TR-H-257

Making Organizational Learning Operational: Implication from Learning Classifier System

Keiki TAKADAMA, Takao TERANO (Univ. Tsukuba), Katsunori SHIMOHARA, Koichi HORI (Univ. Tokyo) and Shinichi NAKASUKA (Univ. Tokyo)

1998.11.12

ATR人間情報通信研究所

〒619-0288　京都府相楽郡精華町光台2-2　TEL: 0774-95-1011

ATR Human Information Processing Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

Telephone: +81-774-95-1011

Fax　　　: +81-774-95-1008

Making Organizational Learning Operational: Implication from Learning Classifier System

Keiki Takadama [*1], **Takao Terano** [*2], **Katsunori Shimohara** [*1]
Koichi Hori [*3] **and Shinichi Nakasuka** [*3]

[*1] ATR Human Information Processing Research Labs.
[*2] Univ. of Tsukuba, [*3] Univ. of Tokyo

Abstract

The concepts of organizational learning in organization and management science cover a very wide range of organization-related activities in organization. Since socially situated intelligence is one of such activities, this paper makes the concept of organizational learning operational from the viewpoint of CMOT (Computational & Mathematical Organization Theory) for investigating socially situated intelligence. In particular, this paper focuses on the characteristics of multiagent learning as one of socially situated intelligence, and analyzes them with our model which introduces four operationalized learning mechanisms in organizational learning. This model is a GBML (Genetics-Based Machine Learning) based architecture, and is composed of the following four mechanisms: (a) reinforcement learning, (b) rule generation, (c) rule exchange, and (d) reuse of organizational knowledge. In this model, agents acquire their own appropriate problem solving functions through interaction with other agents in order to complete given problems. A careful investigation on the characteristics of multiagent learning with our model from the viewpoint of socially situated intelligence has revealed the following implications: (1) four learning mechanisms in our model work respectively as (a) a search function, (b) a generator of search methods, (c) an entity to change the search range, and (d) an entity to effectively limit large search ranges; (2) these four mechanisms work effectively by integrating with other mechanisms, in addition to make up for the defects of the other mechanisms. (3) besides the interaction among agents, the interaction among learning mechanisms is required to implement socially situated intelligence at a high level; and (4) there are two levels in the learning mechanisms for multiagent learning (the individual level and organizational level) and each mechanism is divided into two types (single- and double-loop learning). The integration of these various levels and types of learning mechanisms contributes to improving socially situated intelligence.

Keywords

socially situated intelligence, organizational learning, multiagent learning, learning classifier system

1 Introduction

In recent years, computational science has focused on socially situated intelligence [Agre 96, Epstein 96, Kirn 97, Prietula 98] in addition to social science. In the field of Artificial Intelligence (AI), in particular, a lot of research on multiagents has addressed the problem of making clear the intelligence embedded in multiagent environments. Examples include multiagent learning [Weiss 96, Weiss 97] based on reinforcement learning [Sutton 98, Watkins 92], multiagent evolution which is one type of evolutionary computation [Goldberg 89], and distributed artificial intelligence (DAI) [Gasser 88, Ishida 96] to study the mechanisms of social coordination and the performance improvement in organizational problem solving. In the above literature, many individuals or agents behave according to their own decisions and affect their groups or organizations as a total behavior.

However, the above research has not yet attained an explanation for socially situated intelligence, because they seem to focus only on a small part of intelligence. From this fact, this paper focuses on the characteristics of multiagent learning as one of socially situated intelligence, and analyzes them from the viewpoints of organizational learning [Argyris 78, Duncan 79, March 91, Cohen 95] in organization and management science. This is because various types or levels of intelligence are embedded in organizational learning. To accomplish the aim of this paper, we start by making the concept of organizational learning operational from the viewpoint of CMOT (Computational & Mathematical Organization Theory).

This paper is organized as follows. Section 2 starts by mentioning the organizational learning in organization and management science and Section 3 explains our computational model which introduces the concept of organizational learning. An example for analyzing embedded intelligence is given in Section 4, and Section 5 gives simulations and experimental results. Socially situated intelligence in multiagent learning is discussed in Section 6. Finally, the conclusion is given in Section 7.

2 Organizational Learning and Computational Analysis

2.1 Definition of organizational learning

Research on organizational learning has developed in the context of organization and management science, and a lot of research has focused on economical market systems or human organizations [Argyris 78, Duncan 79, Espejo 96, March 91, Cohen 95]. In organization and management science, organizational learning is roughly characterized as organizational activities for improving the organizational performance or the ability to solve problems which cannot be achieved at an individual level.

However, the features of organizational learning somewhat differ from researcher to researcher, and consequently the definition of organizational learning has become too general for our study. As typical definitions, the following information is well-known: "Organizational learning occurs when members of the organization act as learning agents for the organization, responding to changing the internal and external environments of organization by detecting and correcting errors in organizational theory-in-use [Argyris 78] " or "Organizational learning is defined as the process within the organization by which knowledge about action-outcome relationships and the effect of the environment on these relationships is developed [Duncan 79]".

2.2 Four-loop learning in organizational learning

Since the definition of organizational learning is too general to analyze socially situated intelligence, this paper makes it operational from the viewpoint of CMOT. To do this, this paper starts by selecting and explaining Kim's model as one of the organizational models. This is because his model classifies the learning in detail, compared with the others. According to Kim, organizational learning is composed of the following four kinds of learning [Kim 93]:

- **Individual single-loop learning:**
  Through this learning, an individual improves his/her performance not by changing the contents or the amount of individual knowledge, but by utilizing them. This learning contributes to improving the problem solving efficiency.

- **Individual double-loop learning:**
  Through this learning, an individual extends the range of problem solving by creating and utilizing individual knowledge which includes Macro/Meta knowledge. As an operation, the modification or deletion of knowledge is also included in addition to the creation of knowledge.

- **Organizational single-loop learning:**
  Through this learning, individuals improve their performance not by changing the contents or the amount of total individual knowledge in the organization, but by utilizing them with other individuals. Like the individual single-loop learning, this learning contributes to improving the problem solving efficiency.

