

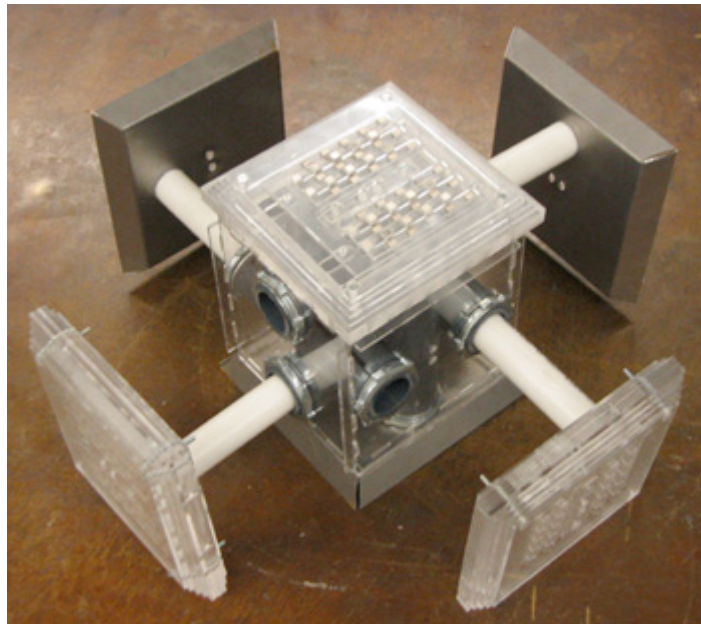
The Design and Programming of Modular Robotic Construction Kits

Mark D Gross, Ellen Yi-Luen Do, Michael Weller *
Design Machine Group
University of Washington, Seattle 98195-5720

The AFRL Space Vehicles Directorate's August 7, 2003 RFI "Space Legos" resonates with several projects currently underway in our research group at the University of Washington, Seattle. Most directly relevant is our project on reconfigurable modular robotic building blocks, described first below. Several of our other projects, described more briefly, are also germane. Beyond the mechatronic challenges of constructing reliable modular robot systems, we believe we are well positioned to research the computer-assisted design software and programming environments needed to manage self-organizing systems building blocks.

The Espresso Blocks Modular Robotic Construction System

Our Espresso Blocks project is building a working prototype of robot building blocks that can configure themselves into different physical arrangements to provide temporary shelter or habitats in extreme environments. A pallet of blocks delivered to the site would configure themselves according to spatial needs and site conditions, and could rearrange themselves as needed. (Although expensive at first, future building materials will embed these technologies.) Servomotors in each block can extend a connection in each of three dimensions and latch with neighboring blocks. Each block will also route services such as power and water. A three-dimensional "15-puzzle" scheme enables a set of blocks to configure themselves into virtually any arrangement.



Espresso Blocks Prototype-2, arms extended.

* {mdgross, ellendo, philetus} @ u.washington.edu

Supporting the design of ruleset-defined crystalline module configurations for space applications

There have been several recent proposals to use modular robotics to assemble structures in space. Yim et al[1] and Shen et al[2] suggest systems that could assemble various components in space. But these proposals fail to identify the obvious advantages of using modular robotics modules as the actual structure of the spacecraft. As Rus and Vona show[3], block-like crystalline modules are capable of taking almost any arbitrary form. Containers of crystalline modules could be launched into space, where the modules would reconfigure themselves to form a superstructure. Specialized components such as solar cells, antennas and positioning jets could also be included, and the modules of the structure itself would distribute these components to the desired positions. To upgrade or repair the system, it would only be necessary to launch new specialized components that the structure could incorporate into itself. Modules from obsolete systems could also be reused to form new systems by sending new instructions and possibly launching new components.

