IEEE/RSJ International Workshop on Intelligent Robots and Systems '89, Sep. 4-6, 1989, Tsukuba, Japan

DESIGN OF AN AUTONOMOUS AND DISTRIBUTED ROBOT SYSTEM: ACTRESS

Hajime Asama, Akihiro Matsumoto*, Yoshiki Ishida**

Chemical Engineering Laboratory,
RIKEN (The Institute of Physical and Chemical Research),
Hirosawa 2-1, Wako-shi, Saitama 351-01, JAPAN

* *Department of Mechanical Engineering*
Faculty of Engineering, Toyo University
Kujirai-Nakanodai 2100, Kawagoe-shi, Saitama 350, JAPAN

** *Department of Precision Machinery Engineering*
Faculty of Engineering, the University of Tokyo,
Hongo 7-3-1, Bunkyo-ku, Tokyo 113, JAPAN

Abstract

A new concept of an advanced robot system, ACTRESS (ACTor-based Robots and Equipments Synthetic System), is presented in this paper. ACTRESS is an autonomous and distributed robot system composed of multi robotic elements. Each element is provided with functions to make decisions with understanding the target of tasks, recognizing surrounding environments, acting, and managing its own conditions, and to communicate with any other components. In order to manage multiple elements to achieve any given task targets, the protocol for communication between elements is discussed for cooperative action between arbitrary elements. This paper deals with the conceptual design of ACTRESS, focusing on the methodology for synthesizing the autonomous and distributed system. Also based on an assumption that mobility is the indispensable function for advanced robot systems, an experimental system using "micromouse"'s is developed as a primitive example of ACTRESS.

1. Introduction

The advanced robot systems are increasingly demanded in various industries in various fields. One example of urgent needs for the advanced robot application is the robotization of maintenance tasks in nuclear power plants, where the decrease of radiation exposure and mental weariness of workers, and the improvement of total reliability and productivity of the plants, are required. The ACTRESS is originally designed as a maintenance robot system. In the process of conceptual design of the maintenance robot system, the actual tasks in nuclear power plants were investigated and analyzed, where parallel action by multiple maintenance workers and their cooperation are essential. By modeling this maintenance workers' system, an autonomous and distributed robot system ACTRESS is proposed in this paper.

Some distributed robotic systems have been already developed[1][2]. Though there are differences in elements' structures, total architecture, and processing mechanisms, these systems act with organizing a body by combining multiple elements physically, and communication between elements is rather frequent. However, considering efficiency of task achievement and diversity of system's functions, a complex distributed system consisting of multiple bodies is required as a superclass robot system. Moreover, another study on communication between uniform mobile robots was reported[3], but robots with individual characters should be incorporated in the system for the purpose of generality.

On the other hand, the advanced robots have been or being actively developed under development by companies, universities, and institutes individually, and each of them have different structures and different functions. Therefore, taking account that the results of these research works are difficult to evaluate relatively by certain criteria, any robotic researches should not be competitive with each other, but acceptant as an integrated system for cooperative tasks. In order to develop the integrated system, the bottleneck is the technology to synthesize multiple robots.

The main objective of the ACTRESS is to develop the technology to synthesize multiple robotic elements including computing systems such as modeler or planner as well as robots. The ACTRESS is supposed to be applicable not only to maintenance but also to complicated tasks in restricted environment such as under-water, calamity, bioindustry, space, and desert.

2. Concept of the ACTRESS

The new strategy in developing advanced robot systems should be to distribute the functions to multiple robots and other equipments. Therefore, the methodology of design of an autonomous and distributed robot system should be established from the conceptual design stage.

2.1 Requirements

Analyzing such tasks as maintenance tasks which are supposed to be difficult to robotize, the requirements for the robots realizing the tasks are often incompatible, and the design of the robots is generally a trade-off problem between these requirements. Some examples of incompatible requirements are listed below;

(1) accurate positioning of heavy objects,

(2) handling of large size objects in a narrow working environment,

(3) manipulation of diverse or unexpected objects, etc.

It is needless to say that it is impossible to design robots with these requirements by the existing technology, moreover it is supposed to be limited to find in any solutions by the conventional strategy to develop any advanced robot systems, which is to provide a single robot with superior multi-functions, even if the robotic technologies advances in future. The only breakthrough of the design problem of advanced robot is to develop the technologies to distribute the required functions to multiple robotic elements.

2.2 Robot System based on Actors Formalism

As The word "ACTRESS" comes from "ACTor-based Robot and Equipments Synthetic System," basic idea for organizing the ACTRESS is based on *the Universal Modular ACTOR Formalism* by Hewitt[4]. This formalism provides a computational model in information processing, and in this formalism, data structures and control structures are inseparably represented by a single kind of objects called *actor* and message passing between the *actors*. This formalism is quite suitable for parallel processing.

