Trans. of the Society of Instrument and Control Engineers
Vol.E-1, No.1, 1/9

Self-Configurable Machine
~Self-assembly of distributed unit structured mechanical system~

Satoshi Murata*, Haruhisa Kurokawa**, Shigeru Kokaji**

We describe a novel methodology for the design of machines which is analogous to some structures found in nature. In conventional machines, design, production, and maintenance functions are usually performed by agencies outside the machine itself, and this is different to many natural systems which have intrinsic mechanisms enabling evolution, proliferation, and self-repair. Artificial systems are becoming increasingly complicated, and are often expected to operate unattended for long periods of time in difficult environments. There is therefore considerable interest in applying ideas developed from observation of biological systems in order to make artificial machines more flexible and robust. In this paper, we propose a novel type of machine which is made from identical interlocking units. Interconnections between units may be changed, and each unit is equipped with the same control software which enables a group of them to move and form shapes, and repair those shapes should a fault occur, automatically without outside intervention. Prototype units have been produced and basic operation of the system demonstrated.

1. Introduction

In the history of development of the machine, there are numerous examples where man has developed devices to imitate (usually with the hope of improving upon in some way) the functions of living beings. The development of the airplane is a prime example where this philosophy has worked well. Undoubtedly originally inspired by the flight of birds, the development of the airplane has brought us machines which are capable of carrying large numbers of people over vast distances at great speed performance which in many ways is far in excess of that of the largest bird.

This example is typical not only of machines, but of many of the artificial systems that man has made. As these systems have become more sophisticated, however, it is apparent that the methods used to design and produce them have become more and more divorced from the examples in the natural world that were their original inspiration. There is a definite, and growing, conceptual gap between the natural world and man-made devices, and we believe that this may be partly responsible for many of the problems which are now appearing as the complexity of artificial systems increases and more is expected from them.

There are three broad areas in which difficulties are becoming apparent. These are:

1) Increasing complexity of design and manufacturing methods.
2) High maintenance costs.
3) Increasing tendency for catastrophic effects arising from system failure.

Most efforts to overcome these difficulties follow traditional methodologies for the design of artificial systems, but paradoxically it often seems that their net effect is to exacerbate rather than relieve the overall problem. In this paper we discuss a different philosophy which we believe has potential for solution of the general problem, that is, a solution which goes some way to filling the conceptual gap between artificial products and the natural world.

One of the biggest overall differences between artificial and living systems is the level of autonomy that each has. This becomes clear from even a cursory comparison of artificial and natural processes for

* Tokyo Institute of Technology, Nagatsuda, Midoriku, Yokohama
** National Institute of Advanced Industrial Science and Technology, Namiki, Tsukuba

system design, production, and maintenance. For example, artificial systems are designed by engineers (usually with the aid of computing facilities), production usually occurs in large factories, and maintenance is performed by human experts. Although many of these processes are now automated, the basic idea of automation is to replace human operators with mechanisms that mimic the operations performed by those operators, with perhaps more power. It is nonetheless true that the processes of design, production, and maintenance are essentially done by agencies which are external to the artificial system itself.

On the other hand, complex living systems are produced in the natural world without design staff or production facilities and they also repair and maintain themselves largely without the help of external agencies. These functions are embedded in the living systems themselves and so they have a high degree of autonomy. Although conventional wisdom is that it may be redundant or inefficient to incorporate these functions into artificial systems, we believe that it is worthwhile revisiting this concept. The success of complex living systems at reproducing and maintaining themselves seems to be a clear indication that we should investigate the possibility of developing artificial systems based on the principles of the natural world, which have survived over a long history and which are radically different to those commonly used today.

This principle has been applied in the field of information processing, with the development of cellular automata for powerful and flexible computation tasks 1,2,3). However, to the best of our knowledge there are few mechanical devices which have cellular characteristics analogous to biological systems although proposals have appeared in the literature 4). The development of the micro-chip, and its application in information processing and robotics, have now provided the means for practical development of these ideas and their application to mechanical systems, and there is increasing interest in the field 5-9). At the same time, progress is being made in the development of control theory for autonomous distributed systems and the use of cellular systems to produce flexible structures is already feasible in some fields, for example front-line astronomical research 10,11).