- **Organizational double-loop learning:**
  Through this learning, individuals extend the range of problem solving as a whole organization by creating and utilizing organizational knowledge which is shared by all of the individuals. Like the individual double-loop learning, the modification or deletion of knowledge is also included in addition to the creation of knowledge.

From the above definition, four-loop learning contributes to improving the ability of problem solving; this means (1) an improvement in the problem solving efficiency and (2) an extension in the range of problem solving. Furthermore, the above definition mentions that (1) there are individual and organization levels in the learning and (2) each learning can be classified in terms of single type or double type.

2.3 Loop leaning in computational organizational learning

Although the four-loop learning in Kim's model is classified in detail as we mentioned in the previous section, each loop learning is not fully computational from the standpoint of CMOT. Therefore, this paper reinterprets each loop learning in Kim's model by defining it as follows, from the viewpoint of distributed artificial intelligence (DAI) or multiagent systems (MAS) [†].

[†]The investigation in the reinterpretation on the concept of Kim's model including loop learning will be published elsewhere.

- **Learning by using individual knowledge: Corresponding to individual single-learning**
  Through this learning, an individual solves given problems by learning how to use individual knowledge which is given or stored in advance. Although the individual interacts with its environment, this learning is performed at an individual level. One example of this learning is order acquisition by using individual knowledge.

- **Learning in the creation/utilization of individual knowledge: Corresponding to individual double-learning**
  Through this learning, an individual solves given problems by creating and utilizing individual knowledge. Although the individual interacts with its environment in the same way as the learning by using individual knowledge, the learning is performed at an individual level. One example of this learning is the creation of new individual knowledge to solve problems which are unable to be solved by stored knowledge.

- **Learning in the exchange of individual knowledge: Corresponding to organizational single-learning**
  Through this learning, individuals solve given problems which are unable to be solved at an individual level by exchanging their individual knowledge with other individuals. In this learning, the individuals interact not only with their environment but also with other individuals. One example of this learning is a task assignment based on the characteristics of partner individuals.

- **Learning in the creation/utilization of organizational knowledge: Corresponding to organizational double-learning**
  Through this learning, individuals solve given problems by creating and utilizing organizational knowledge (which means knowledge at an organizational level). In this learning, individuals interact not only with their environment but also with other individuals in the same way as the above learning in the exchange of individual knowledge. One example of this learning is shown as follows: individuals store the integration of selected individual knowledge as organizational knowledge when they solve given problems most effectively, and utilize it in the subsequent problems. This kind of knowledge seems to be knowledge on the division of work.

Using the above definitions, we define *computational organizational learning*, *individual knowledge*, and *organizational knowledge* as follows.

- **Computational organizational learning:**
  Learning that includes the above four learning mechanisms, *e.g.*, learning by using individual knowledge, learning in the creation/utilization of individual knowledge, learning in the exchange of individual knowledge, and learning in the creation/utilization of organizational knowledge.

- **Individual knowledge:**
  Individual level knowledge stored in each individual independently.

- **Organizational knowledge:**
  Organizational level knowledge shared by all individuals.

The definitions may seem to define only parts of organizational learning, but the authors believe that the above definitions are sufficient for analyzing socially situated intelligence from the viewpoint of CMOT. What we claim in this paper is that our attempt is to analyze socially situated intelligence with operationalized organizational learning.

3 Organizational-Learning Oriented Classifier System

3.1 Architecture

Our computational model (Organizational-learning oriented Classifier System: OCS) [Takadama 98a, Takadama 98b] was originally developed to apply complex engineering problems in multiagent environments. In this paper, however, OCS is used for investigating socially situated intelligence. As shown in Fig. 1, OCS introduces four learning mechanisms defined in the previous section into a learning classifier system (LCS) [Goldberg 89, Holland 78], and is composed of a lot of agents. Each agent in OCS has the following problem solver, memory and learning mechanisms. As an assumption, each agent can recognize its environmental state, but it cannot recognize the total environmental state. This assumption reflects the situation in which it becomes difficult to acquire the appropriate global information as the number of agents increases.

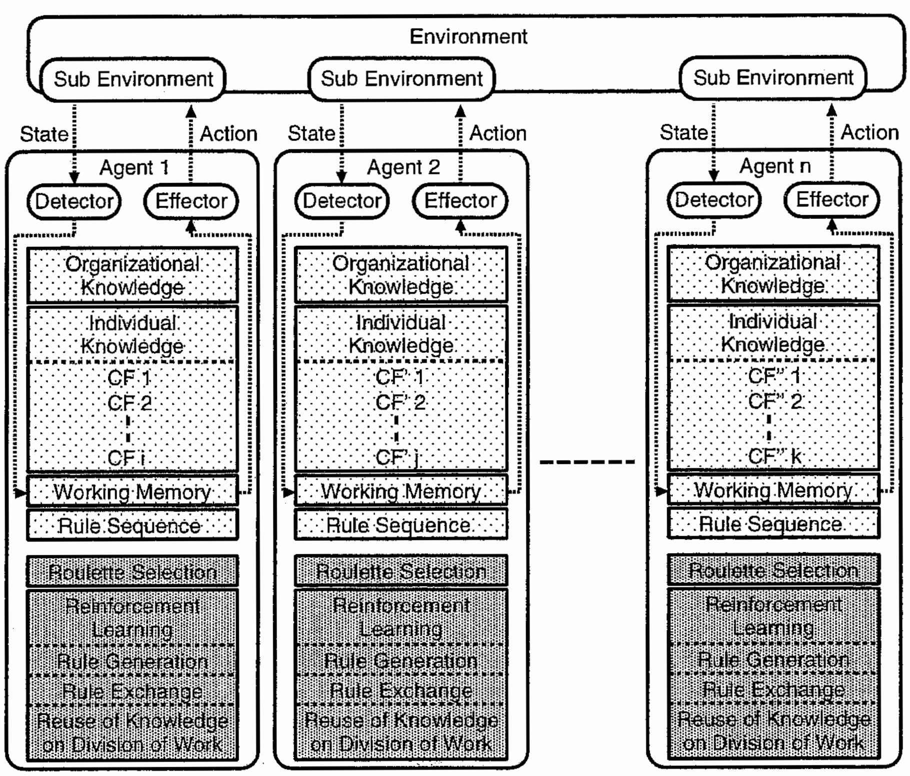

Figure 1: **Architecture of OCS**

- **Problem Solver**
  - **Detector and Effector:**
    The detector changes a part of an environmental state into an internal state and the effector changes an internal state into an action [Russell 95].