In designing ACTRESS, this formalization conforms considerably to the architecture of an autonomous and distributed robot system in the following points:

(1) In this formalism, the objects in information processing are modulized to uniform units, *actors*, while the modules in working operations in this robot system are robotic components.

(2) In this formalism, control is embedded in each object identifying with data flow, while each robotic components in this robot system is autonomous so that control function of the component is incorporated in itself.

(3) In this formalism, behavior between *actors* is defined in terms of message passing, while robotic components communicate with each other for cooperative operations in this robot system.

(4) The autonomous and distributed robot system is characterized by efficiency by parallel actions of robotic components, as this formalism is efficient for parallel processing.

Based on the above discussion, we define robotic components as *robotors*, of which the robot system is composed. Namely, a *robotor* is a robotic *actor*, which is not supposed only to control data, but also to move or manipulate some objects. Message passing corresponds to communication between *robotors*, which is illustrated in fig. 1.

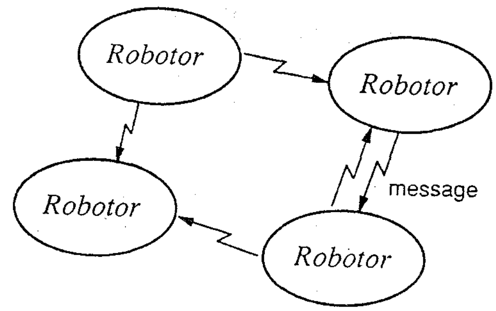


Fig. 1. Communication between *robotors*

2.3 Robotors

Each *robotor* should be an autonomous component, which is assumed to have at least two basic functions:

(1) An ability to make decisions, understanding the target of tasks, recognizing surrounding environments, acting, and managing its own conditions.

(2) An ability to communicate with any other components for parallel tasks with avoiding interference in components' motion, or for cooperative tasks with transmitting necessary information.

In order to make the framework of the ACTRESS general, *robotor* doesn't have to be a robot, if the above-mentioned functions are satisfied. Therefore, not to mention intelligent robots, any elemental systems which are provided with intelligence of the two information processing functions, should be regarded as *robotors*, which include sensor devices, single use robots, specific equipments, and any computing systems of environment modeler, simulator, planner, CAD, and any other database, or knowledge base systems. Some typical examples of *robotors'* inner constructions are listed in fig. 2.

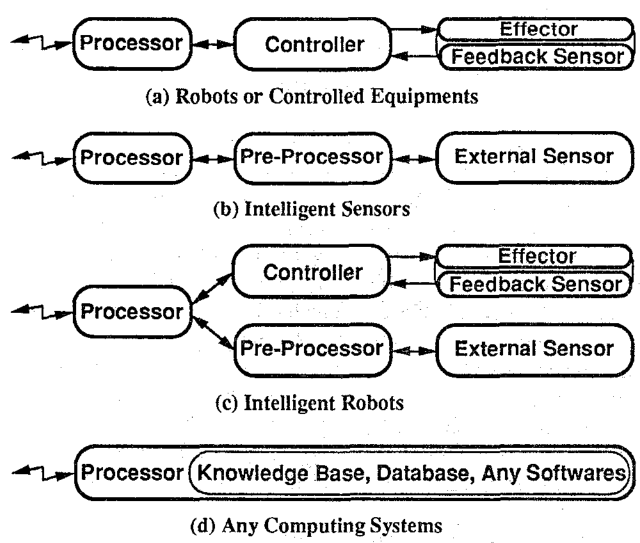


Fig. 2. Typical examples of *robotors'* constructions

2.4 Architecture of the ACTRESS

As mentioned before, the ACTRESS is composed of a set of *robotors*, and the *robotors* may have different structure and functions, because it is natural to accept the *robotor*'s individuality. Fig. 3 illustrates the concept of the ACTRESS, which is an autonomous and distributed system connected by communication network.

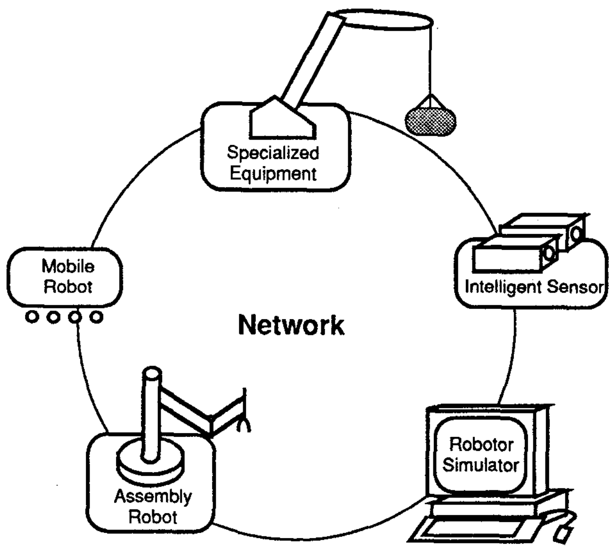


Fig. 3. Concept of the ACTRESS

The ACTRESS is characterized by following advantages:

(1) **Reliability**
As the ACTRESS is a modulized system, high reliability can be realized by replacing the defected *robotor* in case of troubles.

