Interactive Knowledge Externalization and Combination for SECI Model

Yoshinori Hijikata **Toshihiro Takenaka** **Yukitaka Kusumura** **Shogo Nishida**
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka
Osaka 560-8531, Japan
{hijikata, takenaka, kusumura, nishida}@nishilab.sys.es.osaka-u.ac.jp

ABSTRACT
One of the subjects in knowledge management is knowledge externalization and combination in which people express their tacit knowledge and formulate it. Communication among people is important for this process. This paper proposes a model that helps two users to have a discussion for knowledge externalization and combination. In this model, the computer verifies the consistency between the knowledge generated by inductive learning from cases and the two user's knowledge sets expressed by themselves. From the result of this verification, it provides the point at issue for the discussion to improve the knowledge. We conducted a user experiment in the domain of gure-fishing where real experts had participated in. We confirmed the effectiveness in activating people's communication and also in improving the quality of their knowledge.

Categories and Subject Descriptors
H.5.3 [**INFORMATION INTERFACES AND PRESENTATION**]: Group and Organization Interfaces

General Terms
Experimentation, Human Factors

Keywords
knowledge management, SECI model, tacit knowledge, explicit knowledge, discussion-support, groupware

1. INTRODUCTION
Knowledge management is getting popular in business fields. In Knowledge management, people consider how knowledge in a company is produced, analyzed, and used for improving the creativeness and the efficiency of their business [1]. Nonaka's SECI model [2] is one of the popular models fro knowledge management. SECI model is based on the premise that (1) there are two types in human knowledge: explicit knowledge which is represented in an explicit language or figure, and tacit knowledge which people have in their minds and are not represented in an explicit way, (2) these two types of knowledge interact each other when people do some intelligent activities, (3) knowledge in an organization or a group is created when people with different knowledge interact each other.

SECI model models people's knowledge activities by defining the following stages: expressing their tacit knowledge (externalization), combining those explicit knowledge (combination), absorbing those systematized knowledge by each person (internalization) and experiencing other people's tacit knowledge (socialization). When we want to support the externalization stage and combination stage by activating the people's interaction, we notice that methodologies of knowledge acquisition for expert systems [5] or discussion-support systems (or meeting-support systems) in CSCW or groupware [6] may be available.

Knowledge acquisition tries to obtain knowledge from people which is to be inputted in an expert system. Generally acquired knowledge is stored in if-then rule or in frame. As popular methods, there are an interview which directly asks people about the knowledge by following some strategies and a method which the computer conducts an inductive learning [7] from cases and a user (expert) checks the correctness of the learned rules by hand (The details are explained in Section 2). However, these methods do not support the people's real-time discussion for knowledge acquisition.

Discussion-support systems support people's collaborative activities with discussions. There are the following two popular methods (The details are explained in Section 2). One is a method which models the flow of discussion and lets the user put a tag to his/her comment.

K-CAP'07, October 28–31, 2007, Whistler, British Columbia, Canada

The other is a method which provides a tool for formulating ideas like KJ method [8] and also provides an interface to do a discussion on the tool. However, these methods do not consider the content of the discussion.

We can say that existing methodologies in computer science cannot support SECI model enough. This research targets the situation where two experts (after here, just "users") do a discussion in real-time for knowledge externalization and combination (after here, "knowledge EC"). We propose a model in which the computer participates in the users' discussions for notifying potential problems to the users and in which the user easily enriches both of the personal knowledge only for himself/herself and the shared knowledge for the group.

In this model, the system logically verifies the two users' sets of knowledge and provides the point of issue for the discussion. In detail, the two users write down their knowledge in an explicit way respectively in advance. The system detects a difference or an incoherence between these two types of knowledge as a flaw and provides a question for solving the flaw to each user. The two users discuss their wrong decision or the insufficient condition according to the question, and add, delete or change their expressed knowledge.

Recently, people share cases on the Internet in some domains. Cases are data which record what have happened in the target domain. The system also conducts an inductive learning from cases and creates another knowledge. The system conducts the above flaw detection not only between the two users' knowledge sets but also among the two users' knowledge sets and the knowledge set learned from cases. We hope that knowledge learned from cases performs the role of third person who provides the different view point from the two users. In this way, we aim to activate the users' discussion more by the automatically created knowledge.

In the proposed model, the user improves the knowledge set expressed by himself/herself as a personal knowledge set only for himself/herself. The two users also improve the knowledge set learned from cases as a common resource for the group. Becasue this knowledge is automatically created from cases, it has no bias. This is a good seed for creating shared knowledge in a group. We also expect that an explicit definition of knowledge owners leads to getting honest knowledge from users because they do not have to care security or privacy.

