'92 KACC 1992. 10. 19~21

Stochastic Learning Scheme in Quasi-Distributed Management Method for Autonomous Manufacturing Systems

Keiji Suzuki*, Yukinori Kakazu**

* **Dept. of Precision Eng., Faculty of Eng., Hokkaido university.
North-13, West-8, Sapporo 060, JAPAN

ABSTRACT

This paper proposes a new framework of an autonomous and distributed flexible manufacturing system — Multi Client Robot Groups (*MCR*) — and describes a stochastic learning scheme applied to managerial problems of the system. The *MCR* is composed of groups of manufacturing robots, named Client Robots (*CRs*), which are capable of both versatility and independence in their performances. The *MCR* is expected to have high performance because the *MCR* can perform concurrent and corporative processing. However, the system performance is determined by the organizations of the *CR* groups. Therefore the treatment of the managerial problems and organizations of the system are important problems. In this paper, it is assumed that *CR* groups being able to processing tasks are selected stochastically based on the strengths of the robot groups. The learning scheme adjusting the strength is introduced to organize the groups in the system and control the each performance of the groups according to the total system performance. Finally, some experimental results of the learning scheme are shown.

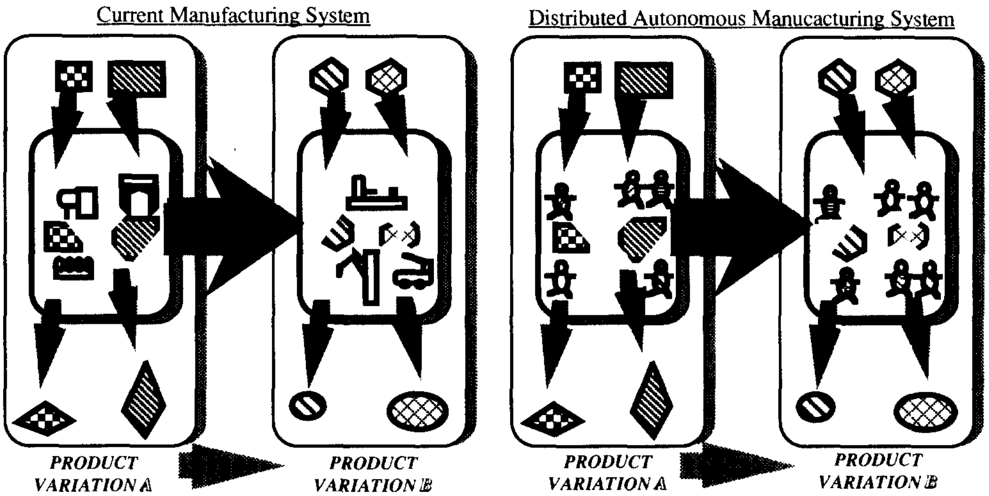
1. Introduction

This paper addresses the management problem in task processing by autonomous and distributed robot groups. In recent years, autonomous and distributed concept has been attracting interest in various fields which require flexibility.[3] One of these fields is an autonomous and distributed manufacturing system.[1][6] The reason that the autonomous and distributed structure in manufacturing systems attracts attention is due to the following points: Changes in the social environment surrounding manufacturing systems and advancement of capability of technical development, response to multi model, small quantity production, and quick response to on-demand systems and failure resistance. In other words, it is a strategy that an autonomous and distributed manufacturing system allows the structures of its manufacturing facilities to dynamically recombine themselves in mechanisms and information processing according to the situations, thus realizing desired flexibility and robustness. In such an autonomous , distribution oriented manufacturing systems, robot systems are incorporated as components and play a major role in exerting dynamic flexibility. Such robot systems are inevitably required to equip autonomous and distributed structures.