Several crystalline module systems are being developed. Rus and Vona's Crystalline Atom system[3] is being developed at Dartmouth and PARC is developing a system called the Telecube.[4] At the Design Machine Group we are working on our own crystalline module specifically designed as a structural building block, the Espresso Block.[5] The greatest obstacle to building structures out of crystalline modules is the development of a control system that can coordinate large numbers of modules to quickly take a desired configuration. Rus and Vona proposed a centralized control system, the "Melt-Grow Planner"[3], that demonstrates how a structure composed of crystalline modules can transition between two arbitrary configurations, but is very slow as the modules move one at a time, and does not scale well as the solution space increases exponentially with the number of blocks. However, there have been promising developments in the effort to develop distributed ruleset-based control algorithms.[6][7] In these schemes each block follows simple rules dictating the actions it should take depending on local conditions. By following local rules the blocks move in parallel to achieve a desired global configuration.

The most challenging aspect of a ruleset-based control algorithm is the design of the ruleset. To effectively use crystalline modules as a structural system, more work needs to be done on tools to support the design of module rulesets and the visualization of the configurations they will produce, and on educating designers to use these new tools.

Other related Design Machine Group projects

Our **Computationally Enhanced Construction Kits** project (funded Sept. 2003 by NSF's Information Technology Research program for five years) is designing construction kit toys (similar to Lego, Erector Set, Meccano) that are enhanced with computational capabilities. Using embedded computation, pieces within a construction kit may communicate with *each other*, with *desktop machines*, and with their *users*; and overall. The project's goal is to understand the design space of computationally enhanced construction kits and explore the educational opportunities these kits afford.

An example is our **FlexM** project, a computationally embedded hub-and-spoke construction kit for architectural design modeling and eventually for full scale building systems. In our first prototype LEDs, photo-sensors, and bend sensors in each hub determine the model's topology and geometry. These data are sent to a CAD program that displays a graphics representation of the physical model and can perform structural calculations. In our next prototype, servomotors on

each hub will actively control the structure's geometry, responding to inputs from other sensors on the model or from centrally issued commands, to fold the structure in various configurations.

An earlier project, **Avoiding Interference conflicts in Architecture Subsystem Layout** (funded by NSF's Design, Manufacture, and Industrial Innovation program from 1992-5) developed a design scheme for avoiding interference conflicts among building subsystems such as water, gas, electricity, and HVAC. Interference conflicts are a costly problem in construction as they often remain undetected until they manifest on site. Our scheme relegates each subsystem to a pre-specified spatial zone or channel. Layout and assembly constraints for each subsystem enforced by a CAD system avoids most interference conflicts, and those that remain can be solved with previously designed solutions.

The Design Machine Group

The Design Machine Group is an interdisciplinary research team in the Department of Architecture at the University of Washington, Seattle. Its mission is to develop computational systems and tools for design. The group, headed by Mark D Gross and Ellen Yi-Luen Do, works strategically with colleagues from computer science and engineering, mechanical engineering, and other departments as appropriate for specific projects. The group also has strategic alliances with industry partners, including Intel and Microsoft.

References

1. Mark Yim, Kimon Roufas, David Duff, Ying Zhang and Sam Homans. "Modular Reconfigurable Robots in Space Applications," *Autonomous Robots* 14 (2/3) March-May 2003. pp99-263.
2. Wei-Min Shen, Peter Will, Berok Khoshnevis. "Self-Assembly in Space via Self-Reconfigurable Robots," *2003 International Conference on Robotics and Automation*, Taiwan.
3. Daniela Rus and Masette Vona. "Self-reconfiguration with Compressible Unit Modules," *Autonomous Robots* 10 (1) January 2001.
4. John Suh, Sam Homans and Mark Yim. "Telecubes: Mechanical Design of a Module for Self-Reconfigurable Robotics," *2002 IEEE Intl. Conf. on Robotics and Automation*.
5. Michael Weller, "Espresso Blocks: Self-configuring building blocks," M.Arch. Thesis, University of Washington, 2003.
6. Zack Butler and Daniela Rus. "Distributed motion planning for 3D modular robots with unit-compressible modules," *Fifth International Workshop on Algorithmic Foundations of Robotics*, December 15-17, 2002, Nice, France.
7. Chris Jones and Maja J. Mataric, "From Local to Global Behavior in Intelligent Self-Assembly," *2003 IEEE International Conference on Robotics and Automation*, September 14-19, Taipei, Taiwan.