2. Design of Mechanical System

With this in mind we have developed a self-configurable mechanical system which is composed of separate units which are in a sense mechanical analogues of the cells found in living organisms. The units are identical, can interlock with each other, and move on a plane to change their relative location in two dimensions. Each unit is equipped with actuators, an information processing device, and facilities for communication with adjacent units. We have also designed software for system control. Once again, all units have identical software under the control of which the units move within their group to form a given shape.

This structure is very similar to that found in living things. When the software is operating, a group of enough units placed randomly on a plane will move, each unit communicating with its neighbors as they do so, until the group as a whole reaches the programmed shape. The group recognizes when the shape has been achieved, and works to maintain that shape if disturbed by external influences. For example, if one of the units is forced out of position, the group moves again to restore the required shape and we anticipate that further development will lead to systems with self-repairing capability similar to that found in the natural world. The units work together as a highly autonomous system, with processing going on in a naturally distributed way. The key characteristics of our system are as follows:

1) The system is composed of identical units.
2) The units can connect mechanically with each other, and connections can be changed as the units work together to achieve a desired end.
3) Each unit is equipped with identical control logic.

The units each communicate with their neighbors, assess the current situation of the group, and make decisions regarding movement in an autonomous fashion, without control from any external supervisory agency.

We describe the prototype units and their software algorithm below, and give the results of computer simulations which demonstrate their capability for self-configuration.

3. System hardware

Fig. 1 is a schematic of one of the units in our system. The mechanical links between units are electromagnetic, and the units move over a horizontal plane on ball castors to keep friction low. Each unit is made from three horizontal layers, each with three lobes. Each of the outer two layers contains three permanent magnets and the middle layer (which is staggered from the outer two by 60 degrees) holds three solenoids, one on each lobe. The solenoid on each lobe of the middle layer is attracted into, or repelled from, the gap between outer layer lobes on neighboring units, dependant on the direction of current flow (**Fig. 2**). Each unit can connect with a maximum of six neighbors.