Multi-Client Robot Groups (*MCR*) consist of robots with flexibility, multiple functions, cooperativeness and are conceived as a robot system composed according to such purposes.[4] A management problem is handled by the task processing of this system. The *MCR* conceived here consists of semi-organized sets of numerous, nonuniform multi-functional robots (*CRs*). The *CRs* are regarded as a task processing facilities that respond to various processing targets and organize themselves to individual or cooperative processing structures according to their requirements, and exerts a high degree of freedom. Individual *CR* is to have some processing functions which are independent or common with other *CR*'s, and one function is to generate other functions by cooperation with other *CR*'s. Furthermore, it can be considered that one *CR* is composed of several sub-*CRs*.

The initial state of the *MCR* is not the possession of a completely organized structure. The organization of this system is achieved by orienting the entire system relieved from the relation of *CR* groups, indicating competition and deadlock against some proposed processing targets. Each *CR* becomes a "client" that responds to various processing targets and possesses a processing method for the proposed target by means of checking its own processing functions or referring to other processing functions for cooperation. The *MCR* utilizes a variety of such relations among numerous Clients for organization. The variety of structures, thus, offers the *MCR* flexibility.

Consequently, in the *MCR*, the handling and management of the relation between CRs generated by processing targets becomes one of the important issues. The control methods for systems composed of such autonomous elements are classified largely into the following: an autonomous and distributes method in which each constitutional element regards the other element groups as an environment, and a centralized management method in which a supervisor is provided with



(a) Current Manufacturing Systems need to reconstructs.

(b) Distributed Autonomous Manufacturing Systems can adapt by changing the organizations.

Fig.1 Schema of the different between current manufacturing systems and the distributed autonomous manufacturing systems.

information about the constitutional elements. Among systems using former method is that which adopts a connectionist model.[5)]

An example of the latter is a CIM concept.[2)] The system discussed in this paper is based on the latter, supervising method, coupled partly with distributed management, i.e., a quasi-distributed management method.

In the next section, characteristics of autonomous and distributed manufacturing systems are compared with other architectures.

2. Autonomous and distributed Task Processing System and Its Management Method

One application which requires the properties of autonomous and distributed robot systems is manufacturing systems. In conventional manufacturing system, high-level functions were provided for manufacturing equipment in response to requirements for high-level products with limited kinds of items. In recent years, as an attempt to respond to multi model, small quantity production, the mode of the manufacturing systems has been developed into FMS and CIM of which architectures have been established as its management method in order to introduce flexibility into high-level functions. Nevertheless, the flexibility required for manufacturing systems including the advancement of varying social requirements, varying management of corporations themselves, and the demand for shortened development time is about to exceed the capacity of the architecture such as CIM.

This also seems to require drastic transformation with respect to production system design. In order words, the required philosophy is not system design on the premise of production for limited kinds of items but ultimately architecture of manufacturing system in response to limitless kinds of items. Therefore, autonomous and distributed architecture is attracting interest as a basic architecture which has flexibility and redundancy for such manufacturing system design. In the conventional manufacturing systems, a change in the item to be produced requires the replacement of their constituents and overall modification of the organizational mode, as shown in Fig.1(a). However, an autonomous and distributed basic system configuration enables response to the above-stated changes which will frequently occur by the dynamic rearrangement of constituents and organizational mode, as shown in Fig1.(b). As constituents for the system having such a mode, robot systems capable of flexible response to environmental changes are considered to play important roles.

Although these autonomous , distribution-oriented robot systems excel in flexibility, expendability, multi-functionality and other properties from the standpoint of its configuration, the problem is how properties such as optimally and constancy are obtained. In other words, this is the problem of the system management method for task processing. It also depends on the characterization of each robot, an independent and dedicated character or a cooperative and universal character. With the former, the central issue is the problem division of job into tasks in processing stream; in the latter, it is the organizing the processing robots to get the adaptive structure for the given tasks. The *MCR* discussed in this paper is based on the latter, and the problem of how to handle its organization is described in the following. Especially in the case of focusing on cooperation, the major issues are optimally and consistency with respect to an overall purpose. Management methods for the above include autonomous and distributed methods and centralized methods. In the case of systems with extremely high dispersively and variety, complete centralized management methods are considered to suffer from difficulties, and yet also in autonomous and distributed methods, the problem

