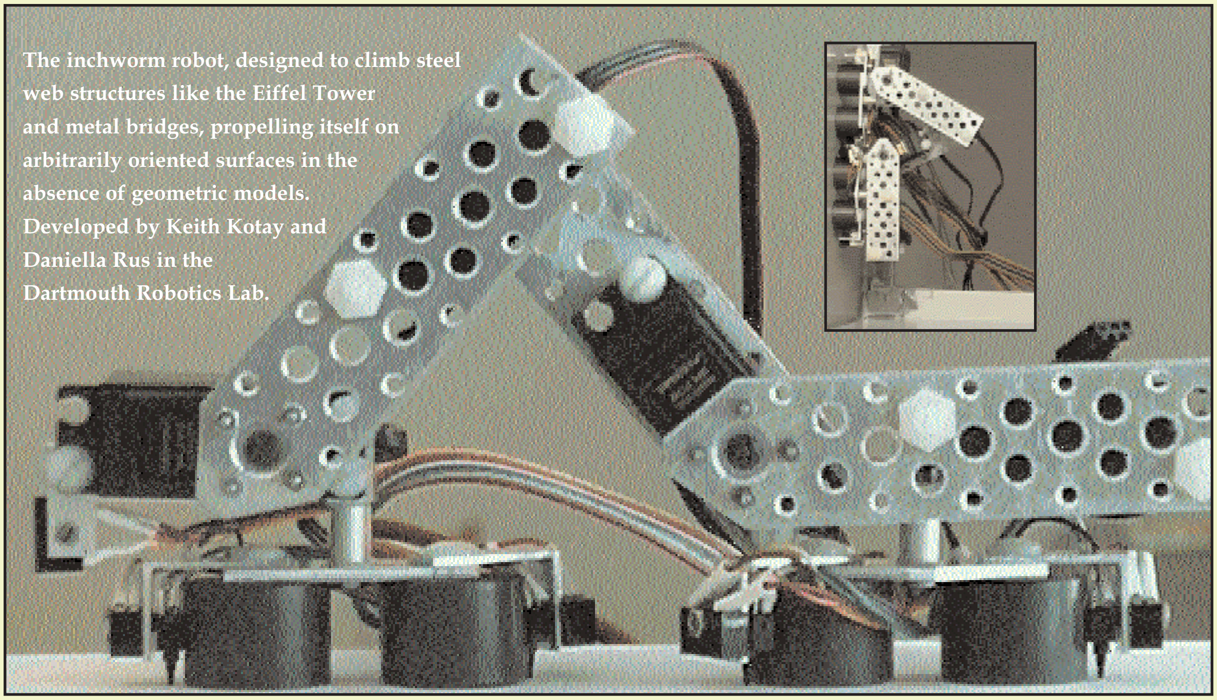
Daniela Rus, Zack Butler, Keith Kotay,
and Marsette Vona



The inchworm robot, designed to climb steel web structures like the Eiffel Tower and metal bridges, propelling itself on arbitrarily oriented surfaces in the absence of geometric models. Developed by Keith Kotay and Daniella Rus in the Dartmouth Robotics Lab.