- Memory
  - **Organizational knowledge memory:**
    This memory is for storing organizational level knowledge to be shared by all agents. In OCS, the agents store knowledge on the division of work, and this knowledge is implemented by a set comprising each agent's rule set acquired when the agents solve given problems most effectively.
  - **Individual knowledge memory:**
    This memory is for storing a set of CFs (classifiers). In OCS, CF is an if-then rule with a strength factor, *i.e.*, the worth of rules, and the production system operates with these CFs.
  - **Working memory:**
    This memory is for storing the results obtained in recognizing a part of environment states and an internal state of an action of fired rules.
  - **Rule sequence memory:**
    This memory is for storing a sequence of fired rules in order to evaluate them. This memory is cleared after the evaluation.
- Learning Mechanism
  - **Roulette selection:**
    This mechanism selects one rule from among plural rules matching a particular situation. In this selection, the one rule is selected probabilistically according to the size of the strength attached to each rule.
  - **Reinforcement learning mechanism:**
    This mechanism is performed as the learning by using individual knowledge described in section 2.3. As a basic mechanism, this mechanism evaluates all rules fired and changes the strength of the rules according to an evaluation when agents solve given problems.
  - **Rule generation mechanism:**
    This mechanism is performed as the learning in the creation/utilization of individual knowledge described in section 2.3. As a basic mechanism, this mechanism creates a new rule when all stored rules in agents do not match a current environmental state.
  - **Rule exchange mechanism:**
    This mechanism is performed as the learning in the exchange of individual knowledge described in section 2.3. As a basic mechanism, this mechanism enables agents to exchange their rules with other agents in a particular interval of time.
  - **Organizational knowledge reuse mechanism:**
    This mechanism is performed as the learning in the creation/utilization of organizational knowledge described in section 2.3. As a basic mechanism, this mechanism enables agents to reuse the knowledge on the division of work by utilizing a set comprising each agent's rule set as initial rules sets before other problems are solved.

3.2 Aim of agent and function

Each agent in OCS cooperates with other agents to solve problems that cannot be solved at an individual level. To do this, agents try to divide given problems by acquiring their appropriate functions through interaction with other agents. In this approach for solving problems, we define the aim of agents in OCS as finding appropriate functions required to solve problems by dividing them. Furthermore, we define a function as a sequence of behaviors determined by if-then rules. For example, $A \to B \to C$ and $C \to B \to C \to A$ are some of the functions used when A, B, C are assumed as one behavior.

This definition means that the learning for acquiring appropriate functions in some agents is affected by the function acquisition of other agents. For example, some agents are affected when one of the A, B, or C behaviors of other agents changes to either the D, E, or F behavior through learning, or when the fired order of the $A, B, andC$ behaviors of other agents changes according to the change of the rule strength.

3.3 Learning in OCS

This section describes how the four learning mechanisms in computational organizational learning mentioned in section 2.3 are implemented in OCS.

3.3.1 Reinforcement learning mechanism

In OCS, agents have a reinforcement learning mechanism which is the same as LCS, and acquire appropriate behaviors which cooperate with other agents through this mechanism. Although this mechanism does not contribute to creating rules themselves, it does enable agents to change the order of the fired rules by changing the strength of the rules. From this fact, this learning mechanism corresponds to "individual single-loop learning" in organization and management science, and works as a kind of "learning by using individual knowledge" in computational organizational learning.

In the context of reinforcement learning, OCS employs *profit sharing* [Grefenstette 88] which reinforces a sequence of rules at once when agents obtain some rewards. As a concrete mechanism in OCS, positive rewards are distributed to all fired rules as shown in Fig. 2 , and the strength of each rule is calculated according to Eq. (1). After the rewards are distributed, the memory for storing a sequence of fired rules is cleared.

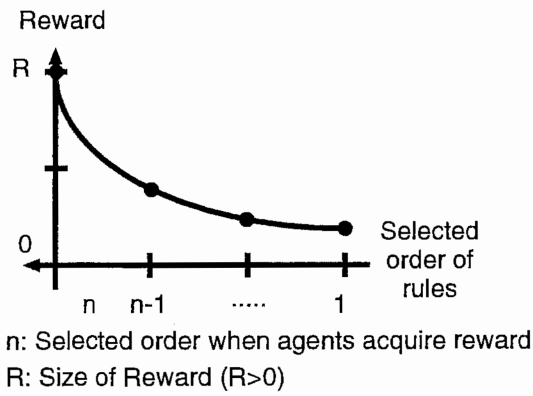

Figure 2: **Reinforcement leaning mechanism**

$$ST(i) = ST(i) + R \cdot G^{n-i}, \quad where\ i = n, n-1, \cdots, 1 \tag{1}$$

In Fig. 2, the vertical and horizontal axes indicate the size of the reward and the fired order of rules, respectively. Note that the rules presented on the right side are fired at the first several selections. Furthermore, ST in the equation represents the strength of the rule, i represents the order of the fired rules, n represents the maximum number of fired rules, R (> 0) represents the size of the reward, and G represents the geometric ratio with the range of $0 < G < 1$. Concerning i, a small i indicates the first several stages in the rule selection.

3.3.2 Rule generation mechanism

Agents in OCS create new rules when all of the stored rules do not match the current environmental state (which indicates a situation in which agents encounter a new environment). Since this mechanism enables the agents to solve given problem not by utilizing acquired rules but by changing the contents or the quantities of rules themselves, the learning in this mechanism corresponds to "individual double-loop learning" in organization and management science, and works as a kind of "learning in the creation/utilization of individual knowledge" in computational organizational learning.

