On The Path to Programmable Matter

Seth Copen Goldstein
seth@cs.cmu.edu

And, lots of other people!
www.cs.cmu.edu/~claytronics

SenSys, 11/07

Programmable Matter

• A programmable material...
• ...with actuation and sensing...
• ...that can change its physical properties ...
• ...under software control...
• ...and in reaction to external stimuli

See video at: www.cs.cmu.edu/~claytronics/

Types of “Programmable Matter”

• Modular Robots
• Sensor Network
• Claytronics
• Synthetic Biology
• FPGAs
• Nanotechnology
Scaling

• Goal: Form dynamic high-fidelity macroscale objects

High-fidelity ⇒ sub-millimeter units
Macroscale ⇒ millions of units
Scaling: Down in size
Up in number

Method: Ensemble Effect

Node Requirements

• Each node must have
  - Computation+Memory
  - Communication
  - Energy Storage
  - Sensing
• Yet, scaling demands nodes be:
  - Simple
  - Small
• Ensemble Axiom:
  A node should include only the functionality necessary to achieve the desired ensemble.

Challenges

• Challenges are all intertwined
  - Hardware/Software trade-offs
  - Ensemble Axiom
• Example: Energy

Hardware

Software

Design Space
Challenges

Node Ensemble

Hardware

Software

- Challenges are all intertwined
  - Hardware/Software trade-offs
  - Ensemble Axiom
- Example: Energy

Ensemble Effect & Motion

Building the Hardware

- Not if, but when

- Many approaches
  - MEMS
  - Bio
  - ?
- Many challenges, but a clear path exists
Current Approach to <1mm

- How to form 3D from a 2D process?
  - begin with foundry CMOS on SOI

A potential approach

- How to form 3D from a 2D process?
  - begin with foundry CMOS on SOI
  - pattern a flower that includes structure and circuits
  - lift off silicon layer
    - flexible
    - harness stress to form a sphere

A potential approach

- How to form 3D from a 2D process?
  - begin with foundry CMOS on SOI
  - pattern a flower that includes structure and circuits
  - lift off silicon layer
    - flexible
    - harness stress to form a sphere
A sanity check

1 mm diameter sphere
- Mass < 1 mg

Electrostatic Actuators
- ~5 body lengths / sec

Communication Capacitors

Power Storage
- Supercapacitor stores enough energy to execute over 200 million instructions or move 2 million body lengths

Power distribution
- Transmission of “energy packets” using capacitive coupling fills reservoir in < 1 μs.

Computation Capability
- 8086 Processor with 256KB memory
- SOI-CMOS 90 nm process with > 2M transistors.

Initial Experiments

Sensor Networks

- Sense physical phenomena
- Lots of nodes
- Compute something
- Communicating with each other

Result: an ensemble effect.
- i.e., The whole is greater than the sum of the parts

Sensor Networks as Programmable Matter

- Both have
  - Many nodes networked together
  - Nodes which can compute & store state
  - Sense the environment
  - Need uncertainty tolerance

- Sensor Networks nodes
  - May be unattended for long periods of time
  - Separate from each other
  - Severely energy constrained

- Programmable Matter
  - More nodes
  - Mobile nodes (actuators!)
  - Nodes in contact, often in a lattice

Rob Reid, AFRL
Two Broad Problem Areas

• Programming the Ensemble: How does one think about coordination of millions of elements?
• Programming the Unit: What is the programming model for a (single) element?

• Let’s focus on the ensemble

Thermodynamics for Computing

Traditional CS
Control each unit’s actions

Engineering Ensemble Effects
Control ensembles of active units

Phy/Chem/Econ
Control global properties of the aggregate

Number of Entities

10^0 10^1 10^2 10^3 10^4 10^5 10^6 10^7 ...

Goal: Understand methods for programming the ensemble as a whole.

Emergent Engineering

moveAround( X, Y, Point ) :-
neighbor( X, Y ),
brightness( X, N ),
brightness( Y, M ),
vacant( Y, Point ),
N <= M.