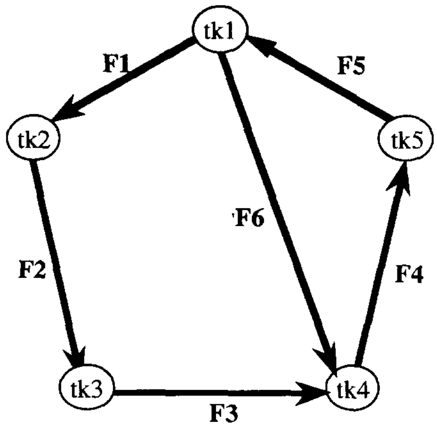


Fig.2. Relations between tasks and their functions.

remains as to how each robot knows the overall purpose and cooperates with others, even though local consistency is easily achieved.

From this perspective, the discussion here starts with centralized management to a quasi-distributed management method based on the stochastic learning scheme. In the next section, the definitions of the system are stated in order.

3.Functional Relations Between MCR and Jobs

In this section, the functional relations between given jobs and the *MCR* are described. A task Tk_i is an element of a certain job and a set of tasks *TASK* is defined as:

$$TASK=\{Tk_i \mid i \in J\} \tag{1}$$

The *MCR* is given a set of jobs *JOB* and it is defined as:

$$JOB=\{JOB_i \mid i=1,2,\cdots,m\} \tag{2}$$

$$JOB_i = (X_i, \leq) \text{ where } X_i \subset TASK \tag{3}$$

The sequence of tasks JOB_i is a plan to arrive at a finished product. Each processing method between Tk_i and Tk_j ($Tk_i < Tk_j$) is assumed to be known and the processing method F_i is described as function:

$$F_i = \delta(Tk_i, Tk_j) \tag{4}$$

$$F = \{F_i \mid i \in J\} \tag{5}$$

The *MCR* is defined as the set of Client Robots, *CRs*, and it is represented in eq.(6). Also the set of elemental functions *F'* of the *MCR* is given in eq.(7), where $F(CR_i)$ represents a function set of CR_i.

$$MCR = \{CR_i \mid i = 1,2,\cdots,n\} \tag{6}$$

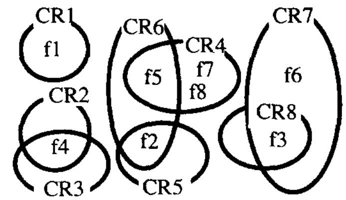


Fig.3. Single Function set of Client Robots.

processing functions	Combinational functions of CR
F1	(f4,f5)
F2	(f1,f2)
F3	(f4)
F4	(f1,f7)
F5	(f2,f7)
F6	(f1,f8)

Table.1. Task processing Functions and Combinational functions of *CRs*.

$$F' = \bigcup_i F(CR_i) = \{f_k \mid k=1,2,\cdots,\eta\} \tag{7}$$

All the functions of the *MCR*, including cooperative functions,are defined as $F_M \subset 2^{F'}$. An example of each *CRs*' functions is given in Fig.1 and the Table.1 is an example of cooperative functions.

It can be assumed that F_M includes equivalent functions. Therefore, the equivalent relation F_R is defined over F_M and it is given in eq.(8).

$$(f_i, f_j) \in F_R \rightarrow f_i = f_j, \text{ where } f_i \in F_M, f_j \in F_M \tag{8}$$

From the above definitions, the *MCR* can treat all of jobs when the set of the *MCR*'s functions F_R includes the set of processing methods *F* of jobs. Following in this paper, it is assumed that the relation $F \subset F_R$.

In the next section, a Petri Net is used for the *MCR* modeling, and it process is described. This model is useful to consider the behaviors of the *MCR*.

4. Modeling of MCR

In supervising the *MCR*, it is necessary to know what state the system is in and what behavior is available from a certain state. For this purpose, a modeling method for the state and behavior of the *MCR* is described using Petri Net.

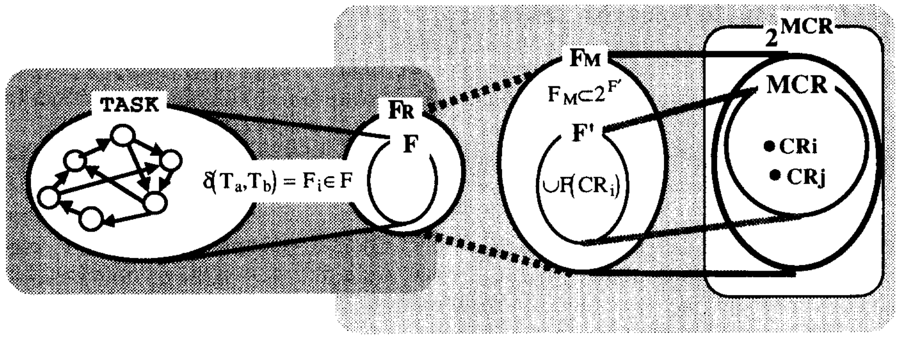


Fig.4 Relation between TASK processing functions and combinational functions of MCR

4.1 Expression of MCR by Petri Net

A Petri Net structure, PN, used for the MCR modeling is given as follows:

$PN = \langle P, T, A, M, C \rangle$
T: Transition set
P: Place set
A: Arc set
M: Marking
C: Color token

Based on this structure, the MCR modeling by PN is conducted according to the following procedures using a task network, as shown in Fig.2, with the relationship between tasks $TASK$, processing functions F, and the MCR.

1) Task is matched with P and F determined between the tasks with T. Then, PN is configurated by the input and output relationship determined from the order of $TASK$.

2) With respect to the above PN, each transition is matched with the processing functions $F_{ij} \in F_M$ (elements of $F_i \in F$); in other words, subdivision is performed into transition representing CR groups which can achieve processing. In this case, the firing time of each transition is matched with the processing time of that group.

3) Place P^{MCR} representing the halt state of each CR is added and connected to all T with respect to its $T_i \in T$.

Finally, T^{id} to delay processing between tasks per unit time is added to each P in order to establish a model for MCR. Fig.5 shows an example of this model.

The action of the model starts with preparing the specified number of color tokens $C(CR_i)$ representing each CR arranged in P^{MCR} and of tokens $C(JOB_i)$ representing jobs provided for the system. The tokens are fed into the location of the task where the job is initiated. This is referred to as the initial state of the model, and the subsequent actions of the model are expressed by either of the following methods.

(1) If there are transition T_r having $C(JOB_i)$ at $P_j \in X_i$ with respect to $JOB_i = (X_i, \leq)$ and having output at $P_j < P_{j+1} \in X_i$, and

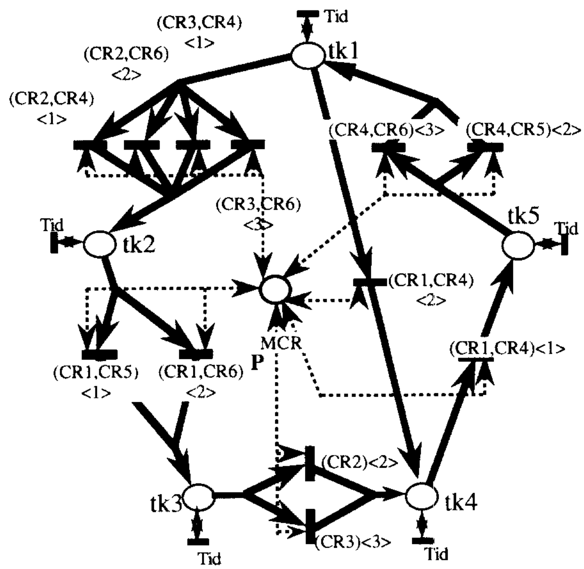


Fig.5 TASK-MCR model represened with petrinet.

if $C(CR_i)$ represented by arc attribute between the place and this T_r in P^{MCR}, then firing is available, receiving these tokens to feed $C(JOB_i)$ into the next P_{j+1} after firing time for each, and return $C(CR_i)$ to P^{MCR}. In this case, other transitions having input with respect to the same $C(JOB_i)$ and $C(CR_i)$ have a competitive relationship.

(2) T^{id} connected to P_j is fired in order to delay processing of the job.

The above modeling allows the handling of all expressions of actions for a certain JOB_i in the MCR. Therefore, the problem of how to utilize each CR group in this model can be replaced by the problem of which of the transition groups in this model is fired and when. In other words, operation is performed by searching a firing series under an appropriate evaluation function for the system. However, the search for the firing series of Petri Net becomes harder with an increase in the scale of a network until it is no longer a practical method. As one of the methods for this problem and for realizing quasi-distributed management, a stochastic learning method is added to the MCR, as described in the following section.

5. Stochastic Learning Scheme for Quasi-Distributed Management

In this section, we investigate the stochastic learning scheme for organization of the CR groups adapted to processing specific jobs. The PN model represents that the activities of CR groups on the system change dynamically according to the task processing, i.e., when the $JOB' = \{tk1, tk2\}$ will be processed, if there exits other jobs being processed by groups including $CR2$ and $CR3$, processing of JOB' must be delayed until these CRs free. Furthermore, even if the state which $CR3$ and $CR6$ can process JOB' is established, the processing intend to be delayed until the activity of $CR2$ in near future being free rather than to be started using the group, $CR3$ and $CR6$, because the processing time of the group, $CR2$ and $CR4$, is shorter than that of $CR3$ and $CR6$. Thus, the purpose of the task management is addressed that how organize CRs keeping the efficiency of the system in the dynamical task processing environment. In this section, the problem is treated in the stochastic learning scheme because the characteristic of the PN models show the difficulty of applying the analytical methods to the problem.

The scheme is based on the probabilistic selection of the groups among competitive groups related a certain job processing in the dynamical environment. The probability of the selection of groups are determined by the strengths of the groups given with the sum of strengths of robots consisting the groups. The learning scheme changes the strengths of robots according to the whole system performance. The strengths include the stack allocation in order to proceed the active delay state. Such stacks, CS, are represented as follows:

$$CS = \{\ CS_j \mid F(CS_j) = \delta(t_g, t_g)\ j=1,\cdots,l\ \} \qquad (9)$$

The strength is given each CRs including CS. The probability of the competitive groups in certain situation of the system is given as follows:

$$Prob(G_\alpha) = \frac{\sum_{CR_{\alpha j} \in G_\alpha} S(CR_{\alpha j})}{\sum_{CR_\beta \in MCR(Tk_a)} S(CR_\beta)} \qquad (10)$$

This equation isn't prohibited to take negative value. Negative value of the equation, $Prob(Ga) < 0$, is interpreted as the probability of the group being zero.

The stochastic learning is accomplished by adjusting the probability of the group according to the efficiency of the system in a specific job processing environment. That is, the strength of CR_i will be increased if the activity of CR_i contributes the efficiency of whole system performance. While the strength of the CR_j will be decreased if the activity of CR_j does not contribute the efficiency. This reinforcement scheme is expressed as follows:

$$S^k(CR_\alpha) = S^{k-1}(CR_\alpha) + \omega(\bar{\Delta} - \Delta_k)(\bar{E}(CR_\alpha) - E^{k-1}(CR_\alpha)) \qquad (11)$$

In this equation, $\bar{\Delta}$ and Δk are the average total processing time of system until k-1 period and the total processing time of system at k period respectively. $E(CRi)$ and $\bar{E}(CRi)$ are the evaluation function of activity of CR_i at k period and its average value until k-1 period respectively. w is a weighted coefficient. Here, the system performance is considered as the total processing time. In this scheme, important issue is the design of the evaluation function of CR activities. The property of the function determine the behaviors of CRs in job processing and the performance of the system adjusted by the learning scheme. The activity evaluation function is settled as follows:

$$E^k(CR_\alpha) = \rho \left(\frac{|Tk_j^\alpha|}{|CR_\beta^\alpha| + \theta} \right) \left(\frac{\Psi(CR_\alpha)}{\Delta_k} \right) \qquad (12)$$

where, $\{Tk_j^\alpha\}$ is set of tasks processed by $CR\alpha$ at k-period; $\{CR_\beta^\alpha\}$ is set of the cooperative robots combined with $CR\alpha$ at k period; $\Psi(CR_\alpha)$ is total processing time of $CR\alpha$ in k period; and ρ , θ are weighted coefficients.

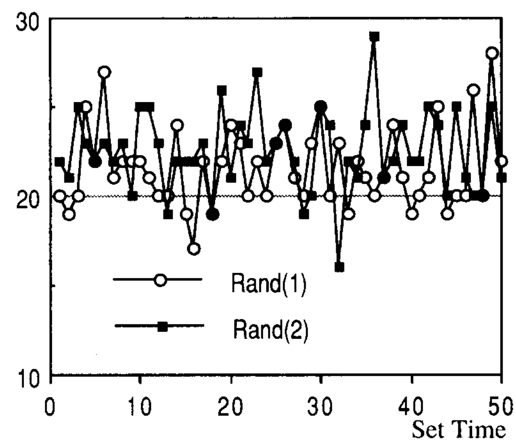


Fig.6 Results of total processing times with random selection.

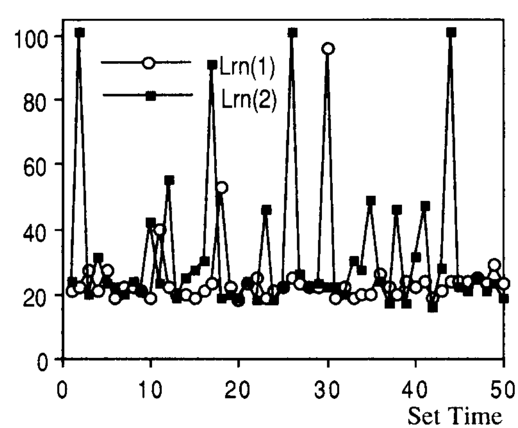


Fig.7 Results of total processing times with stochastic learning scheme.

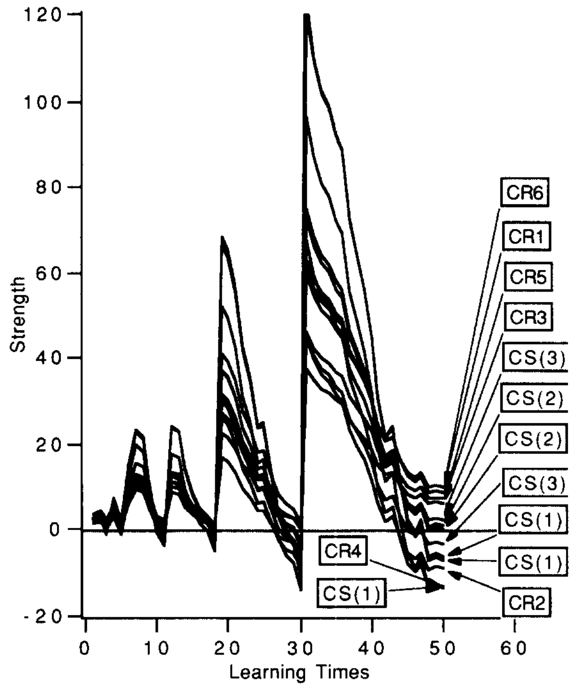


Fig.8 Process of learning strength of each Client Robots processing JOB type: (*Tk1->Tk2*)$\times$15 and (*Tk4->Tk5->Tk1*)$\times$7.