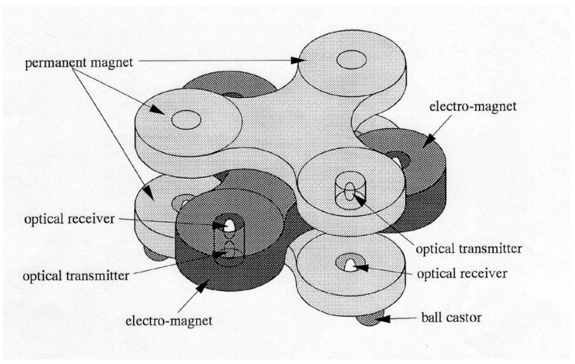

Fig. 1 Schematic of the unit

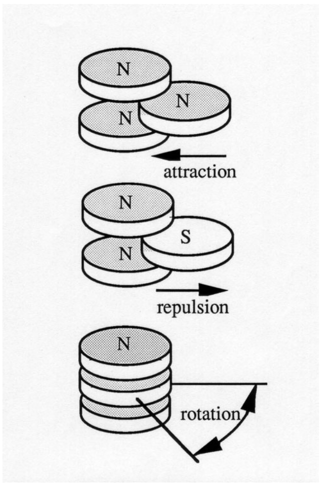

Fig. 2 Principle of action

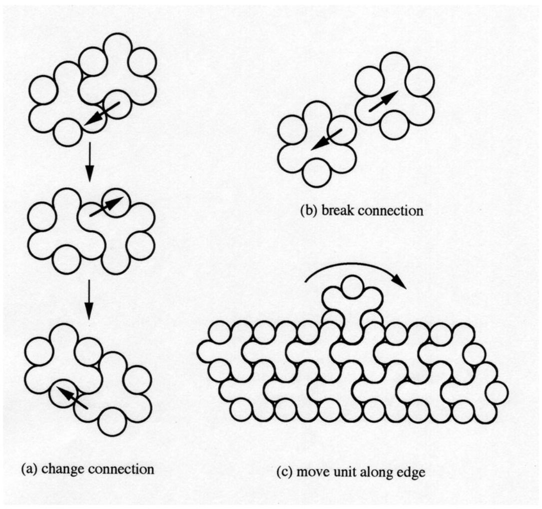

Fig. 3 Basic functions of units

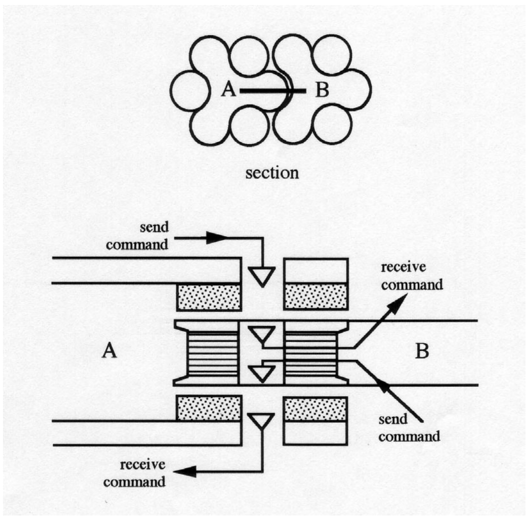

Fig. 4 Devise for communication

Fig. 3 shows how appropriate sequencing and control of the solenoid currents may change connections between adjacent units. For example, the position of the connection between two units may be changed as in (a), or the connection between two units may be broken (b), or a single unit may be moved along the edge of an assembly of units by cycling its solenoid currents appropriately (c).

An onboard microprocessor is installed in each unit to realize autonomous operation. Information may be exchanged optically between coupled units by means of LEDs and phototransistors located in the lobe centers (**Fig. 4**). **Fig. 5** shows an experiment in which

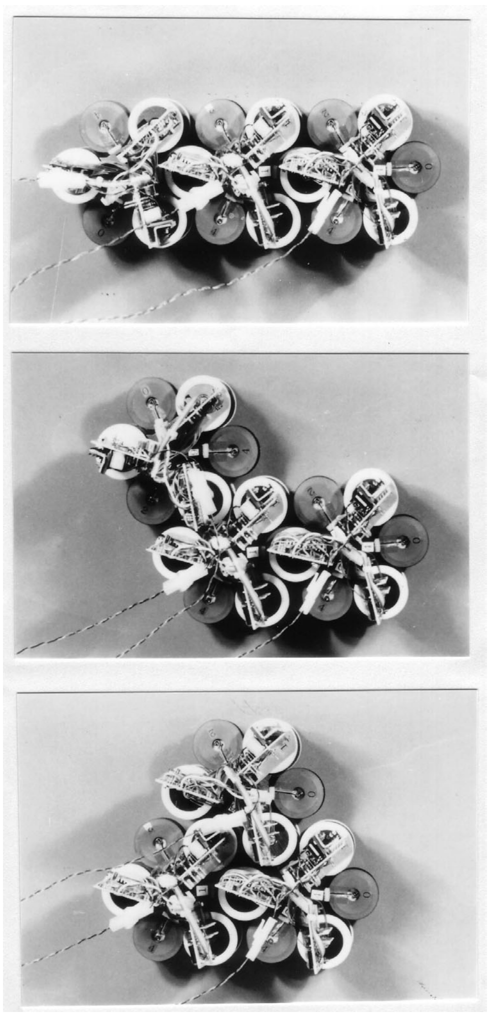

Fig. 5 Experiment

the connection between units was changed as a result of communication between them. The wires above the units provide electrical power only.

4. Software design constraints

4.1 Design Constraints

The software running in the processor controlling each unit is the key to producing a successful machine capable of autonomously configuring itself. Our aim was to produce software which would control the units so that they changed their interconnections while communicating with each other, in order that an initially arbitrary arrangement of units would converge to a predetermined global configuration (**Fig. 6**). In order to make a truly autonomous system with distributed processing in the broadest sense, the software was developed while adhering strictly to two important constraints.

The first of these is that the software in all units must be identical. This guarantees that the units are completely interchangeable which is very important during recovery from system failures. It is only necessary to have one type of replacement unit on hand, and interchangeability considerably simplifies system recovery. Of course, the actual information processed by each unit at any time will depend on its position in the group. In a sense, this situation is analogous to that found in biological systems where each cell has the same basic information stored as a genetic sequence, and the cells differentiate from a homogenous state dependant on overall system requirements.

The second constraint is that communication is only allowed between adjacent units. To a certain extent this is inherent in the structure of the units. It greatly simplifies the hardware architecture and makes the system much more practically feasible. It also has parallels in biological systems where information exchange between cells occurs between physically connected cells only and therefore must be local.