The organization of the remainder of this paper is as follows. Section 2 describes the related work. Section 3 describes the overview of our model. Section 4 explains the definition of flaw and the feedback messages to the user. In Section 5, we did a user experiment in which real experts joined for evaluating our model. Section 6 offers some conclusions.

2. RELATED WORKS

This section introduces existing researches of knowledge acquisition and discussion-support systems.

2.1 Knowledge acquisition

Knowledge acquisition methods by interviews usually exploit some questioning strategies [5]. Although the earlier methods conduct an interview only once, the later methods try to obtain knowledge interactively with the user. In Mole [9], the system shows the problems which happened when using the acquired knowledge in real situations. More [10] and ETS [11] expects the feedbacks from the user by showing the acquired knowledge in styles different from the style the user has used for expressing his/her knowledge.

In knowledge acquisition methods by inductive learning, learning errors are inevitable. Therefore many methods do a deductive explanation on the learned rules [12, 13, 14, 15]. In the deductive explanation, the system matches the learned rules to the rules which the user has inputted beforehand, and shows the inconsistency to the user. However, these methods suppose that only one user does a knowledge acquisition at a time. They do not suppose the situation in which several users discuss the problems for acquiring knowledge in real-time.

Some works support a real-time discussion by several users for knowledge acquisition. AQUINAS [16] asks several users to fill a matrix whose axes are attributes and classes, and prompts a users' discussion by showing the difference of the users' matrices. GRAPE [17] proposes an interface where users in different places can discuss in real-time and collaboratively build a classification tree. In these systems, the users can create a knowledge which roughly classifies cases to small number of categories. However, we cannot create a rule which consider the detailed conditions of attributes.

2.2 Discussion-support system

As popular models of the flow of discussion, there are conversation theory [18] and IBIS model [19]. Many discussion-support systems allow the users to discuss the issues guided by these models [18, 19, 20, 21]. These systems can track the state of the discussion because they ask the users to put a tag representing the type of his/her message. There also exists EMSs(Electronic Meeting Systems) [22] or chat-support systems [23] with laxer process management of conversation like brain storming and NGT(Nominal Group Technique). However these systems do not analyze the content of the conversations. Even if there is incoherence among the conversations, they do not indicate it to the users.

As popular discussion-support systems using a tool for formulating ideas, there are Cognoter [24], GrIPS [25], Tivoli [26] and PReSS [27]. In these systems, the users can put keywords which occur in their minds on the

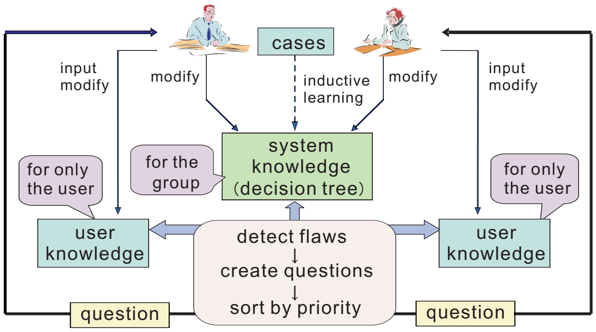

Figure 1: Overview of our proposed model

attribute	: attribute value
water temperature (wt)	: very high / high / medium / low / very low
weather	: fine / clowdy / rain
wind	: strong / medium / weak
weather (previous day)	: fine / clowdy / rain
wind (previous day)	: strong / medium / weak
diff. of water temperature from the previous day	: up / medium / down
max. velocity of tide	: fast / medium / slow
tide type	: spring / middle / neap
red tide	: yes / no
wave-dissipating block	: yes / no
bait	: krill / worm / sea slater / crumb

Figure 2: Examples of attributes and their possible categorical values in gure-fishing domain

screen, change those locations, and link them each other. These systems focus on the process of idea creation (convergence and divergence of ideas). As a function of checking the content of ideas, only GrIPS recommends keywords, which associatively enriches the idea. However it does not indicate the difference or incoherence between ideas.

3. OVERVIEW OF PROPOSED MODEL

3.1 Overview

The overview of our interactive knowledge EC model is shown in Figure 1. Firstly two users express their knowledge in an explicit style (The format is explained in the next subsection) respectively without a discussion. After here, we call this knowledge set "user knowledge." Cases are also inputted to the system. The system conducts an inductive learning from the cases and creates another knowledge. After here, we call this knowledge set "system knowledge". The system detects flaws among the three kinds of knowledge set and creates questions for solving the flaws. Priority is put to each flaw according to its flaw type. Detected flaws and their questions are sorted according to their priority and are shown to the users.