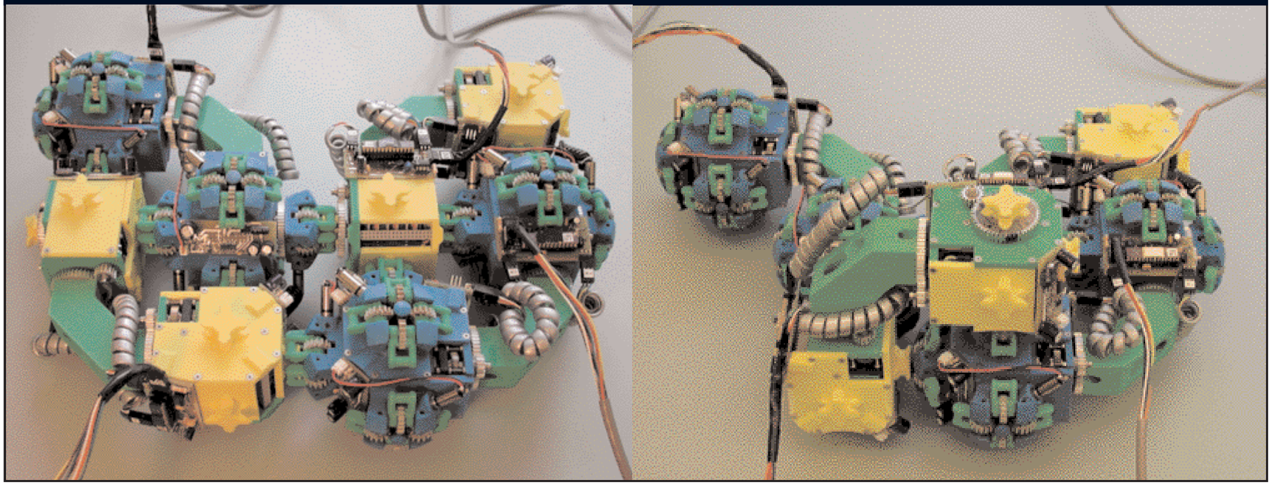
SELF-RECONFIGURING ROBOTS

Mimicking the adaptability of living biological cells, robot modules will reconfigure themselves toward a common purpose within the limits imposed by the local environment.

A robot designed for a single purpose can perform some specific task well but poorly on some other task, especially in a different environment. A fixed-architecture machine is acceptable if the environment is structured, but for tasks in unknown environments, a robot with the ability to change shape to suit the new environment and the required functionality is more likely to succeed than is its fixed-architecture counterpart.

Our vision for ultimate robotic design and functionality is to create vastly more versatile robots by building them out of simple modules with general self-reconfiguration capabilities. Hundreds of small, relatively limited, modules would be able to autonomously organize and reorganize as geometric structures to best fit the terrain on which the robot has to move, the shape of the object the robot has to manipulate, or the sensing needs for the given task. This function mimics the organizational principles of biological systems; a simple living cell does not do too much by itself, but a collection of cells can be organized to form a climbing inchworm, a crawling crab, or a running child. Large collections of small robots

Figure 1. Two different Molecule robots, each consisting of four modules, each composed of two atoms connected by a right-angle rigid bond. The connectors are implemented with electromagnets and a gripper mechanism. The Molecule has two rotational degrees of freedom about the bond and one rotational degree of freedom per atom about a single inter-Molecule connector.



will someday actively organize themselves as the most optimal geometric structure under the local environmental conditions to perform coordinated and useful work, such as repairing a damaged structure, transporting artifacts, or manipulating surface properties.

Modular reconfigurable robots have a number of other advantages over their more traditional, fixed-architecture counterparts. First, they support multiple modalities of locomotion and manipulation. For example, if a particular robot system needs to climb stairs, it would reconfigure itself so it can crawl up stairs. If it needs to cross a gap or reach a window, just-in-time bridges and towers might be created. This adaptability can be achieved by requiring the robot to metamorphose from one shape into another to best match the shape of the terrain in a statically stable gait. Second, these robots are more fault tolerant than their fixed-structure counterparts. For example, if one module fails, a modular reconfigurable robot carrying some additional modules might excise the failed part and replace it with a spare unit. Third, they can be used in tasks requiring self-assembly, such as a structure in outer space or on the floor of the deep ocean. Fourth, they can provide a means for physically modeling 3D data—a real breakthrough in visualization. It is now common to use software to create computer models for 3D data. We envision creating physical models that can be manipulated directly by using modular self-reconfiguring robots. The robot would morph itself into the geometric shape dictated by the data. The user would be able to touch and manipulate the physical model provided by the robot. Ultimately, when the technology is available at a very small scale (even down to nano-scale), ordinary objects, say, a lamppost, would be able to reorganize themselves on demand into other ordinary objects, say, a bench or barricade.

Self-reconfiguration represents a paradigm shift for studying the fundamental principles of organization and reorganization in physical systems. In robotics, self-reconfiguration defines a rich class of questions about designing, controlling, and using massively distributed systems of robots. Self-reconfiguration also offers fertile ground for applying existing concepts in novel ways. For example, the domain involves the need for two kinds of planning algorithm: one to achieve a desired geometric shape and one to globally move the resulting shape in any physical direction. To explore the value of these algorithms and their applications, the following sections cover some of the hardware and algorithmic challenges of achieving task versatility with self-reconfiguring robots.