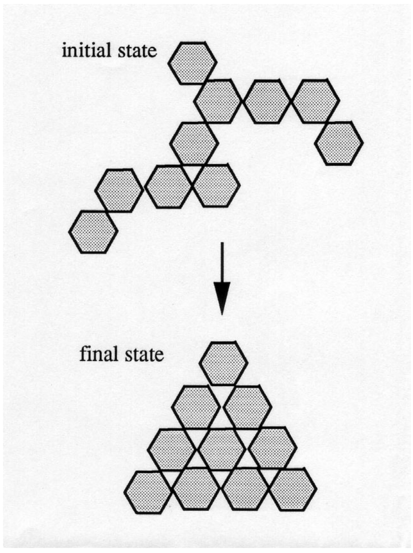

Fig. 6 Self-assembly

4.2 Description of the global configuration

With the above constraints, each unit only directly knows its own connection type, that is, which of its connecting lobes is occupied. Through communication

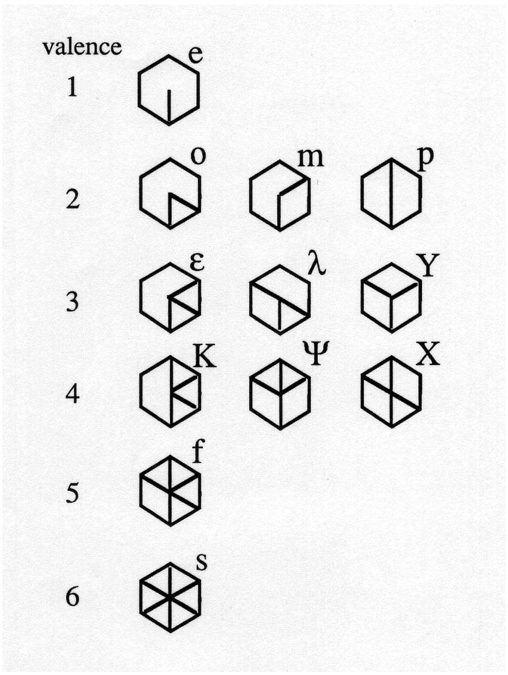

Fig. 7 Connection types

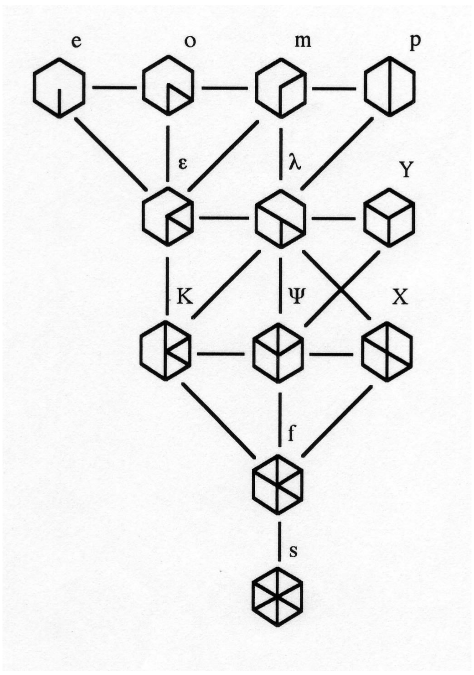

Fig. 8 Type transition diagram

with its neighbors, it can also gain information about their connection type. We define the "state" of a unit using a set containing its own connection type and its neighbors' connection types. We then developed formalism for describing the global configuration of the system in terms of the connection states of the units.

We start by classifying the connection type of any unit into one of 12 connection types as shown in **Fig. 7**. Here, the unit is represented by a hexagon, and the short bars inside it are occupied connecting lobes. In principle, two units may connect to each other with either only one lobe on each occupied (single bond) or with two adjacent lobes on each occupied (double bond). In practice, double bonds arise only temporarily during movement of the units and so we only consider single bonds in describing the system configuration. We introduce a hierarchy of connection types as follows:

$$e < o < m < p < \varepsilon < \lambda < Y < K < \Psi < X < f < s \qquad (1)$$

which will be used later to compare connection states in the system.