However, it is not realistic to create new rules without limit through this mechanism. To overcome this problem, the rule with the lowest strength is removed and a new rule is generated when the number of rules is more than `MAX_CF`, which defines the maximum number of rules. In particular, when the situation does not change because the same rules are selected, the strength of the rules is decreased temporarily and these rules become candidates for a replacement with new rules. Through this mechanism, agents can adapt to changes of their environments. In the rule generation, the condition (if) part of a rule is created to reflect the current situation, and the action (then) part is determined at random. Moreover, the strength values of the rules are set to the same initial value.

3.3.3 Rule exchange mechanism

In OCS, agents exchange rules with other agents at a particular interval of time. Since this mechanism contributes to cooperation among the agents and improves the problem solving efficiency not by changing the contents or the quantities of the rules stored in the organization but by utilizing knowledge through the local interaction among the agents, the learning in this mechanism corresponds to "organizational single-loop learning" in organization and management science, and works as a kind of "learning in the exchange of individual knowledge" in computational organizational learning.

As a concrete algorithm, rules with low strength values are replaced with the rules with high strength values between two arbitrary agents according to the following mechanism. This rule exchange mechanism works as a crossover operation, but it differs form the elite selection principle in conventional evolutionary approaches (which selects one elite population among large populations in this case). In the following mechanism, `CROSSOVER_TIME`, `CROSSOVER_NUM`, and `BORDER_ST` are respectively defined as the interval steps for crossover operations, the number of replaced rules, and the rule strength which decides whether or not the rule must be replaced.

- At `CROSSOVER_TIME` step, two agents are selected at random, and the `CROSSOVER_NUM` rules are replaced. For example, when agent X and Y are selected as shown in Fig. 3, rules with low strength values in agent X and Y are replaced with rules with high strength values in agent Y and X ($CF_1 \sim CF_3$ and $CF'_1 \sim CF'_3$ are replaced, respectively, with $CF'_{k-2} \sim CF'_k$ and $CF_{j-2} \sim CF_j$ in this case). This method not only prevents agents from getting into deadlock situations, but also improves the organizational performance by propagating effective rules among the agents.

- After the crossover operations, the strength values of the replaced rules are reset to their initial values. This reflects the situation in which effective rules in some agents are not always effective for other agents in multiagent environments.
- In order to avoid unnecessary crossover operations, the rules are not replaced when their strength values are higher than BORDER_ST. This method contributes to a quick division of work for solving given problems.

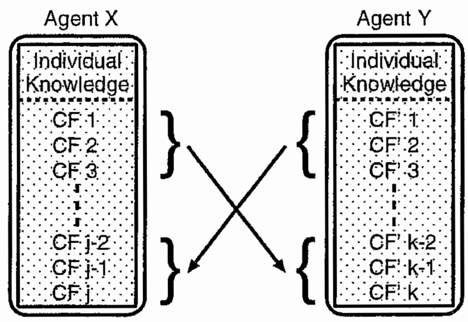

Figure 3: **Rule exchange mechanism**

3.3.4 Organizational knowledge reuse mechanism

Finally, agents store the knowledge on the division of work through the cycle of solving given problems and reuses the knowledge when other problems are provided. Since this mechanism contributes to extending the range of problem solving by creating and utilizing the knowledge on the division of work, the learning in this mechanism corresponds to "organizational double-loop learning" in organization and management science, and works as a kind of "learning in the creation/utilization of organizational knowledge" in computational organizational learning.

Here, we define organizational knowledge in OCS as knowledge on the division of work, and it is implemented by a set comprising each agent's rule set (individual knowledge). For example, when we assume RS(x) as the rule set for the x-th agent, one unit of organizational knowledge is represented by {RS (1), RS (2), $\cdots$ RS (X)}. Concerning this knowledge, the memory of all agents contains the same knowledge in the first stage of OCS.

As a concrete algorithm, agents store the rule sets of all agents when they solve given problems most effectively (*e.g.*, with the minimum number of steps). For example, when we assume that problems are solved by x number of agents most effectively with using their rule sets, a set comprising each agent's rule set which leads some functions (*e.g.*, $A \rightarrow B \rightarrow C \rightarrow B \rightarrow C \cdots$) as shown in Fig. 4 is stored. This kind of knowledge contributes both to reducing the number of iterations and to solving given problems which cannot be solved without this knowledge.

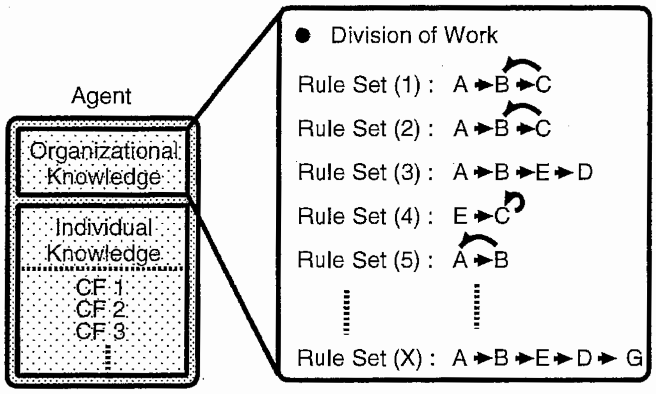

Figure 4: **Organizational knowledge reuse mechanism**

3.4 Supplemental design principles

In addition to the above mechanisms, OCS has the following principles.

- Agents behave according to the behavior in the action part of the selected rule.
- At the beginning, the `FIRST_CF` rules in each agent are generated at random, and the strength values of all rules are set to the same value. In this case, `FIRST_CF` is defined as the number of rules provided to each agent at the beginning.

4 Printed Circuit Board Design Problem

4.1 Problem Description

Since printed circuit board (PCB) problems are familiarity to the multiagent domain, we apply OCS to such a problem for analyzing the characteristics of multiagent learning as one of socially situated intelligence. The goal of this problem is to appropriately place all parts whose total wiring length is short. As an assumption, circuit diagrams are given, and the types and the number of parts are also given in advance.

