Realizing Programmable Matter with Modular Robots

Julien BOURGEOIS
UBFC, FEMTO-ST, CNRS
France

http://members.femto-st.fr/julien-bourgeois/

Work funded by:
ISITE-BFC (contract ANR-15-IDEX-03),
ANR (ANR-16-CE33-0022),
ANR/RGC (ANR-12-IS02-0004-01 and 3-ZG1F),
ANR (ANR-06-ROBO-0009),
ANR (ANR-2011-BS03-005),
DARPA (FA87501010215),
NSF (CNS-0428738) and
Intel Corporation
Microsoft
Where do I come from?

• University Bourgogne Franche-Comté (UBFC)
• University of Franche-Comté (UFC)
• FEMTO-ST Institute/CNRS, 700 researchers and staff
  – Collegium Smyle with EPFL
• CNRS ranked #1 (article count) in Nature Research Index, 2018

Montbéliard (Peugeot Citroën car home city)
Programmable matter examples
Programmable matter

• Nice video but we cannot do magic!
  – Reconfiguration speed
    • Moving is slow, moving millions of modules is VERY VERY slow
    • 12 hours for reconfiguring 800 sliding-cubes!!
    • (or 11.66 hours for moving 1024 Kilobots)
  – Reliability
    • Having millions of modules, you WILL have failures
  – Sturdiness
    • Very few studies about mechanical resistance of such a complex system
Programmable matter applications

Complex surgery → Take an MRI → MRI imaging → 3D model → Interactive training → Programmable matter representation
Programmable matter applications

Complex part design

CAD model

Programmable matter representation

User modifications
Programmable matter applications

Sculpting a shape-memory polymer sheet
Outline

Our Vision

Hardware design

Software

Art

What’s Next?

Our Vision

Hardware design

Software

Art

What’s Next?
Outline

Hardware design

- Integration
- Electronics
- Geometry
- Actuation
- Structure
- Power
Claytronics Atoms: Catom

~meters (2006)

~decimeters (2007)

~centimeters (2007)

~millimeters (2012)

3D shape
Towards 3D: Geometrical basics

1. We replace connection points by **12 square connectors**.

2. Then we can place **8 hexagons** and **6 octagons**.
   - Truncated cuboctahedron

3. Electrostatic actuators make catoms turning around neighbors.
   - We place curved surface over hexagonal and octagonal faces.
   - These curves are part of cylinders and planes in order to obtain continuous surfaces

---

Motion examples
Outline

- Hardware design
- Structure
- Power
- Actuation
- Electronics
- Integration

Geometry
Structure
Structure
Structure
Outline

Hardware design

- Integration
- Electronics
- Actuation
- Geometry
- Structure
- Power
Power

- Experiments with 7.05 mW power consumption by an actuator
- Connected actuators
  - Work using the square of input
  - No need for diode
  - Higher power efficiency

Actuation

Pyralux flexible

LPKF Protolaser U3
Outline

Hardware design

Integration
Electronics
Geometry
Actuation
Structure
Power
**M³ : Michigan Micro-Mote : A mm³ Sensing Platform**

### mm³ generic sensing platform
- Modular die-stacked structure
- Enables diverse technology
- 12.3mm² in 2.5mm³
- Swappable layers

 Requires standard communication interface between layer

---

Slide from David Blaauw
## Types of Sensors

<table>
<thead>
<tr>
<th>Dimension</th>
<th>F Series</th>
<th>P Series</th>
<th>N Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>2 x 4 x 2 mm³</td>
<td>5 x 5 x 3 mm³</td>
<td>7 x 7 x 5 mm³</td>
</tr>
<tr>
<td>Sensing Modalities</td>
<td>Temp, Pressure, Light</td>
<td>Temp, Pressure, Light</td>
<td>+ Motion, Humidity</td>
</tr>
<tr>
<td>Lifetime</td>
<td>1 month</td>
<td>3 – 5 years</td>
<td>5 – 7 years</td>
</tr>
<tr>
<td></td>
<td>3 years (w/ harvesting)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radio Range</td>
<td>5 cm</td>
<td>1 m</td>
<td>20-50m</td>
</tr>
</tbody>
</table>

Credit: Cubework
Optical Programming: GOC

- GOC programming interface
  - Separated front-end for isolating light exposure within the system
  - Up to 840bps transmission by faster clock speed and larger frontend diode
  - Tradeoff between programming speed & sleep power