The users discuss the problems guided by the question from the system. Each user improves his/her user knowledge as his/her personal knowledge. The improved user knowledge will be used only by the user. Both the users also improve the system knowledge as knowledge stored for the group. The improved system knowledge will be used by many people in the group. This means that to the neutral knowledge which is obtained from cases, the knowledge which both the users can confidently recommend to others will be left or added. Like this, our model tries to realize knowledge EC from the viewpoint of respecting each user's knowledge and keeping the more sophisticated knowledge for the group. After revising one or a few units of knowledge, the users ask the system to detect flaws again from the revised knowledge sets. By repeating this, the users can interactively conduct knowledge EC with the system.

3.2 Format of knowledge

We use if-then rule as a knowledge representation in our model. We target classification knowledge as a knowledge type. Classification knowledge is a type of knowledge which consists of a decision and information for making a decision. Concretely this knowledge is expressed in class, attribute and attribute value. We use categorical value as attribute value for making the users express their knowledge easier. Figure 2 shows an example of class, attribute, and attribute value in the domain of gure-fishing which is used in the user experiment in Section 5. In this case, there are 11 attributes like weather and bait. There are several categorical values to each attribute. Class is either "good catch" or "bad catch". Altough we have boolean value in class in this example, the model allows us to use N-class value. The followings are the formats of a case, user knowledge, and system knowledge.

A case is represented in attribute values to all the attributes and which class the case belongs to. The rule format for a user knowledge is as follows:

if (attribute, value) $\cdots$ (attribute, value) then (class)

In "if part" ("conditional part"), each set of attribute and attribute value is called "conditional clause". The relationship between adjacent conditional clauses is AND. The model does not allow us to use OR between conditional clauses. If the user should write OR in the rule, he/she has to split the rule into two rules.

System knowledge is created by decision tree. Decision tree is a tree structure in which an internal node shows an attribute to be tested, an edge shows attribute value, and a leaf node shows a class. This tree structure is created from cases by inductive learning [7]. One path from the root node to a leaf node can be seen as one if-then rule. The rule format for a system knowledge is as same as that for a user knowledge. Examples of user knowledge and system knowledge in gure-fishing domain are shown in Figure 3-(a-c).

User knowledge (User A)

Rule1 : if (wt, very high) then (bad catch)
Rule2 : if (wt, high) (tide type, spring) then (bad catch)
Rule3 :
Rule4a : if (wt, high) (tide type, neap) (bait, slater) then (good catch)
Rule4b : if (wt, high) (tide type, neap) (bait, krill) then (bad catch)
Rule4c : if (wt, high) (tide type, neap) (bait, worm) then (bad catch)
Rule4d : if (wt, high) (tide type, neap) (bait, crumb) then (bad catch)
Rule5 : if (wt, low) (diff of wt, medium) then (good catch)
⋮

(a) User knowledge (User A)

User knowledge (User B)

Rule1 : if (wt, very high) then (bad catch)
Rule2 : if (wt, high) (tide type, spring) then (good catch)
Rule3 : if (wt, high) (tide type, middle) then (good catch)
Rule4 : if (wt, high) (tide type, neap) then (bad catch)
Rule5 : if (wt, low) then (good catch)
⋮

(b) User knowledge (User B)

System knowledge

Rule1 : if (wt, very high) then (bad catch)
Rule2 : if (wt, high) (tide type, spring) then (good catch)
Rule3 : if (wt, high) (tide type, middle) then (good catch)
Rule4 : if (wt, high) (tide type, neap) then (bad catch)
Rule5 : if (wt, low) then (bad catch)
⋮

(c) System knowledge

Figure 3: Examples of rules in gure-fishing domain

4. FLAW DETECTION

In our model, the system picks up a rule respectively from one user's user knowledge, another user's user knowledge, and the system knowledge. Then it checks whether or not there is a flaw among those three rules. The system checks all the combinations of rules from the three kinds of knowledge set. A flaw is put to the combination of rules.

4.1 Definition of flaw

We define three kinds of flaw as follows:

(Type1) Contradiction:
Among two or three rules, the conditional parts are same, but the classes are different each other.

(Type2) Lack, excess or replacement of condition:
Pick up two rules from the three rules. Seeing from one rule, there is a conditional clause with an attribute, which the rule does not use, in the other rule (lack of condition). