From Extropians to Evolutionary Robotics

What Machines (Don't) Tell Us About (Trans)Human Nature

Simon D. Levy
PHIL 330
18 May 2011
MEET THE EXTROPIANS

By Ed Regis

Extropians want a society of self-generating "spontaneous order," not massive legal structures imposed from above. They also want as much control over the physical universe as is compatible with natural law. You might say they want to become more than human - transhuman.

GROWTH!

There's been nothing like this movement - nothing this wild and extravagant - since way back in those bygone ages when people believed in things like progress, knowledge, and - let's all shout it out, now -
Life at Wal-Mart
Charles Platt at 11:03 PM Sunday, Feb 1, 2009

(Charles Platt is a guest blogger)
Extropians want a society of self-generating "spontaneous order," not massive legal structures imposed from above. They also want as much control over the physical universe as is compatible with natural law. You might say they want to become more than human—transhuman.

There's been nothing like this movement—nothing this wild and extravagant—since way back in those bygone ages when people believed in things like progress, knowledge, and—let's all shout it out, now—GROWTH!
So Where Are the Freakin’ Robots Already?!
So Where Are the Freakin’ Robots Already?!

1927

1956
So Where Are the Freakin’ Robots Already?!
So Where Are the Freakin’ Robots Already?!
So Where Are the Freakin’ Robots Already?!
They’re Heee-eere (sort of)
Group Exercise:
Be the Crane!
Why Is It So Difficult?

Pollack, Lipson, Hornby, & Funes (2001):

We propose that both the morphology and the controller should evolve at the same time.
Issues

• Central goal: *higher complexity at lower cost*

• Only possible when design & construction are fully automatic

• In real evolution, brain & body co-evolve in “a long series of mutual adaptations”
Key Technologies

• Increasing fidelity of mechanical simulation
• Rapid prototyping
• Better understanding of coevolutionary dynamics / self-organizing systems
Evolution of Coevolution

- Early work: Hillis sorting networks (1992)
- Problems as parasites, solutions as hosts
Evolution of Coevolution

K. Sims (!) body/brain coevolution in simulation (1994)
Body/Brain Coevolution: Simulator Desiderata

- *Representation* - simulator should cover a “universal space” of mechanisms
- *Conservative* - “margin of safety” for imperfections (c.f. Clark’s “007 Principle”)
- *Efficient* - simulator should be (much) faster than reality
- *Buildable* - results should be convertible from simulator to real robot
Generation I: Legobots

- Minimal simulator - forces for joining, separating LEGO pieces
- No brain *per se*; just body
- Structure must be viable at each stage of evolution
Generation I: Legobots

Task #1: support a small weight

Wednesday, May 18, 2011
Generation I: Legobots

Task #2: bridge a gap
Generation I: Legobots

The constructed bridge:
Generation I: Legobots

Task #3: lift a small weight:
Generation I: Legobots

Main point: evolved three kinds of structure normally associated with design

• Table
• Cantilever
• Triangular support
Generation II: Genetically Organized Lifelike Electro-Mechanics (GOLEM)
FYI: Golem, the Original Cyborg
Generation III: Modularity
Generative Design
(Tinkerbots)

• Problems with first two generations
  • No *modularity* (reusability); just mutation
  • Can’t *scale up* to complex designs
Modularity

• Real engineers use same components (modules) in many different solutions: engines, wheels, ...

• In a more subtle way, nature does too: hearts, limbs, ...
Generativity

- Nature doesn’t directly mutate or recombine modules; instead, the code (DNA) gets recombined/mutated and then generates the individual.

- From an engineering perspective, this sort of approach supports much larger, more complex structures.
Generativity
Generativity via L-Systems

• A simple rule-based system inspired by plant growth

• Start with a variable (symbol), then replace it by other variables and numbers, and repeat for new symbols
L-Systems Example #1

variables : A, B
constants : none
start : A
rules : (A \rightarrow ABA), (B \rightarrow BBB)

http://en.wikipedia.org/wiki/L-system
L-Systems Example #1

variables: A, B
constants: none
start: A
rules: (A → ABA), (B → BBB)

http://en.wikipedia.org/wiki/L-system
L-Systems Example #1

variables : A, B
constants : none
start   : A
rules   : (A → ABA), (B → BBB)

http://en.wikipedia.org/wiki/L-system
L-Systems Example #1

variables : A, B
constants : none
start : A
rules : (A → ABA), (B → BBB)

http://en.wikipedia.org/wiki/L-system
L-Systems Example #1

variables : A, B
constants : none
start : A
rules : (A → ABA), (B → BBB)

A
ABA
ABA
ABABABA
etc.

http://en.wikipedia.org/wiki/L-system
L-Systems Example #1

variables : A, B
constants : none
start : A
rules : (A → ABA), (B → BBB)

Let A mean "draw forward" and B mean "move forward".
L-Systems Example #2

variables : F
constants : 4, 5, 6, 7
start : F
rule : F → l[5+F][7-F]-l[4+F][6-F]-l[3+F][5-F]-IF

http://www.cs.unm.edu/~joel/PaperFoldingFractal/
Evolving L-Systems to Build Robots (Hornby 2001)

• “Genome” is L-system rules represented as sequences

• Can then crossover and mutate rules; e.g., (A → ABA) could mutate to (A → ABB), etc.

• Symbols (A, B) are treated as instructions for building robot components
Evolving L-Systems to Build Robots (Hornby 2001)
The Current Scene

Emergent Self-Model