Programming M3 Stack with ICE Board via GOC

M3 Stack with Solar Cell and GOC Frontend on Top Layer
Outline

Integration
Electronics
Geometry
Actuation
Structure
Power

Hardware design
Integration of M3 mote and catom

- Integration on skeleton with flexible printed circuit
Integration of M3 mote and catom

- Electrical interconexions

Unstacking
Integration of M3 mote and catom

- Mote unstacking
- Chips bonded on a polyimide film with ACF
Construction of a 3D Catom from an unfold
The Blinky Blocks

- Micro-controller
  - ARM Cortex M0
- Sensors
  - IMU: Orientation and tapping
  - Microphone: Sound
- Actuators
  - 2 LEDs: Glow in different colors
  - Speaker: Play sounds
- Communications
  - 6 USART communications at 6Mbps max
Outline

- Our Vision
- Hardware design
- Software
- Art
- What’s Next?

Our Vision
Simulation environment

  - Multi-targets (Blinky Blocks, Smart Blocks, Robot Blocks, Claytronics)  
  - Multi-languages (C/C++, Meld, Javascript, Python)  
  - Interactive  
  - Include debugging  
  - Available in your web browser online at:  
  - **First MSR simulator on the web thanks to WebGL!**  

- One ambition: make VisibleSim the reference simulator for modular robots and distributed programming initiation

Dhoutaut Dominique, Piranda Benoit and Bourgeois Julien, "Efficient Simulation of distributed Sensing and Control Environments" in "iThings 2013, IEEE Int. Conf. on Internet of Things", Beijing, China, pp. 452--459, aug. 2013
Smart Blocks, Robot Blocks and Blinky Blocks

- **Smart Blocks**
- **Robot Blocks**
- **Blinky Blocks**
Catoms

- 3D catoms
- 2D catoms
Outline

Software Center

Mechanical modeling
Self-Reconfiguration
Coding maps
Synchronization
Simulator
Center
Algorithms for programmable matter

- Finding the center of a distributed system: ABC-Center, PC2LE, k-BFS-SumSweep [AINA 16] [IROS 15]
  - In real time
  - To optimize many algorithms

- Synchronizing large set of micro-robots: MRTP [JNCA 18] [PDP 16]
  - For synchronized actions with the external environment
    - Lighting at the same time
    - For mechanical actions

- Memory problem: CSG4PM [SAC 17]
  - Coding goal shapes

- Distributed detection of mechanically unsafe reconfiguration
  - Detecting loss of balance and breakage

- Self-assembly algorithms [AAMAS 18]

- Self-Reconfiguration algorithms
  - With map of the goal shape [NCA 16] [IEEE IoT 16] [AIM 14] [ISPA 14]
  - Without map of the goal shape [JPDC 15] [CN 15] [ROBIO 15] [JoS 14] [PDP 14] [JNCA 14] [AINA 14] [NCA 13] [SAC 13] [UIC 13] [EUROCON 13]
Problem

• Catoms network is forming a graph
  – G(V,E): V = modules, E = connections

• Is it difficult to find a central node?
  – Center: minimizes the maximum distance to all others
    \[ Center = \arg\min_{v_i \in V} \max_{v_j \in V} d(v_i, v_j) \]
  – Centroid: minimizes the average distance to all others
    \[ Centroid = \arg\min_{v_i \in V} \frac{1}{|V|} \sum_{v_j \in V} d(v_i, v_j) \]
Our contribution

- 3 distributed algorithms
  - K-BFS SumSweep
  - ABC-Center (two versions), ABC-Center-Tree (also known ABC-Center-V2)
  - Probabilistic Counter based Central Leader Election (PC2LE)

- Inspired from existing external-graph analysis algorithms
- All based on intuitive heuristics
- **Experimental evaluation of the accuracy**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type of center</th>
<th>Time</th>
<th>Memory (per module)</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-BFS SumSweep</td>
<td>center, centroid</td>
<td>$O(k \times d)$</td>
<td>$O(\Delta)$</td>
<td>$O(m \times n^2)$</td>
</tr>
<tr>
<td>ABC-CenterV2</td>
<td>center</td>
<td>$O(#steps \times d)$</td>
<td>$O(\Delta)$</td>
<td>$O(m \times n^2)$</td>
</tr>
<tr>
<td>PC2LE</td>
<td>center, centroid</td>
<td>$O(d)$</td>
<td>$O(\Delta +</td>
<td>\text{probabilistic counter}</td>
</tr>
</tbody>
</table>