(2) **Extensibility**
As each robotor acts independently under normal conditions, the ACTRESS is easy to extend only by introducing the new *robotors*. Communication between any *robotors* is assumed in the ACTRESS.

(3) **Flexibility**
The ACTRESS is possible not only to deal with any change of requirements, but also to generalize its applications.

(4) **Efficiency**:
It is efficient for multiple *robotors* to achieve their own tasks parallelly. In addition, diverse tasks can be achieved by arbitrary combination of *robotors*.

(5) **Adaptability**
The ACTRESS is adaptive to existing facilities, because the ACTRESS includes any robots and equipments, and any cooperative operations are supposed.

Concerning communication between *robotor*s, two communicating conditions should be supposed. When a *robotor* acts independently, it is only required to monitor the status of other *robotor*s with occasional communication. On the other hand, when a *robotor* executes a task cooperating with other *robotor*s, it is required to share the control signals with frequent communication. Therefore, a hierarchical protocol is provided in the ACTRESS.

3. Communication in ACTRESS

The communicating function between elements is vital in order to manage the autonomous and distributed system to achieve given tasks. With regard to the ACTRESS, communication scheme, especially protocol used in the communication, should be defined in coordination control between multiple *robotors*. Some communication methods are already reported[5], which are rather *ad hoc* schemes. We discuss conceptual design of general protocol for the whole system, in order to deal with arbitrary control between any *robotors*.

3.1 Requirements of the Protocol

We should take account of two kinds of protocols for communication in ACTRESS. One is the protocol for establishment of connection and guarantee of data transmission, and another is the protocol for recognizing and understanding the content of transmitted data. We define the former as *communication protocol*, and the latter as *message protocol*, which is named after *message* transmitted between *actors*. The *communication protocol* is so-called protocol in general computer networks. On the other hand, the *message protocol* corresponds to languages in communication. In other words, the *communication protocol* denotes the specification of the communication, and the *message protocol* denotes the syntax of the transmitted information. Concerning transmitted information, any sensor signals or control signals can be shared by means of the *communication protocol* for dynamic coordination control between *robotor*s, which corresponds to data communication by bus connection. On the other hand, messages representing status of the self *robotor* are transmitted by means of the *message protocol*, which corresponds to commands communication by LAN (Local Area Network) connection.

Protocol should be applicable to any communicating structures. In addition, a communication structure depends on the hardware of robots and the environment, not on the protocol. Low layer of the *communication protocol* should be designed according to the *robotor's* hardware, while the interface between layers should be defined.

Concerning the protocol in the ACTRESS, reliability and extensibility are required. Reliability in communication can be represented as follows;

(1) low possibility of error occurrence,
(2) high maintainability for errors recovery, and
(3) tolerance in case of faults or errors.

Besides, extensibility denotes no limitation for extending or modifying the total system, which can be represented as follows;

(1) high ability of system reconfiguration when adding new elements to the system, and
(2) upper compatibility in extended or modified system.

Protocol in LAN is acceptable as *communication protocol.* In other words, the *communication protocol* should be compatible to the LAN protocol considering that the ACTRESS includes the communication between computers. OSI (Open Systems Interconnection) Reference Model was proposed as a general framework of computer networks[6], where hierarchical architecture with seven layers is defined. The most essential requirement for the *communication protocol* is high reliability of communication. In order to achieve high reliability of communication, we should consider the error from noises along communication media, and the error from misrecognition of the information. Protocol hierarchy realizes high reliability by providing each layer with function of preventing misunderstandings and their recovery. In addition, as any case of LAN connection or bus connection should be supposed, protocol should be hierarchized and distributed, in order to transmit data of different quality using identical media without misrecognition. Typical communicating structures are shown in fig. 4.