Hardware Systems

Self-reconfiguring robot systems employ physical mechanisms allowing modules to dynamically and automatically configure and reconfigure themselves into more complex forms. Creating self-reconfiguring robot systems poses many engineering challenges centered on designing the basic self-reconfiguring module and inter-module connections, as well as on how to aggregate distributed systems from the modules.

We want the basic module to be as small and simple as possible in terms of physical size and numbers of components, linkages, and functions, because the smaller the module, the greater the range of shapes that can be built from it. The modules should also be able to function independently of one another. Although simple modules are easier to design and build, simplicity may constrain functionality. Simplicity is also a key consideration in designing the intermodule connection mechanism. Because the modules make and break connections frequently, the connections should be simple and reliably independent of human intervention.

Other important design issues include communication between modules, actuator power, and the method used to supply electrical power to the system. Inter-module communication is necessary for module cooperation and distributed control of the various components in a system. A good connection mechanism can also be used to transmit messages between modules. Actuator power is the amount of force actuators have to exert to move the modules around; minimally, modules must be able to move their own weight. In a 3D system, modules must be able to move their own weight against the force of gravity. But supplying electrical power to modules is difficult and costly in terms of design and fabrication. Moreover, if the modules supply their own power using batteries, their weight and size increase, thus requiring more power to move them around. One possibility is to use the connection mechanism to simultaneously transmit power to all modules.

Several research groups have developed solutions to designing self-reconfiguring robots. For example, Toshio Fukuda et al. of Nagoya University first proposed in 1990 the idea of a cellular robotic system in which a set of specialized modules called "cells" are coordinated to accomplish a complex task [1]. The first hardware systems were developed a few years later at the Mechanical Engineering Laboratory (MEL) in Japan, as well as at Johns Hopkins University and Xerox PARC; the sophistication of the devices has increased ever since. Meanwhile, several groups have contributed groundbreaking ideas, such as the expansion/contraction actuation mechanism of [6], the connection mechanism of [3], and the deformation-based actuation of [4]. A good overview of the state of the field is presented in [5].

Most proposed unit-reconfigurable robot systems are actuated by rotating a module relative to the rest of the robot or expanding and/or contracting a module; their connection mechanisms are magnetic or mechanical. Modular robots are characterized as either homogeneous (all modules are identical) or heterogeneous (the modules are different). For example, Mark Yim of Xerox PARC defined a reconfiguring system to be "unit modular" if it is homogeneous. Most existing self-reconfiguring robot systems are based on the unit-modular approach.

Our own work has focused on the principle of mechanical simplicity, or the simplest design with the fewest components to accomplish the job. We've also characterized the properties that would confer a unit-modular robot system with self-reconfiguration [6]. Guided by these results, we've developed two unit-modular systems: the Molecule robot and the Crystal robot. The main goal of the former is self-reconfiguration in 3D. The latter uses a novel actuation mechanism, called scaling, that allows an individual module to double in size by expanding or halve its size by contracting, thus providing more robust motion than the previous rotation-based actuation systems. Instead of moving individual modules by rotating them, we change the overall robot shape by expanding and contracting the modules.

The Molecule robot. A Molecule robot consists of multiple units called Molecules, each consisting of two atoms the size of an apple linked by a rigid connection called a bond [2] (see Figure 1). Each atom has five inter-Molecule connection points and two degrees of freedom. One degree of freedom allows the atom to rotate 180 degrees relative to its bond connection; the other allows the atom (thus the entire Molecule) to rotate 180 degrees relative to one of the inter-Molecule connectors at a right angle to the bond connection. The current design uses R/C servomotors for the rotational degrees of freedom. A feature of the prototype is the use of a gripper-type connection mechanism for linking the various individual molecules.

> Ordinary objects, say, a lamppost, would be able to reorganize themselves on demand into other ordinary objects, say, a bench or barricade.

The rotating connection points on each atom are the only ones required for Molecule motion. The other connection points are used for attachment to other Molecules in order to create stable 3D structures. Each Molecule also contains a microprocessor and the circuitry needed to control the servomotors and connectors. Each atom is four inches in diameter, and the Molecule weighs three pounds. Each individual Molecule has three basic motion capabilities: linear motion in a plane on top of a lattice of identical Molecules, irrespective of the absolute orientation of the plane; convex 90-degree transitions between two planar surfaces composed of Molecules; and concave 90-degree transitions between two planar surfaces composed of Molecules.