• Attributes:
  - Ensemble level thinking
  - Concise understandable program
  - Scalable
  - Amenable to proof
  - Robust to failure and environmental uncertainty

Automatic Sensor Calibration as Probabilistic Inference

• Sensor calibration
  - manufacturing and deployment introduces bias
  - sensor cannot remove bias independently

• Probabilistic inference for calibration:
  - exploit correlations between nearby sensors
  - use probabilistic graphical model to represent correlation

\[
P(B_{11} \mid \bar{m}_1, \ldots, \bar{m}_{54})
\]

\[
P(T_{11} \mid \bar{m}_1, \ldots, \bar{m}_{54})
\]

Guestrin, CMU
**E³ abstraction**

Specification

Logical

Overlay

Topological

Geometric

**In detail for probabilistic inference...**

Specification

Logical

Overlay

Topological

Geometric

**Building and Maintaining a Real Overlay**

Field deployment: 97 sensor nodes deployed in office building

13 messages per node to form junction tree (average)

robust to node and link failure

**A General Architecture**

Logical

Overlay

Topological

Geometric
The E³ abstraction

**Specification**
High-level description of the Objective, including goals, tasks, and constraints.

**Logical**
Algorithmic Implementation of Specification

**Overlay**
Distributed data structure (integrity, consistency, access)

**Topology**
Communication-centric relationship between nodes.

**Geometry**
Physical sensor node data (e.g., location, sensors)

Transforming Shapes

- **Shape Change in Claytronics**
- **Requires:**
  - Determine which robots can move
  - Keep modules connected
  - Parallel movement
  - Dealing with unit failure
- **One example:**
  - In C++ ~1000 lines

Morphing Ensembles of Robots

- Transform shape of ensemble from into target shape without disconnecting any nodes.
- Move leaves towards root of their tree.
- Cover surface with forest of trees. Roots are near target.
- Links between adjacent neighbors. Distinguish interior nodes from exterior ones.
- 3D location in ensemble consistent coordinate system.

Morphing Ensembles of Robots

- Transform shape of ensemble from into target shape without disconnecting any nodes.
- Move leaves towards root of their tree.
- Cover surface with forest of trees. Roots are near target.
- Links between adjacent neighbors. Distinguish interior nodes from exterior ones.
- 3D location in ensemble consistent coordinate system.
Morphing Ensembles of Robots

Transform shape of ensemble from target shape without disconnecting any nodes.

Move leaves towards root of their tree.

Cover surface with forest of trees. Roots are near target.

Links between adjacent neighbors. Distinguish interior nodes from exterior ones.

3D location in ensemble consistent coordinate system.

Ensemble Compilation

• Programs are 20x shorter
  Understandable & Proof friendly

• Ensemble Compilers are key:
  - Where data lives
  - Minimize communication
  - Keep data consistent

• E³ Abstraction ⇒ Efficient ensembles

---

The E³ abstraction exploits...

new compiler techniques

embed algorithm onto distributed data structure

Logical
Algorithmic Implementation of Specification

Overlay
The distributed data structures necessary to create ensemble effects from the local communication relationships.

Topology
Communication-centric relationship between nodes.

Geometry
The primary interface to the physical world And where the nodes are in the world.

Global Behavior from local rules

• Concise specifications
• Embarrassingly parallel

• Examples:
  - Amorphous computing [Nagpal]
  - Graph grammars [Klavins]
  - Programming work [Kod.
  - CA+Gradients [Stoy]
  - Hole motion [DeRosa]
  - Boyd model [Boyd]
  - Turing stripes

• Goal: Compile Global specification into unit rules
  Predict global behavior from set of unit rules

Global behavior

Local rules

Compile into

Predict

Program Length

Lines of Code

Gradient Routing Localize Morph

Runtime Messages

C++ Meld

0 5 10 15 20

0 5 10 15 20
Thermodynamics for Computing

Goal: Understand methods for programming the ensemble as a whole.

Programmable Matter

- Open up an entire new application space
  - Antennas (Programmable Antennas)
  - Design (100x protein model)
  - Entertainment (WoW in your living room)
  - Interaction (telepario)
  - Rescue (paramedic on demand)
  - Metal Man (fault tolerant robotics)

- Vehicle for studying CS problem of the future:
  How do you design, program, maintain, and use a billion component system?