In this task domain, the parts are designed as agents in OCS, and have 10 kinds of primitive behaviors as shown in Table. 1. Using these behaviors, the parts aim to acquire their own functions (a sequence of behaviors) to minimize the total wiring length through local interaction among parts. As an assumption, the parts can recognize overlapping areas and the local wiring length, but they do not know the total wiring length and their effective behaviors beforehand.

4.2 Index of Evaluation

In this PCB design, the parts continue to acquire their functions to minimize the total wiring length from the initial placement when all parts are placed. In this cycle, we count one *step* when all parts perform one behavior as shown in Table 1, and count one *iteration* when the parts are all placed. Furthermore, the accumulated steps are defined as $\sum_{i=1}^{max\_iteration} step\ (i)$ and are assumed as the computational complexity. In this equation, i, $step\ (i)$, and $max\_iteration$ indicate respectively the iterations, the steps of i iterations, and the iterations when the values of the steps converge.

Table 1: **Primitive behaviors in parts**

Primitive Action		Symbol
Stay	[1 kind]	`a`
Movements for reducing overlapping areas	[4 kinds]	`b,c,d,e`
Movements for minimizing the local wiring length	[1 kind]	`f`
Rotation	[3 kinds]	`g,h,i`
Transference (Large movements)	[1 kind]	`j`

5 Simulation

5.1 Contents of experiment

In the simulation, the effect of the following each mechanism and the integration of all of the mechanisms are investigated.

- **Reinforcement learning mechanism:** Corresponding to individual single-learning
- **Rule generation mechanism:** Corresponding to individual double-learning
- **Rule exchange mechanism:** Corresponding to organizational single-learning
- **Organizational knowledge reuse mechanism:** Corresponding to organizational double-learning

Actually, the following 15 cases are tested with PCB design problems mentioned in the previous section, and the differences in the 15 cases are compared. In the experiment, a PCB design problem comprising 92 parts is addressed. Concretely, 40 parts are added to the original 52 parts, and the types of parts include a CPU, condenser, register, jumper, and so on. Using real CAD data, we evaluate (1) the total wiring length, (2) the steps, (3) the iterations and (4) the accumulated steps.

- **Cases 1, 2, 3, 4** : R, G, X, K
- **Cases 5, 6, 7, 8, 9, 10** : RG, RX, RK, GX, GK, XK
- **Cases 11, 12, 13, 14** : RGX, RGK, RXK, GXK
- **Cases 15** : RGXK

In the above cases, R indicates a **R**einforcement learning mechanism, G indicates a rule **G**eneration mechanism, X indicates a rule e**X**change mechanism, and K indicates an Organizational **K**nowledge reuse mechanism. For example, RGXK indicates the case that all four of the mechanisms are included.

5.2 Setting for experiment

When we apply OCS to PCB design problems, the learning mechanisms in OCS are designed as follows.

- **Reinforcement learning mechanism:**
  Rewards are distributed to a sequence of fired rules in each part according to the reinforcement learning function as shown in Fig. 2 or Eq. (1) when all parts are placed, and the parts restart in finding the appropriate placement using their rules continuously.

- **Rule generation/exchange mechanism:**
  When parts cannot remove overlapping areas or cannot make appropriate spaces among parts due to the current selected rules, the strength of these rules is decreased temporarily and these rules are removed in the rule generation/exchange mechanism.

- **Organizational knowledge reuse mechanism:**
  Organizational knowledge implemented by a set comprising each agent's rule set is prepared using a small PCB size in advance, and is reused as an initial rule set of parts in the PCB design with 92 parts. In this case, the small PCB size is composed of 45 parts and its layout is designed by adding 20 parts on to the 25 original parts. When we assume RS_y(x) as the rule set of the x-th agent which designs the PCB with y parts, the way of reuse is described as follows.

$$RS_{92}(x) \leftarrow RS_{45}(mod((x-1),45)+1), \quad x = 1,\cdots,92$$

As another setting for this experiment, the variables of OCS are set as follows: `FIRST_CF` (the number of initial rules of each agent) is 50, `MAX_CF` (maximum number of rules) is 100, `CROSSOVER_TIME` (the interval steps for crossover operations) is 100, `CROSSOVER_NUM` (the number of replaced rules) is 5, `BORDER_ST` (the lowest strength of the rule not for removal) is -50.0, R (the size of the reward) is 1, and G (the geometric ratio) is 0.5.

5.3 Experimental results

Table 2 shows the results of PCB re-design in terms of (1) the total wiring length, (2) the average steps which are calculated until the value of the wiring length converges, (3) the iterations which are counted until the value of the wiring length converges and (4) the accumulated steps (= (2)×(3)).

In this table, the attributes of the horizontal and vertical axes indicate the learning at an organizational level and individual level, respectively. Furthermore, "—" in the left-top side box indicates that there is no experiment, and "×" in each box indicates that the agents cannot solve the given problems. The number in the first (upper), second, third, and fourth (lower) line of each box indicate the total wiring length, the average steps, the iteration counts and the accumulated steps, respectively. Moreover, R, G, X, and K indicate each of the learning mechanisms mentioned in section 5.1, and the four types of arrows indicate the effects of the four learning mechanisms with the change of a numerical value. From this table, we can make the following remarks.

Table 2: **Total wiring length (first line), Average steps (second line), iteration counts (third line) and Accumulated steps (fourth line)**

Organizational Loop Learning / Individual Loop Learning	None	Rule Exchange (X)	Knowledge Reuse (K)	Rule Exchange & Knowledge Reuse (X)+(K)
None	—	×	×	×
Reinforcement Learning (R)	×	×	×	×
Rule Generation (G)	G 24858 580 12 6960	GX 24208 423 25 10575	GK 31218 2752 27 74304	GXK 24616 130 19 2470
Reinforcement Learning & Rule Generation (R)+(G)	RG 24503 750 128 96000	RGX 23663 626 89 55714	RGK 25490 820 24 19680	RGXK 23206 138 38 5244

- - -→ Effect of Reinforcement Learning
—·—→ Effect of Rule Generation
——→ Effect of Rule Exchange
·········→ Effect of Organizational Knowledge Reuse

- **Reinforcement learning mechanism:**
  Reinforcement learning minimizes the total wiring length, but increases both the steps and the iteration counts. From this factor, this mechanism contributes to finding an effective solution, but it leads to a large computational complexity.