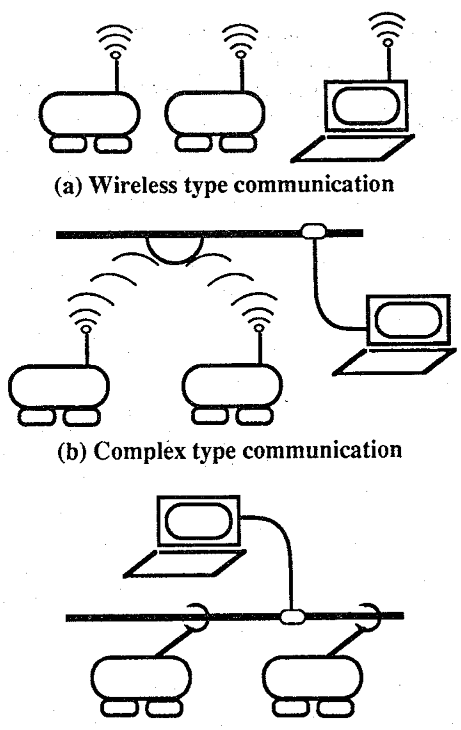


Fig. 4. Communicating structures

3.2 Design of the Message Protocol

The *message protocol* should be independent of the *communication protocol.* In order to clarify the requirements for protocol design, information which is transmitted in the ACTRESS should be discussed first. Communication of diverse information should be supported according to the level of *robotors'* combination, which includes electrical signals such as control signals or sensor signals, and from low level data such as position, velocity, and force, and to high level data such as task target. The architecture of the information corresponds fairly well to one of robot languages, which was expressed as hierarchy of command level, object level, and task level[7]. By means of network, communication between arbitrary *robotors* at an arbitrary level should be possible.

We introduce a new framework with *levels* in the *message protocol.* The levels are distinguished from the layers in the *communication protocol*, and not always hierarchically structured. The levels are divided as follows:

(1) **Control Level**
Control signals such as signals for controllers or from sensors are transmitted in this level. Information transmitted in this level depends on the hardware of equipments.

(2) **Physical Level**
Physical data such as position, velocity, and force are transmitted in this level. Information implies physical dimensions.

(3) **Procedural Level**
Procedures to operate *robotors* are transmitted in this level. Interpreter on procedures should be provided for each *robotors*, and commands, requests, and acknowledgements can be transmitted between *robotors* using the interpreter.

(4) **Knowledge Level**
Knowledge stored in each *robotors* is transmitted in this level. Only the receiver in communication can utilize the knowledge. The environment model which mobile *robotors* refer to, corresponds to the information in this level.

(5) **Conceptual Level**
Concepts including intentions or objectives are transmitted in this level. The task targets which are orders to make any *robotors* to accomplish any tasks, correspond to the information in this level.

In general, it is difficult to classify all the transmitted information into these levels, but appropriate definition is required for designing applicable protocol. The information through all above-mentioned levels in the *message protocol* can be data transmitted in the application layer in the OSI reference model. Taking account of overheads in communication, however, low level information in the *message protocol* is actually desired with abbreviation of some high layers in the *communication protocol* as shown in fig. 5.

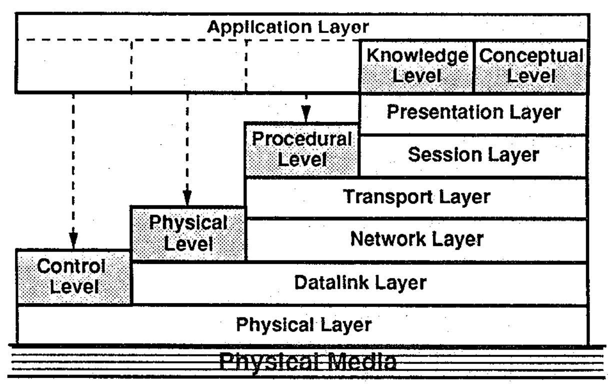


Fig. 5. Layer abbreviation of levels in the *message protocol*

The rule of abbreviation of layers can be defined previous to the communication, and can also be defined as occasion demands in the middle of communication. Comparing to the robot language, programs at motion level, at object level, and at task level, correspond respectively to information transmitted at physical level, at procedural level, and at conceptual level. As knowledge is not supposed to be shared in the existing robots, there is no programming level in robot language, which corresponds to knowledge level in the *message protocol.*

4. Experimental System with Micromouses

As an example of the ACTRESS, an experimental system is developed. The purpose of the experimental system is not the advancement of the functions of robotic elements which the system consists of, but the evaluation of performance of the synthesized system. Assuming that mobility is the indispensable function for advanced robot systems, micromouses are adopted as *robotors* in the experimental system, which are the most primitive mobile robots with minimum functions.

4.1 A Task of Moving Obstacles

For the purpose of evaluation of the system's performance, the primitive task of moving scattered obstacles is dealt with, where both independent motion and cooperative motion of the robot elements exist. While no manipulators are provided for micromouses, the task can be accomplished only by pushing obstacles to the side of the room, and the goal of the task is a state of all the obstacles being moved to the side of the room.