These primitive motions for a Molecule relative to a substrate can be combined and sequenced by the robot control system to achieve global motions for the entire robot. We've found that a four-Molecule robot is the smallest one that can move in general ways in the plane [2]; the smallest one that can also climb stairs is an eight-Molecule robot. Figure 1 shows our prototype four-Molecule robot.

The Crystalline atom. The Crystal is our novel self-reconfiguring module, a mechanism with some of the same motive properties as biological muscles that can be closely packed in 3D space and that can attach themselves to similar units (see Figure 2). Each of the 24 Crystal modules we've developed so far is actuated by expansion and contraction of its four faces. By expanding and contracting the neighbors in a connected structure, an individual module can be moved in general ways relative to the entire structure. Crystal atoms never rotate relative to each other; their relative movement is actuated by sliding via expansion/contraction. This basic operation has yielded new algorithms for global self-reconfiguration planning [6]. When fully contracted, each atom is a square measuring two inches on each of its sides. When fully expanded, each atom is a square with a four-inch side. Each atom weighs 12 ounces. Crystal robot systems can realize a wide range of geometries; for example, Figure 2 (top) shows a dog-shaped object transforming itself into a couch-shaped object.

Crystalline robot systems are dynamic structures. They move using sequences of reconfigurations to implement locomotion gaits and undergo shape metamorphosis. The dynamic nature of these systems is supported by the ability of individual modules to move globally relative to the entire structure. Unlike all previously proposed unit modules, in which modules are relocated only by traveling on the surface of a structure, Crystalline atoms can be relocated by traveling throughout the volume of a Crystal. Instead of propagating the module along the surface of the robot structure (requiring a linear number of expansion and contraction operations in the modules on the surface of the cube), the same goal can be achieved using a constant number of internal expansion and compression operations [6].

Algorithmic Issues

The algorithmic challenges involved in achieving self-reconfiguring robotic systems in a distributed fashion concern the metamorphosis of a given structure into a desired structure and how to use self-reconfiguration to implement multiple (adaptive) locomotion and manipulation gaits. These issues can be formulated as motion-planning problems. The key observation for automated planning is that most self-reconfiguring systems consist of identical modules. Since all modules are also interchangeable, it is not necessary to compute goal locations for each element. Thus, self-reconfiguration is different from the related intractable warehouse problem in which modules are assigned unique IDs and have to be placed at desired locations. Several groups have proposed architecture-dependent planners [2–4, 6, 8, 9]. This work can be divided into two categories of approaches: centralized and decentralized. The former is easier to analyze for performance guarantees but is not scalable for large robots. The latter supports parallelism but is generally more difficult to analyze.

Most approaches to planning have two parts: a set of device-level primitives for controlling the motion of one module relative to a structure or substrate; and general planning algorithms built by "composing" these primitives. Two of our centralized planning approaches are described in detail in [2, 6]; other groups [3, 4, 7–9] have pursued similar approaches.

Some of the most interesting future applications for self-reconfiguring robots promise to employ thousands of modules working together. Such systems represent ultra-high-degree-of-freedom systems that might be able to synthesize a robotic pet or a couch at one's request. However, centralized planning algorithms move one module at a time and may be too slow and impractical for controlling robots made of thousands of modules. For this reason, it is important to consider distributed planners that are scalable, support parallelism, and are better suited for operation in unstructured environments.

Distributed planning for Crystal robots. One possible approach is an algorithm called PACMAN distributed control we've developed for unit-compressible systems like our Crystal robots. An overall desired shape, or locomotion gait, is given to the robot's modules, each of which then determines whether or not it needs to move using only local information. If motion is necessary, each module initiates a path search through its fellow modules using only local information at each step. After a path is found, it is instantiated by marking each atom through which the path travels. We therefore developed data structures called "pellets" as a way to mark the path a module should follow to perform its part of the reconfiguration. This process could be viewed as the engineering equivalent of Hansel and Gretel's bread crumbs left through the forest, but since the pellets are permanent, the result is better. Because of the Crystal's unique actuation principle, a single physical module does not follow the entire path. Rather, it exchanges identities with other modules along the path, so it appears to follow the entire path while actually moving only locally. Additionally, by marking each pellet with the identity of the

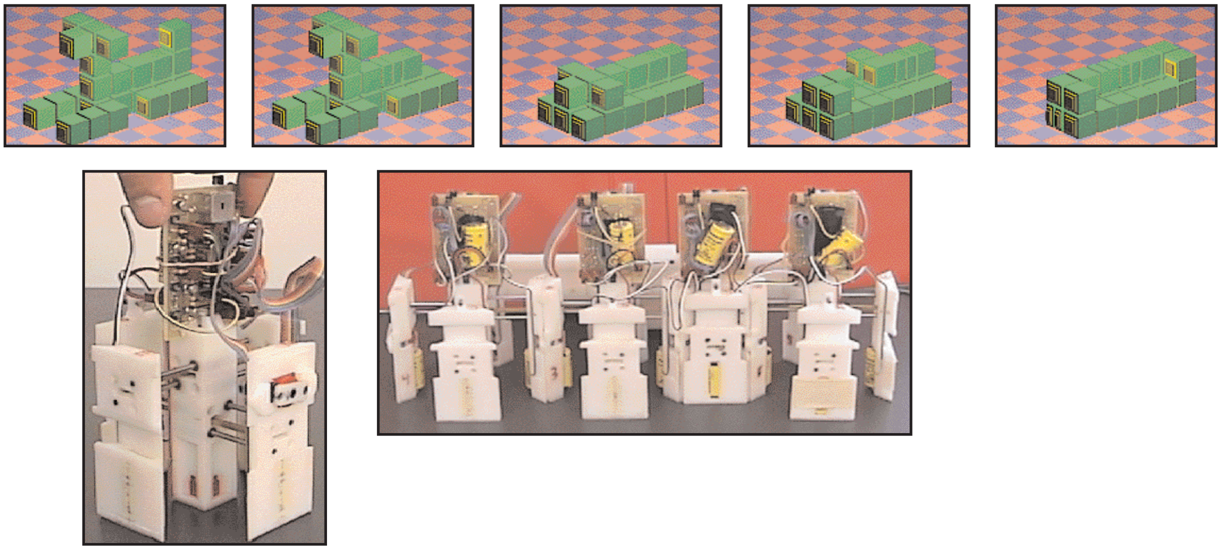


Figure 2. (Top) Five snapshots from a simulation using Crystalline robots showing the intermediate steps in the transformation from dog-shaped object to a couch-shaped object; (bottom left) the physical prototype of the Crystalline atom; and (bottom right) a robot consisting of four Crystals.

module that is to eat it, paths for several modules can coexist in the Crystal robot. This navigation and configuration strategy in turn allows modules to perform simultaneous reconfigurations without relying on a central clock or the actuation of only a single path at a time.

The reconfiguration process involves two main steps. First, a path is planned for each module in a distributed fashion. The result is a set of pellets distributed through the atoms of the robot. Once the pellets are in place, the actuation happens asynchronously, as each atom looks for pellets and "eats" them without adhering to a strict schedule. This strategy means that although the intermediate structure of the crystal is undetermined, the final structure is as specified.

This actuation protocol can direct the active module to move and trade identities with other modules along the path, eventually resulting in a module appearing at a location specified in the goal statement. It provably allows for many paths to be planned and executed simultaneously through the robot, since each active module needs to look only in its immediate neighborhood to discover and actuate the next step on its path.

General approaches to decentralized planning. The current direction in self-reconfiguring robotics focuses on designing and building hardware and developing algorithms coupled to specific hardware. We are at a point where we can step back to examine more general questions about self-reconfiguration planning in an architecture-independent way. It is important to examine architecture-independent algorithms that can be instantiated for many different systems because they have the potential of providing a more general science base for the field. By outlining general principles for reconfiguration planning, we hope to learn how to better design hardware and control algorithms.

The ability of self-reconfiguring systems to change shape can lead to water-flow-like locomotion algorithms allowing the robot to conform to the terrain on which it has to travel. These algorithms have the potential for working well in unstructured environments. In most existing self-reconfiguring robot systems [2, 3, 6, 9], an individual module can move in general ways relative to a structure of modules by traveling on the surface of the structure. Specifically, an individual module is capable of:

- Linear motion on a plane of modules;
- Convex transitions into a different plane; and
- Concave transitions into a different plane.