- **Rule generation mechanism:**
  Parts are not able to be placed without a rule generation mechanism. This implies that this mechanism is indispensable when necessary and appropriate rules are not known beforehand.

- **Rule exchange mechanism:**
  Rule exchange mechanism not only minimizes the total wiring length but also reduces the steps. Since a big change (increase or decrease) in the iteration counts is not found, this mechanism contributes to finding effective solutions with small average steps.

- **Organizational knowledge reuse mechanism:**
  Although organizational knowledge reuse mechanism reduces the iteration counts, there is no notable tendency in the total wiring length or the average steps. However, this mechanism finds the minimum total wiring length with reducing both the steps and the iterations by integrating other mechanisms in OCS.

6 Discussion

6.1 Analysis

In order to investigate socially situated intelligence, this section analyzes the simulation results, and examines the characteristics of each learning mechanism in computational organizational learning and the effect of the integration of these four learning mechanisms.

In the case that all parts are placed on a PCB, we investigate the change of the functions (a sequence of behaviors) of one part in every iteration, and show the change as in Table. 3. In this table, the 10 symbols (a $\sim$ j) indicate the primitive behaviors mentioned in section 4, and one function is composed of the combination of these 10 behaviors. Especially in the function, the behaviors on the left side indicate behaviors selected in the first several steps, and the behaviors on the right side, in constant, indicate behaviors selected in the last several steps. Furthermore, "( )" indicates the first behavior, which is different from the behavior in the previously acquired function. R, G, X and K indicate each of the learning mechanisms mentioned in section 5.1.

Since the tendency of the other parts is almost the same as shown in Table 3, this section analyzes the results in one part. From this table, we make the following remarks.

- **Effect of the rule generation mechanism**
  - **Function change: G**
    A part with the rule generation mechanism (represented as G) creates new rules when the part encounters a new environment and has no rules matching the current environmental state. Due to this mechanism, the location of the function change is shifted to the right side as shown in Table 3–1. This means that parts cope with the current situation with their own rules and create new rules only when they cannot cope with the current situation by their own rules.

    From this result, we find that this mechanism enables parts to adapt to the changes in their environments by creating necessary rules only when they are needed. Since the variety of behaviors increases by the creation of new behaviors, this mechanism works as a generator of search methods or works as a generator of search operators from the viewpoint of search.
  - **Total wiring length and computational complexity: G**
    Since rules are not created frequently, solutions (*i.e.*, the total wiring length in this simulation) by the rule generation mechanism converge in small iterations. However, this mechanism does not alway create appropriate rules. Due to this fact, the total wiring length may become long and the average steps may become large. To support the understanding of the reason for this, let us assume a situation in which a lot of parts overlap each other as shown in Fig. 5 (a). If the grey shaded part creates a rule for rotation when it has no rules for this situation, this rule is selected continuously as show in Fig. 5 (b). However, the situation does not change drastically, and the part cannot reduce the overlapping area by itself.

    Since the total wiring length may become long and the steps needed for reducing overlapping area may increase in the case of a high density of parts, the transference as shown in Fig. 5 (b) or the movement as shown in Fig.

Table 3: **Acquired function in every iteration**

Table 3–1

Iteration	Acquired function by G mechanism
1	bgdgbcfgcbccdcbgfificcific
2	bgdgb(f)ccfificdcbdcbfcci
3	bgdgbfc(f)cbcgdgccifcfcifbd
4	bgdgbfcfcbc(d)cicbifccfbdici
5	bgdgbfcfcbcdcicb(c)iifccfb
6	bgdgbfcfcbcdcicb(d)fcbciffciic
7	bgdgbfcfcbcdcicbdfcbc(c)ifci
8	bgdgbfcfcbcdcicbdfcbc(e)icdifi
9	bgdgbfcfcbcdcicbdfcbc(c)ific
10	bgdgbfcfcbcdcicbdfcbccifi(g)igd
⋮	⋮

Table 3–2

Iteration	Acquired function by RG mechanism
1	bgdgbcfgcbccdcbgfificcific
2	bgdgb(f)ccfificdcbdcbfcci
3	bgdgbfc(f)cbcgdgccifcfcfi
4	bgdgbfcfcbc(i)ccbbdfcccifci
5	bgdgbfcfcbcicc(c)cidiffcfcgdi
6	bgdgbfcfcbcicccci(f)fdfcc
7	bgdgbfcfcbcic(d)cgjbbhdificcfcbgdbib
8	bgdgbfcfcbcicdcgjb(h)cifificcccf
9	bgdgbfcfcbcicd(g)bcdfcfiic
10	bgdgbfcfcbcicdgb(f)cciccbcfif
11	bgdgbfcfcbc(c)dgbdcdcbcfficfi
12	bgdgbfcfcbccdgb(c)cifccgdiff
13	bgdgbfcfcbccdgbccifc(d)cgdiff
14	bgdgbfcfcbccdgb(d)ccifccffibi
15	bgdgbfcfcbccdgbdc(i)fccfibficjbhed
⋮	⋮

Table 3–3

Iteration	Acquired function by GX mechanism
1	bgdgbcfgcbccdcbgfificcf
2	bgdgb(i)fcbcdfgbccfciccfgdciif
3	bgdgbifc(c)dccbbficfibc
4	bgdgbifc(b)cbcicfigcdfci
5	bgdgbifcbc(d)cficgcicicf
6	bgdgbifcbcdcf(g)ccbdficcibgi
7	bg(b)ccdcbficbggcdifcfci
8	bgbcc(b)dcdcbdeegfifcgcifcfci
9	bgbccbdcdcbd(f)cificicif
10	bgbccbdc(i)cfcfidgficcbigd
⋮	⋮