Fig. 8 is a diagram representing the distance between connection types. A link between types indicates that one type may be changed to the other by a single step movement of the unit. The distance between two connection types is defined as the number of links on the shortest path between them. For example, the distance between types "e" and "f" is 3. This definition of distance roughly reflects the energy required to transit from one type to another. It should be noted here that not only does the connection type of the moving unit change, but the connection types of its neighbors are also affected.

The global configuration of the system is expressed in terms of connection state, taking into account the hierarchy described in (1). For example, assume that 10 units are connected together to form a triangle as shown in **Fig. 9**. This shape is described as follows:

$$\begin{array}{l} o(K,K) \\ K(o,K,K,s) \\ s(K,K,K,K,K,K) \end{array} \qquad (2)$$

The first statement, o(K,K) means that the type "o" units at the corners of the triangle each connect with two type "K" units on the sides. The second statement, K(o,K,K,s) means that the type "K" units on the sides each have four neighbors of types "o", "K", "K", and

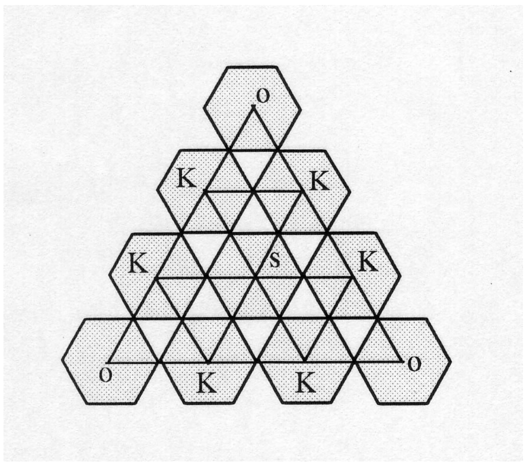

Fig. 9 Triangle configuration

"s". Finally, the last statement means that the unit at the center of the triangle has type "s" and is surrounded by 6 type "K" units. Note that the types in parentheses are sorted according to the hierarchy in (1). This set of three statements is known as the "description" of the global configuration of the system.

The length of the description depends on the complexity of the configuration. Any configuration gives rise to a single description, but the reverse is not always true because two or more configurations can be built from some descriptions. However, for a small system with high symmetry as in this example, the description is compact and uniquely determines the configuration.

5. Method for self-configuration

It is possible to imagine several algorithms which would give the system the capability to configure itself to form a target shape from an arbitrary starting point. We adopted perhaps the simplest of these which works as follows:

We assume that a description of the target shape is given to all units before configuration starts. Evolution of the system then proceeds in discrete time steps at each of which every unit compares its own current connection state with the description of the final system state. It then evaluates its current "comfortableness", which is characterized by a parameter we call the "difference index" that is a rough estimate of the distance between its current connection state and the closest connection state in the target system description. If the unit judges its "comfortableness" to be good enough it does not move in that time step. However, if the "comfortableness" is judged inadequate, the unit moves randomly in an attempt to find a better position.

This strategy seems rather simple, but experience indicates that it almost always leads to successful convergence to the required configuration. It resembles the process which occurs when sand is packed into a cast. Under vibration (analogous to the random movement in our case), the grains of sand (corresponding to our units) gradually reach every corner of the cast.

We describe calculation of the difference index with reference to the following example. Assume that the current connection state of a unit is "e(K,Y,s)" and the target configuration of the system is that described by (2). The distance between this state and the first statement in the description is evaluated by:

$$\begin{aligned}&\text{distance}[\ \varepsilon(K,\Psi,s),\quad o(K,K)]\\&\quad= \text{distance}[\varepsilon(K,\Psi,s),\quad o(-,K,K)]\\&\quad= \text{distance}[\ \varepsilon, o] + \text{distance}[\ K,-] + \text{distance}[\ \Psi, K]\\&\qquad + \text{distance}[\ s, K] = 1+2+1+2 = 6 \quad (3)\end{aligned}$$

In this evaluation we take into account the differences in sequence lengths in the parentheses by inserting "-" in the parentheses of the shorter statements to make the lengths uniform. The symbol "-" is taken as connection type "e", for convenience.

We calculate the distances to other statements in the description in the same way. In this example the distance is 3 for the second statement and 12 for the third statement. The difference index is the smallest of these distances, i.e. in this case it is 3. If the difference indices of all units are zero, we assume that the system has reached its target configuration.