The details for how to accomplish these goals are architecture-dependent. We can use these motion abstractions as the basis for a general decentralized cellular-automata algorithm that can work for any system capable of such motions. The cellular automata control uses the primitives within local rules to model water-flow-like locomotion.

We have developed several sets of simple rules for cellular automata that produce reliable and provable locomotion; Figure 3 shows snapshots from a simulation of a blob-like robot moving over irregular obstacles using this approach. Each rule requires a set of

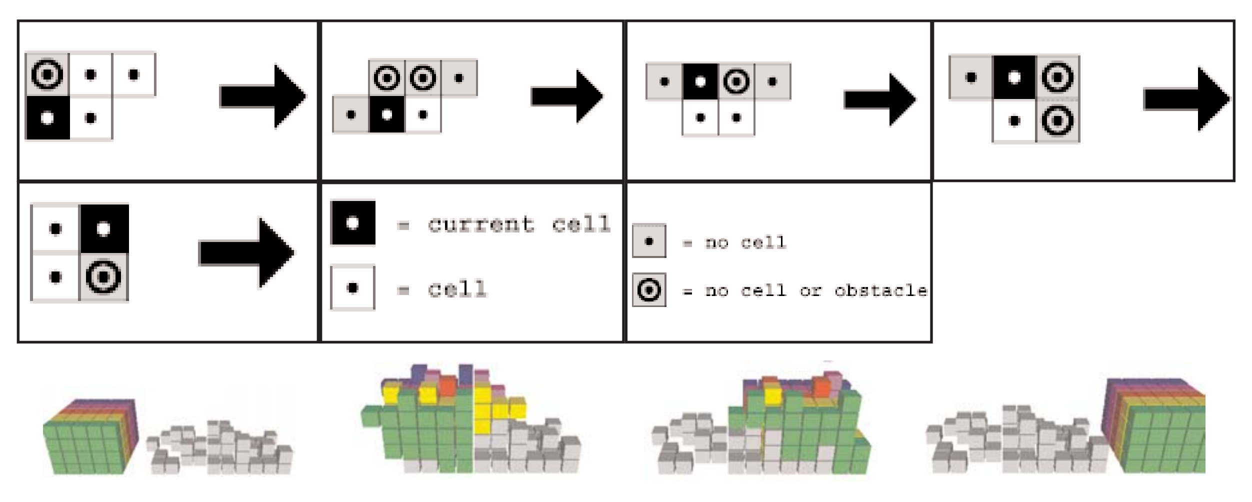


Figure 3. (Top) Five rules for rightward locomotion without obstacles. The robot modules are represented as squares; a variety of symbols characterize the local configuration of the active cell, which is represented as a black square with a dot in the middle. The first rule allows, for example, the active module to move up by one unit when its top is free and the surrounding structure consists of at least three units arranged in an L shape. (Bottom) Screenshots of a compliant locomotion simulation using cellular automata rules; the light gray squares model obstacles, or terrain irregularities.

preconditions on the neighborhood of the cell that, when activated, causes the cell to move to an adjacent location on the surface of the system. These correct abstract algorithms can be instantiated on a variety of robot systems with their correctness properties intact.[1]

In addition to the locomotion task, we have investigated other reconfiguration tasks, including: self-replicating robots (dividing a large modular system into a collection of smaller systems, an operation useful for distributed search and rescue applications); and self-merging robots (regrouping two smaller robots into one robot). These tasks are interesting by themselves and useful for applications involving distributed monitoring and surveillance. Moreover, insights into how to approach these problems will improve our understanding of more general problems, such as how the number of modules in a robot affects the robot's task repertoire and whether a large number of small simple modules is more powerful than a single complex robot.

Conclusion

Self-reconfiguring robots are able to adapt to the operating environment and required functionality by changing shape. They consist of a set of identical robotic modules that can autonomously and dynamically change their aggregate geometric structure to suit different locomotion, manipulation, and sensing tasks. However, creating robots with self-reconfiguration capabilities is a serious challenge now being met through new designs for reconfigurable systems and new ideas about algorithmic planning and control that confer autonomous reconfigurability. We've discussed hardware design issues and presented two solutions developed in our laboratory. We also discussed planning issues and illustrated a hardware-specific distributed planner and a generic distributed planner that can be instantiated to many different designs.