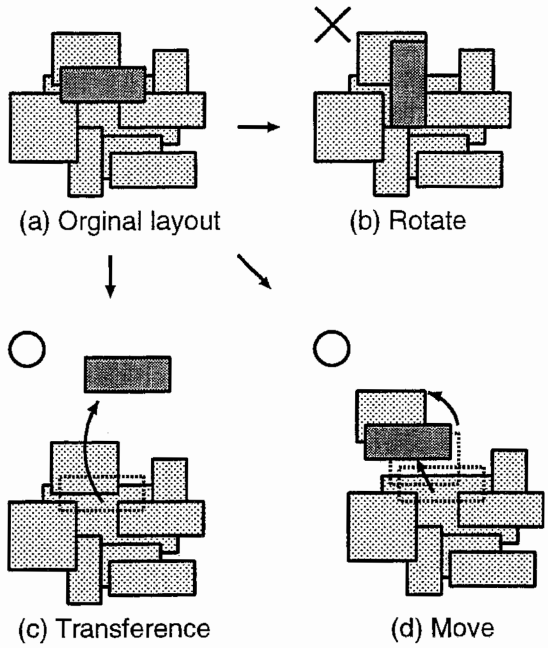

Figure 5: **Behavior when part density is high**

5 (d) is preferred. Especially in the former case, the transference contributes to reducing both the total wiring length and the average steps, because this behavior enables parts to jump to areas where the part density is low. In the latter case, on the other hand, the movement for making other overlapped parts move to the other side contributes to reducing the average steps, because the overlapping areas are removed quickly.

- **Effect of the reinforcement learning mechanism**
  - **Function change: RG**
    In the case of integrating the rule generation mechanism and reinforcement learning mechanism (represented as R), the function changes around the middle location, as shown in Table 3–2. This indicates that it is necessary for parts to learn to acquire new functions continuously because the behaviors of some parts affect the learning of other parts in multiagent learning. Therefore, the function of a part changes again and again in Tab. 3–2. From the viewpoint of search, this mechanism works as a entity that finds effective solutions or works as the learning by using operators.
  - **Total wiring length and computational complexity: RG**
    Since the functions of each part change in every cycle of learning when the reinforcement learning mechanism is introduced, the iterations counted until the solution converges become large. However, the search area is extended and the parts have a chance to acquire the transference or the movement instead of selecting the rotation as shown in Fig. 5, even in the case of a high density of parts This kind of learning contributes to minimizing the total wiring length.
- **Effect of the rule exchange mechanism**
  - **Function change: GX**
    In the case of integrating the rule generation mechanism and rule exchange

mechanism (represented as X), the function changes at the left side location (which means that the behavior changes in early steps), as shown in Table 3–3. This indicates that the rules that cannot contribute to reducing overlapping areas like the rotation shown in Fig. 5 (b) are removed and new rules are introduced from other parts through the rule exchange mechanism, and also indicates that these rules are selected in the first several selections because the density of parts is high especially in the first steps.

From the viewpoint of search, this mechanism works as an entity to change the search range or works as an entity to change search operators. This kind of mechanism enables parts to get out of deadlock situations or local minimum solutions, and also contributes to shifting the direction for finding effective solutions by removing ineffective rules.

- **Total wiring length and computational complexity: GX**
  In the case of introducing the rule exchange mechanism, the rules are exchanged continuously until the overlapping areas are removed by acquiring the transference or the movement instead of selecting the rotation as shown in Fig. 5, especially when the density of parts is high. Due to this mechanism, the total wiring length becomes short and the average steps become small.

- **Effect of the organizational knowledge reuse mechanism**
  - **Total wiring length and computational complexity: GK, RGK, GXK, RGXK**
    By introducing the organizational knowledge reuse mechanism (represented as K), the iterations become small as shown in Table 2. This result suggests that organizational knowledge on the division of work is general and it contributes to utilizing the effective characteristics. From the viewpoint of search, this mechanism works as an entity to effectively limit large search ranges or works as the creation/utilization of macro operators.

- **Effect of Integration of four learning mechanisms**
  From the results, we have found that all of the three indexes (which mean the total wiring length, the average steps, and the iterations) in RGXK which introduce all of the above mentioned four mechanisms become short or small. This is because the four learning mechanisms in OCS work respectively as (a) a search function, (b) a generator of search methods, (c) an entity to change the search range, and (d) an entity to effectively limit large search ranges, and because each mechanism works in different dimensions. However, this feature also indicates that the solution or the computational complexity ($= averagesteps \times iterations$) becomes worse when one of the mechanisms is missing.

  From this factor, the four mechanisms in OCS work effectively by integrating with the other mechanisms, in addition to make up for the defects of the other mechanisms. In particular, the effective knowledge is utilized by *exploiting* the characteristics of a search space through the rule exchange mechanism and the organizational knowledge reuse mechanism, and the ineffective knowledge is modified/removed by *exploring* another search space through the reinforcement learning mechanism and the rule generation mechanism. Furthermore, this factor suggests that iterations among various levels or types of learning mechanisms in addition to iterations among agents contribute to improving the collective performance and the ability of the problem solving.

6.2 Socially situated intelligence and computational organizational learning

Through CMOT based simulation using OCS and the analysis mentioned in the previous section, we find that the following results contribute to implementing or improving socially situated intelligence.

- First, the architecture, in which a lot of agents behave according to their decision making and are affected through local interaction with other agents, needs to implement socially situated intelligence. This means that we have to design the organization as a group of a lot of agents, instead of designing it as one agent. OCS is satisfied with this condition.

- Second, the integration of the four learning mechanisms in computational organizational learning contributes to improving socially situated intelligence. This indicates that iterations among learning mechanisms in addition to iterations among agents are required to implement socially situated intelligence at a high level.

- Third, there are two levels in learning mechanisms in terms of the level between individual and organizational loop learning, and each mechanism is divided into two types in terms of the level between single- and double-loop learning. Since the collective performance and the ability of problem solving become worse when one of the learning mechanisms is missing, the integration of various levels and types of learning mechanisms contributes to improving socially situated intelligence.

