Spatial Computing, Synthetic Biology, and Emerging IP Challenges

Jacob Beal
November, 2010
Spatial Computers

Robot Swarms

Biological Computing

Sensor Networks

Reconfigurable Computing

Cells during Morphogenesis

Modular Robotics
How can we program these?

- Desiderata for approaches:
  - Simple, easy to understand code
  - Robust to errors, adapt to changing environment
  - Scalable to potentially vast numbers of devices
  - Take advantage of spatial nature of problems

One answer: continuous space programs!
Example: Mobile Streaming
Example: Mobile Streaming
Example: Mobile Streaming
Example: Mobile Streaming

I want Alice to be able to listen in on this great conversation.
Geometric Program: Channel

(cf. Butera)
Geometric Program: Channel
Geometric Program: Channel

(cf. Butera)
Geometric Program: Channel

Source

Destination

(cf. Butera)
Geometric Program: Channel

Source  
Destination

(cf. Butera)
Geometric Program: Channel

(cf. Butera)
Geometric Program: Channel

(cf. Butera)
Computing with fields

source

gradient

+ 

destination

gradient

distance

<=

dilate

width
Computing with fields

source

gradient

+ 37

destination

gradient

distance

<=

width

10
dilate
Amorphous Medium

- Continuous space & time
- Infinite number of devices
- See neighbors' past state

Approximate with:
- Discrete network of devices
- Signals transmit state
Proto

(def gradient (src) ...)
(def distance (src dst) ...)
(def dilate (src n)
    (<= (gradient src) n))
(def channel (src dst width)
    (let* ((d (distance src dst))
            (trail (<= (+ (gradient src)
                        (gradient dst)) d)))
        (dilate trail width)))

[Beal & Bachrach, '06]
Proto's Families of Primitives

Pointwise

Feedback

Restriction

Neighborhood

delay

+ 41 7

+ 48

restrict

nbr

any-hood
Why use continuous space?

- Scaling & Portability
- Robustness
- Composability

![2000 devices](image1)

![150 devices](image2)
Energy Management

Zome Energy Networks

Swarm Robotics

Synthetic Biology

Morphogenetic Engineering

Proto

Device Kernel

(def gradient (src) ...)
(def distance (src dst) ...)
(def dilate (src n)
  (<= (gradient src) n))
(def channel (src dst width)
  (let* ((d (distance src dst))
          (trail (<= (+ (gradient src)
                      (gradient dst))
                 d)))
    (dilate trail width)))

Extending Flippers
Larger Motor Driver
Larger Drive Motor
Extending Tracks
IP Challenges

Many parts: free, protected, & commercializable?
IP Challenges

Many parts: free, protected, & commercializable?

Thank you, Creative Commons!
Zome Energy Networks

Swarm Robotics

Energy Management

Proto

Device Kernel

Synthetic Biology

Morphogenetic Engineering

(def gradient (src) ...)
(def distance (src dst) ...)
(def dilate (src n)
   (<= (gradient src) n))
(def channel (src dst width)
   (let* ((d (distance src dst))
           (trail (<= (+ (gradient src)
                         (gradient dst))
                    d)))
    (dilate trail width)))
Computation via Transcription Network

DNA → regulatory protein → RNA polymerase → ribosome → RNA → Decay → Protein
Proto BioCompiler

High-Level Language

(def band-detector (signal lo hi)
  (and (> signal lo)
       (< signal hi)))

(let ((v (diffuse (aTc) 0.8 0.05)))
  (green (band-detect v 0.2 1)))

Compile

Genetic Regulatory Network

Optimize

Simulate

Assemble

Living Cells

w. Weiss
Band detect: code

Proto

```
(def band-detector (signal lo hi)
  (and (> signal lo)
       (< signal hi)))

(let
  ((v (diffuse (aTc) 0.8 0.05)))
  (green (band-detect v 0.2 1)))
```

simpler, more reusable

Engineered Bacteria

[Beal & Bachrach, '08]  [Weiss '05]
Band detect: behavior

Proto

Engineered Bacteria

[Beal & Bachrach, '08]

[Weiss '05]
Motif-based Compilation

Dataflow Network

IPTG → not → green
Motif-based Compilation

Dataflow Network

IPTG → not → green

LacI outputs arg0 outputs arg0 GFP outputs
Motif-based Compilation

Dataflow Network

Genetic Regulatory Network
Classical Optimization can be Adapted

- Example: XOR circuit
Classical Optimization can be Adapted

- Example: XOR circuit

After optimization: ~50% improvement
And on to larger organisms...
IP Challenges

IP Types
- DNA sequences
- databases
- software
- patents
- organisms

Communities
- biologists
- computer scientists
- students
- industry
- CAD engineers

Regulators
- CDC
- FDA
- National Institutes of Health

Many components; integration with vendors
Full automation: no human interpretation
Summary

• Proto allows complex spatial computing problems to be solved with simple programs.
• Proto & other approaches beginning to link together to automate synthetic biology
• Major IP thunderclouds on the horizon...
Proto is available

http://proto.bbn.com
(or google “MIT Proto”)

• Includes libraries, compiler, kernel, simulator, platforms, tutorial
• Licensed under GPL (w. libc-type exception)