The obstacles are assumed to have certain dimension and suitable mass, which micromouses can push. Therefore, the obstacles are defined by attributes of two dimensional geometry and weight. Obstacles' weight is classified into two classes of *light* and *heavy* in this system. *Light* is the case where one micromouse can push to move the object, and *heavy* is the case where cooperation of two micromouses is necessary to push it. The geometry of obstacles and the room is rectangular with certain height, so that the task of moving obstacles by micromouses can be discussed in two dimensional space. An example of the task environment is illustrated in fig. 6, where 1 to 10 are the obstacles, and A and B are the micromouses.

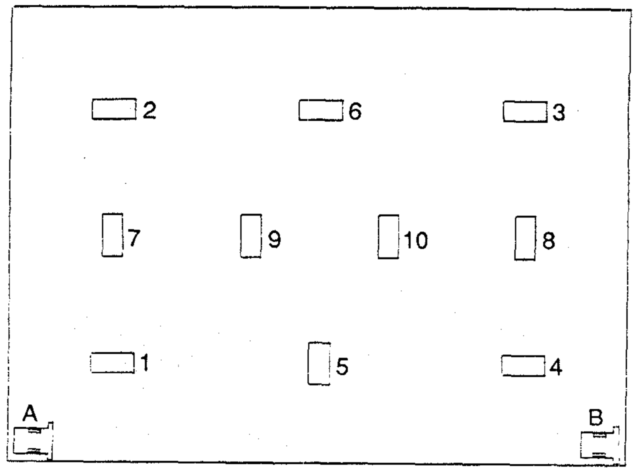


Fig. 6. Example of the task environment

4.2 System Configuration

The configuration of the experimental system is shown in fig. 7. The system consists of three *robotors*, namely two micromouses and a micromouse simulator. The two micromouses are controlled by each personal computers through digital I/O, and sensor data are sent to the computer as well.

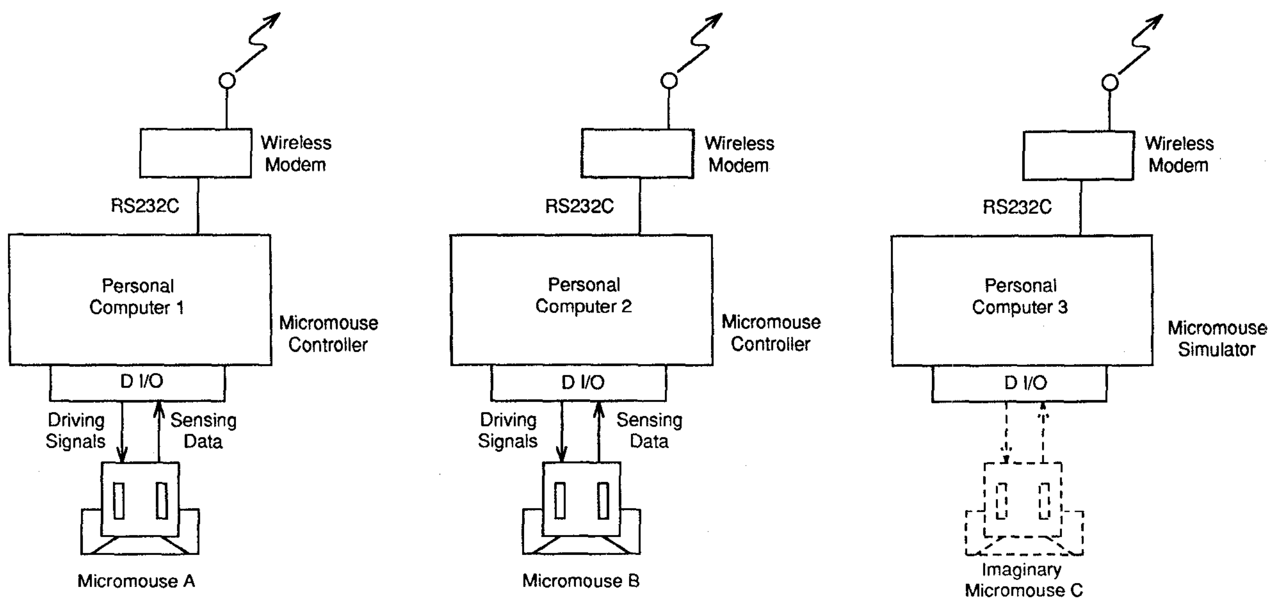


Fig. 7. Configuration of the experimental system

Two main wheels of the micromouse are driven by two stepping motors, optical sensors for position detection and tactile sensors for collision detection are provided for micromouses. The decisions of contact with an obstacle, transferability of the obstacle, and arrival at the goal position, are based on the sensed information.

