving a goal.

- Robots:
  - Real-time, dynamic environments.
  - Sense and change (act in) that environment.
Intelligence

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

• Conventional AI planning searches for the right set of actions, requires sets of primitives.

• Learning searches for the right parameter values, requires primitives and parameters.

• parameter: variable state.

• Evolution and development are learning.