Notation:

$n = \# \text{modules}, \ m = \# \text{links}, \ d = \text{diameter}, \ \Delta = \text{maximum number of neighbors}$
Outline

Software Center

- Mechanical modeling
- Self-reconfiguration
- Simulator
- Coding maps
- Synchronization
- Center
Time Synchronization

• Needed for distributed coordination

72-Blinky-Blocks scroller synchronized with our protocol (MRTP)

• Unsynchronized scroller

Start 1min20s later 20mins later

1m20s later... 20m later...
Target shape encoding

Using Constructive Solid Geometry (CSG) for describing the shape

CSG file is transferred to catoms

Each catom decides if it is in the shape or not

Efficient scene encoding for programmable matter self-reconfiguration algorithms
T Tucci, B Piranda, J Bourgeois - ACM SAC, 2017
1. `translate([17, 0, 0]) difference() {`  
2.  
3.  
4.  
5.  
6.  
7.  
8.  
9.  
10.  
11.  
12. `translate([0, 0, 7])`  
13. `cylinder(92.30, 30, true);`  
14. `};`
Outline

Software Center

Mechanical modeling

Self-Reconfiguration

Simulator

Coding maps

Center

Synchronization

Coding maps
Self-reconfiguration and self-assembly

• Self-assembly algorithms
  – With CSG map of the goal shape [AAMAS 18]

• Self-Reconfiguration algorithms
  – With map of the goal shape
    • For 2D horizontal shape [PDP 16]
    • For 2D Vertical shape [NCA 16]
  – Without map of the goal shape
    • Meta algorithm [IEEE IoT 16] [AIM 14] [ISPA 14]
    • Chain to square
      – Sequential movements [NCA 13] [SAC 13] [UIC 13] [JoS 14]
      – Parallel movements [JPDC 15] [EUROCON 13] [AINA 14] [PDP 14] [CN 15] [ROBIO 15]
    • X to square [JNCA 14]
Outline

- Mechanical modeling
- Self-Reconfiguration
- Simulator
- Coding maps
- Center
- Synchronization
- Software
Mechanical modeling

• Distributed detection of mechanically unsafe reconfiguration
  – Detecting loss of balance and breakage

Benoit Piranda, Pawel Chodkiewicz, Pawel Holobut, Julien Bourgeois, Jakub Lengiewicz, « Distributed autonomous detection of mechanically unsafe reconfiguration scenarios », in preparation
Mechanical modeling

• Detecting loss of balance

Benoit Piranda, Pawel Chodkiewicz, Pawel Holobut, Julien Bourgeois, Jakub Lengiewicz, « Distributed autonomous detection of mechanically unsafe reconfiguration scenarios », in preparation
Mechanical modeling

• Detecting breakage

Benoit Piranda, Pawel Chodkiewicz, Pawel Holobut, Julien Bourgeois, Jakub Lengiewicz, « Distributed autonomous detection of mechanically unsafe reconfiguration scenarios », in preparation
Reactive matter / Interactive sculptures / membranes experimentation

Scenocosme : Grégory Lasserre & Anaïs met den Ancxt
Reactive matter / Interactive sculptures / Lighting and sonorous feedbacks
Reactive matter / Interactive sculptures / Mashrabiya design
Reactive matter / Murmuration
Outline

Our Vision

Hardware design

Software

Art

What’s Next?

Our Vision

Hardware design

Software

Art

What’s Next?
What’s next?

- **Hardware**
  - Latching and actuation
  - Integration of the first 3D catoms
  - Scaling down the catom
  - First experiments
  - New catom design (deformation)

- **Software**
  - 3D Self-reconfiguration algorithm
  - More real test cases (gravity and forces)
  - Comparison between SR algorithms
Bibliography

Thank you for your attention!

Questions

All the source code at: http://github.com/claytronics

More information at: http://projects.femto-st.fr/programmable-matter/

All videos at:

OMNI Team (FEMTO-ST/DISC/OMNI)

« An adult scientist is a kid that never grew up », Neil DeGrasse Tyson