The micromouse simulator can be regarded as another imaginary micromouse, where an environment model is implemented. Wireless modem is provided for each computers, in order to communicate with any other micromouses. The specification of the communication by the wireless modem is identical to RS-232C. Fig. 8 is the picture of the experimental system with two micromouses.

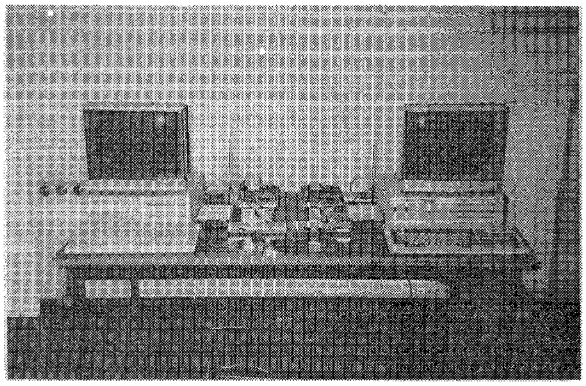


Fig. 8. Picture of the experimental system

The states of all the obstacles are managed in each micromouses according to information acquired by communication. The states of the obstacles are described by the following three kinds of labels;

N: not handled yet,

F: finished,

X: under operation by micromouse X.

4.3 Task Planning

An identical software is installed in all the micromouses. The algorithm is roughly divided into the following modes;

(1) individual mode,

(2) cooperative mode sending requests to another micromouse, and

(3) cooperative mode for supporting another micromouse.

The main procedures in the algorithms are summarized as follows;

(1) selection of an obstacle,

(2) path planning of the obstacle being transferred,

(3) decision of contact position with the obstacle,

(4) path planning of the micromouse,

(5) movement for approaching, and

(6) transference of the obstacle by pushing.

An obstacle being transferred is selected by estimating the distance between the micromouse and the obstacles with the state label N. Path planning of the transferred obstacle is planned so as not to collide against any other obstacles. The contact position denotes the initial position and direction of the micromouse for obstacle transference, which is modified in the case of cooperative motion, as shifted parallel to the obstacle's surface. Micromouses approach to the obstacles from the opposite side of obstacles against the wall of the room. In case that the obstacle is found too heavy to be transferred by a single micromouse, the algorithm is switched from the individual mode to the cooperative mode sending requests to another micromouse. In the case of any request of support being sent, the algorithm is switched to the cooperative mode for supporting another micromouse. When the micromouse finishes transferring any obstacles with no request of support from others, the algorithm is returned to the individual mode. Fig. 9 shows the flow in individual mode.

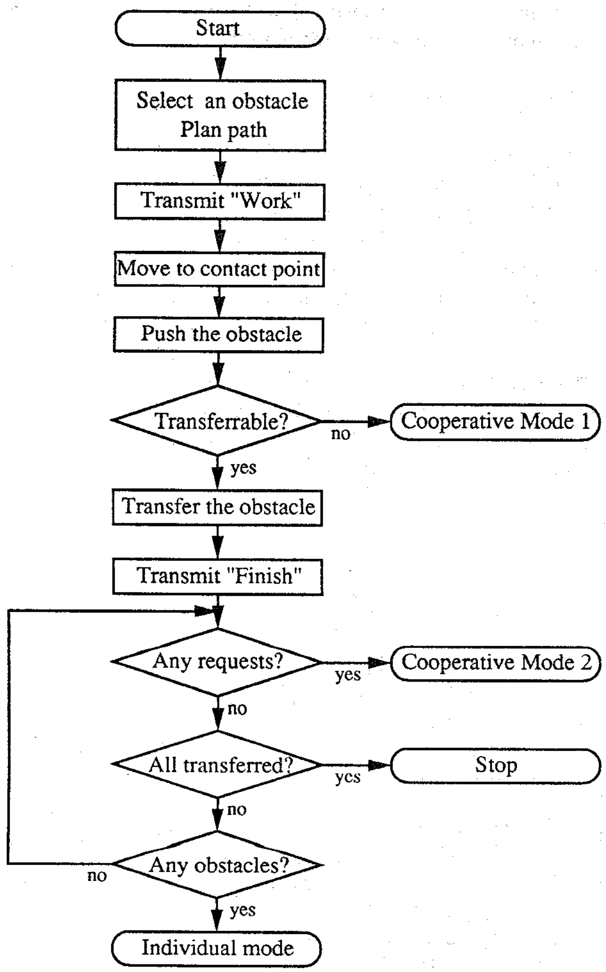


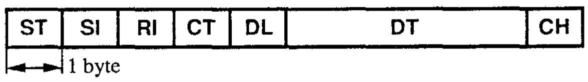
Fig. 9. Flow chart of algorithm in individual mode

In the path planning of the micromouse, motions of two wheels are calculated, associated with the obstacle's path and the contact position. In case that any collisions are estimated during the movement for approaching or transference by pushing, the micromouse which determined the movement earlier gets priority.