What is important to be mentioned here is that these results acquired by an analysis on operationalized organizational learning can be applied to a lot of systems with multiagent architectures and can contribute to engineering and social science in terms of implementing social situated intelligence at a high level.

However, some issues to be uncovered remain in our OCS framework. First, this paper omits any investigation on the size effect, which depends on the combination of various levels or types of learning mechanisms. From this factor, the next topic of research has to focus on analyzing other levels or types of learning mechanisms and also on the effects of integrating these mechanisms. Second, we have only analyzed socially situated intelligence in terms of the improvement of both the collective performance and the ability of problem solving. This does not imply that there are no possibilities to other CMOT approaches to socially situated intelligence. Third, each leaning mechanism in this paper might be designed in a different way and other computational definitions of organizational learning might be made. Fourth, the relationship between the organizational structure and socially situated intelligence has to be analyzed, and the effect of knowledge creation [Nonaka 95] which able to cope with explicit and tacit knowledge should be investigated from the viewpoint of socially situated intelligence.

7 Conclusion

This paper makes the concept of organizational learning operational from the viewpoints of CMOT, and analyzes characteristics of multiagent leaning as one of socially situated intelligence with four operationalized learning mechanisms in organizational learning. The main results are summarized as follows: (1) four learning mechanisms in our model work respectively as (a) a search function, (b) a generator of search methods, (c) an entity to change the search range, and (d) an entity to effectively limit large search ranges; (2) these four mechanisms work effectively by integrating with other mechanisms, in addition to make up for the defects of the other mechanisms. (3) besides the interaction among agents, the interaction among learning mechanisms is required to implement socially situated intelligence at a high level; and (4) there are two levels in the learning mechanisms for multiagent learning (the individual level and organizational level) and each mechanism is divided into two types (single- and double-loop learning). The integration of these various levels and types of learning mechanisms contributes towards improving socially situated intelligence.

Future research includes the following.

- An analysis of socially situated intelligence with other CMOT approaches.
- An investigation on socially situated intelligence with other designs of learning mechanisms or with other computational definitions of organizational learning.
- An investigation on the relationship between the organizational structure and socially situated intelligence
- An analysis of the effect of knowledge creation from the viewpoint of socially situated intelligence.

References

[Agre 96] P.E. Agre and S.J. Rosenschein (1996), *Computational Theories of Interaction and Agency* The MIT Press.

[Argyris 78] C. Argyris and D.A. Schon (1978), *Organizational Learning*, Addison-Wesley.

[Cohen 95] M.D. Cohen and L.S. Sproull (1995), *Organizational Learning*, SAGE Publications.

[Duncan 79] R. Duncan and A. Weiss (1979), "Organizational Learning:Implications for organizational design", in *Research in organizational behavior*, B.M. Staw (Ed.), Vol. 1, JAI Press.

[Epstein 96] J.M. Epstein and R. Axtell (1996), *Growing Artificial Societies*, MIT Press.

[Espejo 96] R. Espejo, W. Schuhmann, M. Schwaninger and U. Bilello (1996), *Organizational Transformation and Learning*, John Wiley & Sons.

[Gasser 88] L. Gasser and A. Bond (1988) *Readings in Distributed Artificial Intelligence*, Morgan Kaufman Publishers.

[Goldberg 89] D. E. Goldberg (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.

[Grefenstette 88] J. J. Grefenstette (1988), "Credit Assignment in Rule Discovery Systems Base on Genetic Algorithms", *Machine Learning*, Vol. 3. pp. 225–245.

[Holland 78] J.H. Holland and J. Reitman (1978), "Cognitive Systems Based on Adaptive Algorithms", in *Pattern Directed Inference System*, D.A. Waterman and F. Hayes-Roth (Eds.), Academic Press.

[Ishida 96] T. Ishida, Y. Katagiri and K. Kuwabara (1996) *Distributed Artificial Intelligence* (in Japanese), Corona.

[Kim 93] D. Kim (1993) "The Link between individual and organizational learning", *Sloan Management Review*, Fall, pp. 37–50.

[Kirn 97] S. Kirn and G. O'hare (1997), *Cooperative Knowledge Processing*, Springer-Verlag.

[March 91] J.G. March (1991), "Exploration and Exploitation in Organizational Learning ", *Organizational Science*, Vol. 2, No. 1, pp. 71-87.

[Nonaka 95] I. Nonaka and H. Takeuchi (1995), *The Knowledge Creating Company*, Oxford University Press.

[Prietula 98] M.J. Prietula, K.M. Carley and L. Gasser (1998), *Simulating Organizations* AAAI Press/The MIT Press.

[Russell 95] S. J. Russell and P. Norving (1995): *Artificial Intelligence: A Modern Approach*, Prentice-Hall International.

[Sutton 98] R.S. Sutton, A.G. Bart (1998), *Reinforcement Learning – An Introduction –*, The MIT Press.

[Takadama 98a] K. Takadama, S. Nakasuka and T. Terano (1998), "Printed Circuit Board Design via Organizational–Learning Agents", *Applied Intelligence: Special Issue on Intelligent Adaptive Agents*, Kluwer Academic Publishers, Vol. 9, No. 1, pp. 25–37.

[Takadama 98b] K. Takadama, K. Hajiri, T. Nomura, M. Okada, S. Nakasuka and K.Shimohara (1998), "Learning Model for Adaptive Behaviors as an Organized Group of Swarm Robots", *International Journal of Artificial Life and Robotics*, Springer-Verlag, to appear.

[Watkins 92] C.J.C.H. Watkins and P. Dayan (1992): "Technical Note: Q-Learning", *Machine Learning*, Vol. 8, pp. 55–68.

[Weiss 96] G. Weiss and S. Sen (1996), *Adaption and Learning in Multi-Agent Systems*, Lecture Notes in Artificial Intelligence, Volume 1042, Springer-Verlag.

[Weiss 97] G. Weiss (1997), *Distributed Artificial Intelligence Meets Machine Learning – Learning in Multi-Agent Environments –*, Lecture Notes in Artificial Intelligence, Volume 1221, Springer-Verlag.