Although the difference indices indicate which units should be moved in attempting to reach the required system configuration, they give no information as to the practicality of moving any given unit. To take an extreme example, a unit which has current connection type "s" cannot move because all its lobes are occupied, and even units with some free lobes may not be able to move because they may not have enough torque or because the system may become disconnected if they do. We therefore define a movable unit as one which:

1) can rotate without carrying other units
and
2) leaves the whole system connected after the movement.

According to these criteria, we define units with connection types "e", "o", and "e" as movable. Units which do not meet these criteria are not allowed to move no matter how large their difference indices may be.

At first sight, it may seem sufficient to simply set a fixed threshold for difference index, and then use the definition of movability given above in order to determine whether a unit should be moved at any given time step. However, the difference indices become small as the system approaches its final shape and it is difficult to choose a threshold which is appropriate for the initial stages of development of the system and yet still allows it to converge precisely to the required configuration. Some kind of variable thresholding system is required.

In order to determine accurately which units to move at each time step, we use a simulated diffusion process from which we obtain a running weighted average of difference indices over the system for each unit at each time step. The unit's own difference index is compared to this in order to determine if it should be moved or not. For units which should be moved, the movability criterion is then applied to see if movement is practical at this time step.

We develop a "diffusion variable", x, for each unit according to the following equation:

$$(d/dt)x(i) = K[\Sigma^{c(i)}_{j=1} x(i,j) - c(i)x(i)] \qquad (4)$$

Where:

x(i)	is the diffusion variable of the i-th unit
K	is the diffusion coefficient
x(i,j)	is the diffusion variable of the j-th unit neighboring the i-th unit
c(i)	is the number of units connected to the i-th unit

The initial value of x is set to the initial difference index for that unit. Eventually, x(i) converges to the average value of difference indices over the system. We limit x to positive values, i.e. if $x(t+1) < 0$ then $x(t+1)$ is set to zero. The diffusion variable and difference index for the unit are both updated when it moves, but this does not guarantee conservation of the sum of x. We overcame this difficulty by modifying the diffusion equation as follows:

$$(d/dt)x(i) = K[\Sigma^{c(i)}_{j=1} x(i,j) - c(i)x(i)] - L \qquad (5)$$

(Effective only for movable units)

Where L is a leak constant.

Each unit then compares its current difference index with the value of x by evaluating the following inequality:

$$G\, x(i) < \text{difference index}(i) \qquad (6)$$

where G is a parameter known as the activation threshold ratio that governs the sensitivity of the system to difference index. If the inequality is satisfied and the unit is movable, it then immediately moves randomly left or right by one step. This procedure has two main advantages. Firstly, units with zero difference indexes do not move. Secondly, as the system approaches its final configuration and the difference indices become small, units with relatively larger indices still move in a process which doesn't stop until the final configuration is reached and all difference indices become actually zero.

6. Simulation results

We evaluated the effectiveness of the method described above by simulating the movement of 10 units to form the triangle shown in **Fig. 9** as the target configuration. At the beginning of the simulation, the units were arranged in a straight line. The parameters we used in the simulation were:

K = 0.02 (The iteration time step in the difference form of equation (5))
G = 1.25 (for connection types "e" and "o"),
20.0 (for connection type "ε")

L = 0.15 (for connection types "e" and "o"),
0.02 (for connection type "ε")

We made connection types "e" harder to move than the other movable types in order to give the units a tendency to form a convex hump.

The sequence of system configurations from the starting point to the final target shape is a stochastic process because of the random direction of movement of the movable units. (**Fig. 10** shows a change of difference variable in a simulation result.) We therefore estimated the efficiency of the configuration method by running 1000 simulation trials, and found that in 964 of these the system reached the required target configuration before 4000 time steps. In the remaining 36 cases, the system fell into a deadlock condition, that is, it was trapped in a configuration different from the target. For example, a triangle with a hole in the middle represents a deadlock condition for 10 units attempting to reach a filled triangle target configuration. Once the hole forms, the units cannot cut the loop and fill with this configuration algorithm. **Fig. 11** is typical of the configuration sequences generated by the simulation.

The randomness in the process makes both the convergence speed and success rate worse as the scale of the system becomes larger. When we simulated convergence to a bigger triangle made from 15 units, we found that the success rate decreased to 734 cases out of 1000 trials.