Micromouses check the request of support from other micromouses when it finishes pushing any obstacles, or when it requires support of others, so that the conflict of requests can be avoided. In cooperative tasks, the micromouse which sends requests to another, takes the leadership.

4.4 Protocols

In this system, every *robotor* is an individualized micromouse, so that it is not necessary to communicate frequently between two *robotors*, even when two micromouses are doing cooperative tasks. Therefore, protocol corresponding to the procedural level of the *message protocol* is supported in this system. The data format of the protocol in this system is shown in fig. 10.



ST : Start
SI : Sender Identifier
RI : Receiver Identifier
CT : Communication Type
DL : Data Length
DT : Data
CH : Checking Sum

Fig. 10. Data format of the protocol

In this figure, an *identifier* denotes the identification of *robotors* (micromouses, in this case). The length of DT is variable, which can be defined by DL. *Communication type* in this system is shown in table 1.

Table 1. Communication type

Symbol	Contents
TXD	Sending Data
RXD	Acknowledge of Data Reception
ERR	Failure of Data Reception

In this system, DT includes *working condition* and *obstacle identification. Working conditions* are listed as follows:

I. Be ready for action.
II. Start pushing an obstacle.
III. Request for cooperation.
IV. Finish pushing an obstacle.
V. Be ready for cooperation.
VI. Completing the task.

When the micromouse turns operative, the micromouse broadcasts working condition I. After the micromouse recognizes that all the micromouses are operative, and the micromouse selects the obstacle being transferred, it broadcasts its working condition II. In case that the obstacle is too heavy to be transferred, and no request for cooperation is sent, it broadcasts its working condition III. When the micromouse finishes transferring the obstacle to the goal position, it broadcasts working condition IV. In the cooperative mode for supporting another, when the micromouse moves to the cooperative contact position, it broadcasts working condition V. When there are no obstacles with the state label N, it broadcasts working condition VI.

4.5 Experiments and Discussions

Experiments to move up to ten obstacles to the side of the room using two micromouses and one micromouse simulator, were performed. The executed results in the task environment of fig. 6, which were simulated on the displays of two micromouse controllers, are shown in fig. 11 and fig. 12. In the case of controlling actual micromouses, possibility of accomplishing tasks was not high in this system, because of the unreliability of sensor signals and the low accuracy of mobile mechanisms, so that improvement of sensors and mechanisms was concluded to be necessary. However, monitoring of micromouses' performance, parallel actions and cooperative actions of multiple micromouses with communication based on the designed protocol were verified.

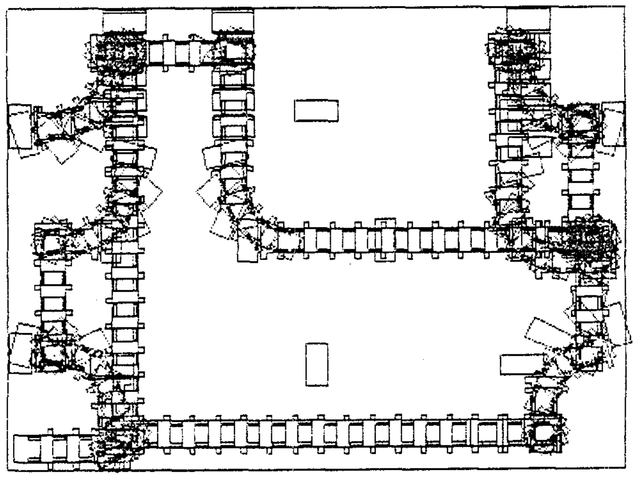


Fig. 11. Simulated results of micromouse A

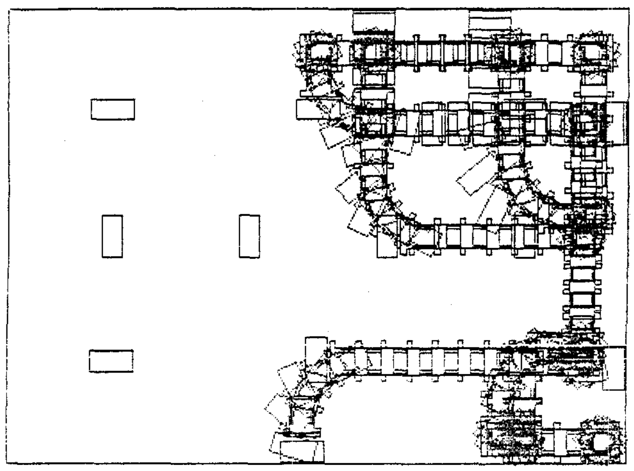


Fig. 12. Simulated results of micromouse B

In this experiments, the weight was only the unknown factor for micromouses, while geometric information can be acquired by communication with a micromouse simulator with an environment model. However, various cases depending on functions of micromouses and computers, should be assumed even concerning the experimental system using micromouses. Then, requirements for communication in the experimental system should be clarified.