These results are encouraging first steps toward creating self-reconfiguring robotics applications. However, we have a way to go before we can engineer modular self-reconfiguring robot systems that can be embedded into the physical world and respond in real time to requests for self-assembly. Because these robot systems will constitute long-lived distributed systems, all the supporting hardware and software will have to be robust, long-lasting, fault-tolerant, scalable, and self-healing. The hardware will have to rely on simple and robust actuation. The units will have to be powered for long periods of time. Adding and removing units into the system will have to be incremental, in that these changes should affect the overall system only locally. When units break, the system should be able to repair itself without altering overall global functionality. The units will have to be networked with a reliable wireless ad-hoc communication infrastructure. And control will have to be highly parallel, scalable, and distributed.

---

[1] We instantiated them to the Molecule robot [2], the MEL Fracta robot, and the robot in [3]. The proof is developed from three statements: At any point in time, at least one of the five rules in Figure 3 applies to one cell in the system; any sequence of rules produces a net rightward motion; and no sequence of rules causes the system to disconnect its components from one another. The correctness result for decentralized locomotion is encouraging, as it is generally difficult to prove the correctness of distributed algorithms specified in a bottom-up fashion.

To develop such systems, we have to improve our understanding of the general properties that confer modular robots with self-reconfiguration capabilities, as well as more generic (rather than architecture-specific) solutions to control and planning. Programming and giving commands to these systems should be at least as easy as writing a HTML page. These issues are being addressed by the research community, motivated by the exciting vision of versatile robots achieving the same level of flexibility as biological systems of cells.

REFERENCES

1. Fukuda, T. and Kawauchi, Y. Cellular robotic system (CEBOT) as one realization of a self-organizing intelligent universal manipulator. In *Proceedings of the 1990 IEEE Conference on Robotics and Automation* (Cincinnati, May 13–18). IEEE Computer Society Press, 1990, 662–667.
2. Kotay, K. and Rus, D. Locomotion versatility through self-reconfiguration. *Robo. Autonom. Syst. 26* (1999), 217–232.
3. Murata, S., Yoshida, E., Kurokawa, H., Tomita, K., and Kokaji, S. Self-repairing mechanical systems. *Autonom. Robo. 10,* 1 (Jan. 2001), 7–21.
4. Pamecha, A., Ebert-Uphoff, I., and Chirikjian, G. Useful metrics for modular robot motion planning. *IEEE Transact. Robo. Autom. 13,* 4 (Aug. 1997), 531–545.
5. Rus, D. and Chirikjian, G., Eds. Special Issue on Self-Reconfiguring Robots. *Autonom. Robo. 10,* 1 (Jan. 2001), 1–124.
6. Rus, D. and Vona, M. Crystalline robots: Self-reconfiguration with unit-compressible modules. *Autonom. Robo. 10,* 1 (Jan. 2001), 107–124.
7. Shen, W.-M., Will, P., and Castano, A. Robot modularity for self-reconfiguration. In *Proceedings of the International Society for Optical Engineers (SPIE) Conference on Sensor Fusion and Decentralized Control in Robotic Systems* (Boston 1999), 236–245.
8. Unsal, C. and Khosla, P. A multi-layered planner for self-reconfiguration of a uniform group of I-cube modules. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Maui, HI). IEEE Press, Piscataway, NJ, 2001, 598–606.
9. Yim, M., Zhang, Y., Lamping, J., and Mao, E. Distributed control for 3D shape metamorphosis. *Autonom. Robo. 10,* 1 (Jan. 2001), 41–56.

DANIELA RUS (rus@cs.dartmouth.edu) is an associate professor in the Computer Science Department of Dartmouth College, Hanover, NH.
ZACK BUTLER (zackb@cs.dartmouth.edu) is a research associate in the Computer Science Department of Dartmouth College, Hanover, NH.
KEITH KOTAY (kotay@cs.dartmouth.edu) is a Ph.D. candidate in the Computer Science Department of Dartmouth College, Hanover, NH.
MARSETTE VONA (vona@helios.jpl.nasa.gov) is a member of the technical staff at NASA's Jet Propulsion Laboratory.

This work was done in the Dartmouth Robotics Lab. Support was provided through a U.S. National Science Foundation CAREER award IRI-9624286 and NSF awards IRI-9714332, EIA-9901589, IIS-9818299, and IIS-9912193, the Sloan foundation, and a NASA Spacegrant award.