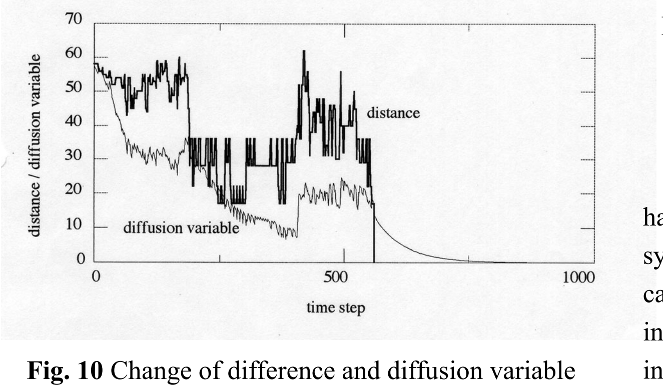

Fig. 10 Change of difference and diffusion variable

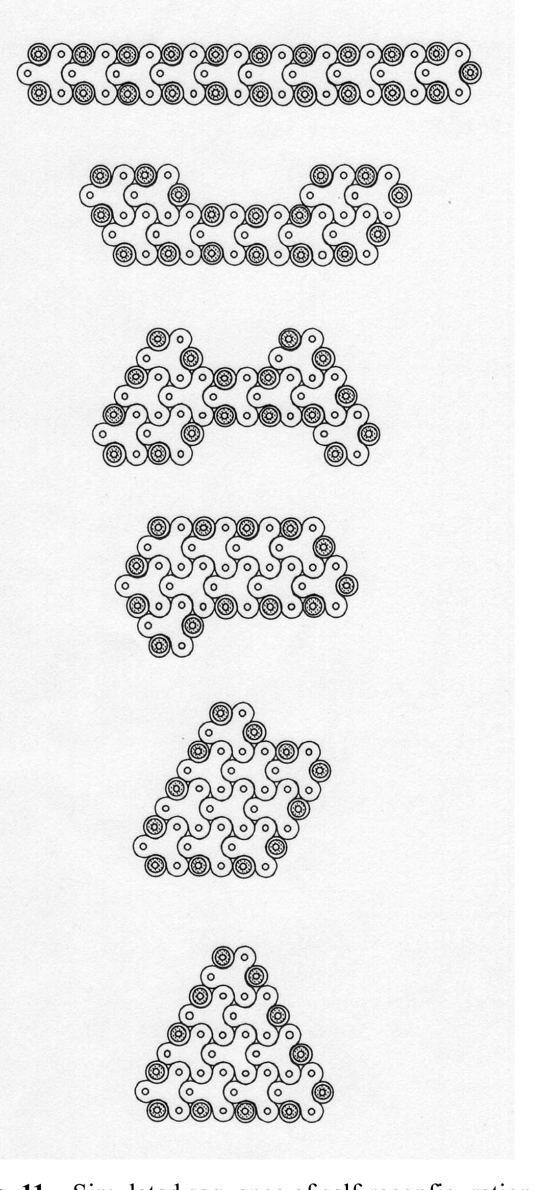

Fig. 11 Simulated sequence of self-reconfiguration

7.Conclusion

The self-configurable machine described here has a structure which is very close to that of biological systems, in that it is made from identical units which all carry the same information, much like the way in which information is carried by chromosomes. Our results indicate that this type of system has the potential for novel functions such as self-organization and self-repair. (**Figs. 12, 13** illustrate such functions available by this system.)