(1) **Estimation of the own position**

The position of the micromouse is desired to be managed by itself. Sensors for environment recognition should be installed on micromouses in order to estimate its own position. In case of a micromouse driven by stepping motors, its own position is necessary to calculate based on the rotational displacements of the micromouse's wheels.

(2) **Recognition of obstacles and other micromouses**

Local environment information is desired to be recognized by the micromouse using sensors, but global environment information should be acquired communicating with a computing system with an environment model. In order to realize intelligent planning of paths, motions, or tasks, it is necessary for a micromouse to refer to the global information, but the micromouse has to recognize the moving objects near by in the actual motion.

(3) **Collision detection**

In order to detect collisions with obstacles and other micromouses, sensors for collision detection are desired to be provided for micromouses. However, in order to detect collisions in a simulating system, an environment model should be constructed and managed dynamically according to the movements of each micromouses. It is efficient that the simulating system sends messages and the desired paths for avoiding the collisions when any approximations are estimated by the simulating system. The warning messages of the approximations or collisions sent from the micromouse simulator corresponds to the data at procedural level in the *message protocol*. The information of position, orientation, and velocity of micromouses corresponds to the data at physical level in the *message protocol*.

(4) **Path planning**

As the number of *robotors* becomes large, collisions between micromouses and obstacles become more possible, and it becomes more difficult to plan paths avoiding not only obstacles but also other moving micromouses previous to motion execution. Therefore, local path planning function should be provided for each micromouses according to the sensed or transmitted information of the dynamic approximations or collisions, and global path planning function should be provided for the micromouse simulator with sending the planned path according to the requests from micromouses. Assuming software resources are distributed and shared, transmission of the knowledge or programs on path planning should be possible, which correspond to data at knowledge level in the *message protocol*.

(5) **Task assignment**

If we assume no supervisory system in an autonomous and distributed system, the task assignment is difficult to optimize. An intelligent task assignment is introduced by decision making based on self evaluation of performance, and by communication at concept level in the *message protocol*. The assignment once planned should be easily modified according to any additional information acquired by the actual motion of micromouses. The task assignment previous to the actual motion is not practical in case of large number of obstacles and/or micromouses with different functions. Moreover, more kinds of *working conditions* such as "busy" and "idle" should be prepared for describing micromouses' states.

5. Conclusion

As an advanced robot system of next generation, a concept of an autonomous and distributed robot system, ACTRESS, was designed. The ACTRESS has such a general architecture that multi functions as well as high reliability, extensibility, flexibility, efficiency, and adaptability, can be realized by parallel action and cooperative action. The communication function in the ACTRESS and basic design of the protocol was discussed. Then, an experimental system using micromouses was developed, and appropriateness of the synthetic system was verified. Finally, communication between micromouses and a micromouse simulator was discussed assuming various cases. Considering the experimental results and discussions, functional improvement of robotic components and increase of kinds and/or numbers of the components are the future problems.

Reference

[1] Fukuda, T., and Nakagawa, S.: "A Dynamically Reconfigurable Robotic System (Concept of a System and Optimal Configurations)," Proc. IECON '87, (1987)588-595.

[2] Kokaji, S.: "A Distributed Mechanical Structure and its Control System," Proc. of ISMM Int. Symp., Computer Appl. in Design, Simulation and Analysis, (1988)58-62.

[3] Premvuti, S., and Yuta, S.: "Radio Communication on Self-Contained Mobile Robots for Cooperative Motions," Proc. of 5th Annual Meeting of Robotics Society of Japan, (1987)193-196(in Japanese).

[4] Hewitt, C., *et al.*: "A Universal Modular Actor Formalism for Artificial Intelligence," Proc. of IJCAI, Palo Alto, California, (1973)235-245.

[5] *e.g.* Fukuda, T., *et al.*: "A Study on Dynamically Reconfigurable Robotic Systems (10th Report, Communication Network System of CEBOT)," Proc. of 6th Annual Meeting of Robotics Society of Japan, (1988)575-576(in Japanese).

[6] Tanenbaum, A. S.: "Computer Networks," Prentice-Hall, Inc., (1988).

[7] Bonner, S., and Shin, K. G.: "A Comparative Study of Robot Languages," IEEE Computer, 15, 12(1982)82-96.