The purpose of the function searches the situation that only the limited CRs just works for processing specific jobs and keeping total performance of the system.

In next section, we show the results of experiments of the scheme.

6. Experiments of the Stochastic Learning Scheme

Components, *CRs*, of the *MCR* in the experiments are shown in Fig.5. Especially, *CS* has the 7-stack robots including: 3-robots delay the task while 1-unit time, rest of each 2-robots delay it while 2-unit time and 3-unit time respectively. We examined random selection scheme and the stochastic learning scheme in this experiments. In each experiments, 2-types jobs are applied to the *MCR*. The first type of jobs consists of 15 number jobs: *Tk1->Tk2* and 7 number jobs: *Tk4->Tk5->Tk1* in Fig.5. The second type of jobs consists of 15 number jobs: *Tk1->Tk2->Tk3->Tk4*. In each experiments, the unit period of trials is given that the system finishes the whole jobs and the performance of the system is evaluated by the processing time in one period. In the leaning scheme, the weighted coefficients are settled as follow: w=0.001, r=2.0, q=1.0.

The first experience proceeds that feasible *CRs* groups including *CS* are randomly selected to process the tasks. The result of the performance of the *MCR* is shown in Fig.6. The second experience proceeds that *CRs* are selected according to the probability adjusted by the learning scheme in each periods. The result of the experiment is shown in Fig.7. Both results are evaluated by the total processing time according to the periods. These result show the learning scheme can't give the performance absolutely optimal but keeps the average performance of the system.

The Fig.8 shown the result of the strength of each *CR* and *CS* adjusted by the learning scheme relating the first type of jobs. Since the probabilities of selecting feasible groups are determined by the sum of *CRs* strengths, the *CR* having a negative value of the strength inhibits the groups including itself selected out for task processing. Thus, the leaning scheme keeps the system performance and reduces the processing robots. This properties of the scheme is result of the design of the evaluation function of CR activities.

7. Conclusions

For autonomous and distributed robot systems that are regarded as important in realizing more flexible manufacturing systems, the *MCR* is conceived which involves nonuniform multi-functional robots as constituents and conducts varied cooperative processing. For the task processing management of this system, Petri Net was used in modeling and the stochastic leaning scheme were introduced into MCR. Through the experiments, we showed the efficiency of the adaptive strategy organizing the components robots suited for the dynamic environments of the system. This learning strategy will be basis of quasi-distributed management method: it dynamically organizes the suitable groups according to characteristics of jobs.

References:

1)N.Okino: Bionic Manufacturing Systems, JSPE-56-10, pp.76-79 (1990)

2)P.K. Wright, D.A. Broune: Manufacturing Intelligence, Addison-Wesley (1988)

3)H.Asama: Distributed Autonomous Robotic System Configurated with Multiple Agents and Its Cooperative Behavior, JSPE-57-12, pp.2112-2117, (1991)

4)K. Suzuki, Y. Kakazu: A Study of Autonomous and Distributed Robot Groups for Flexible Manufacturing System, Proc. of CAPE-91 (1991)

5)S. Mikami, Y. Kakazu: A Connectionist Approach to Autonomous and Distributed Manufacturing System, Proc. of CAPE-91 (1991)

6) H. Van Dyke Parunak: Manufacturing Experience with the Contact Net, Distributed Artificial Intelligence, PITMAN PUBLISHING (1987)