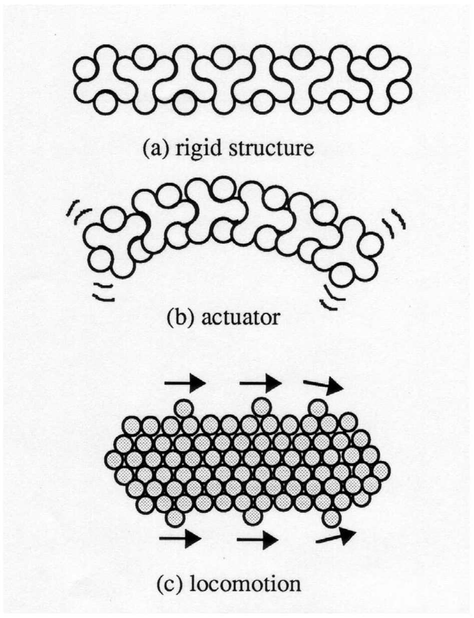

Fig. 12 Various functions of fracta

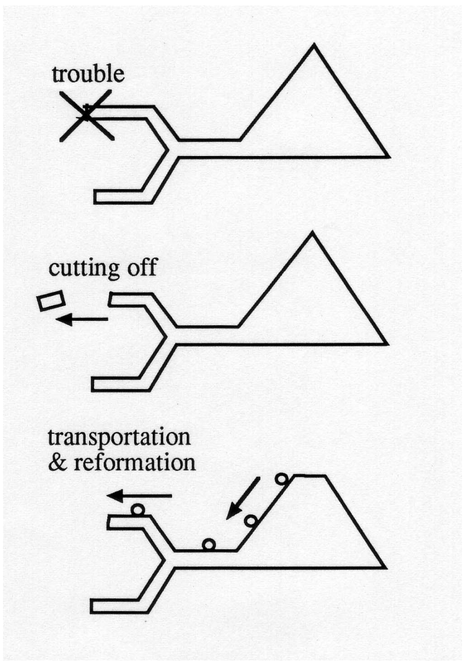

Fig. 13 Self-repairing process by the units
Step 1. Detect and identify faulty units
Step 2. Cut off faulty part
Step 3. Transport spare units and reassemble original part

Although direct imitation of biological systems is sometimes too redundant and ineffective, the problems which modern designers encounter when making highly complicated systems suggest that conventional design methods for artificial systems need fundamental changes. The self-configurable system described here may be a step towards the solution.

References

1) J. von Neumann: Theory of Self-reproducing Automata, Univ. of Illinois Press (1966).

2) C.G. Langton: Self-Reproduction in Cellular Automata, Physica D, 10(1-2), pp.135-144 (1984).

3) A.Lindenmayer : Mathematical Models for Cellular Interactions in Development, J. Theor. Biol., 18, pp.280-315 (1971).

4) L.S.Penrose: Self-Reproducing, Sci. Amer., 200, 6, pp.105-114 (1959).

5) S.Kokaji: A Fractal Mechanism and a Decentralized Control Method, Proc. USA-Japan Symp. Flexible Automation, pp.1129-1134 (1988).

6) T.Fukuda et al.: Concept of Cellular Robotic System and Basic Strategies for its Realization, Computers Electro. Eng., Vol.18, No.1, pp.11-39 (1992).

7) P.Liang, G.Beni : Robotic Morphogenesis, IEEE Int. Conf. Robotics and Automation, pp.2175-2180 (1995).

8) M.J.Mataric, Minimizing Complexity in Controlling a Mobile Robot Population, IEEE Int. Conf. Robotics and Automation, pp.830-835 (1992).

9) G.S.Chirikjian : Kinematics of a Metamorphic Robotic System, IEEE Int. Conf. Robotics and Automation, pp.449-455 (1994).

10) M.Iye, K.Kodaira: Primary Mirror Support System for the SUBARU Telescope, SPIE Proc.2199, pp.762-772 (1994).

11) J.E.Nelson, P.Gillingham: An Overview of the Performance of the W.M.Keck Observatory, SPIE, Proc.2199, pp.82-93. (1994)

Satoshi MURATA (Member)
He received the B.E., M.E. and Dr.Eng degrees in aeronautical engineering from Nagoya University, in 1984,86,and 97 respectively. In 1986, he joined the Mechanical Engineering Laboratory, AIST, MITI. From 2001, he is an associate professor of Tokyo Institute of Technology. His

interests include distributed mechanical system and emergent system design.

Haruhisa KUROKAWA (Member)
He received B.E. and M.E. degrees in precision machinery engineering, and Dr. Eng. in aero/astronautical engineering from the University of Tokyo in 1978,81 and 97 respectively. He is a currently member of Distributed System Design Group of Institutes of Advanced Industrial Science and Technology (AIST).

Shigeru KOKAJI (Member)
He received B.E., M.E. and Dr. Eng. degrees in precision machinery engineering from the University of Tokyo in 1970,72 and 86 respectively. He is a currently the head of Distributed System Design Group and also a vice director of Intelligent system Institute of Institutes of Advanced Industrial Science and Technology (AIST).

Reprinted/Translated from Trans. SICE, Vol.31, No.2, 254/262 (1995).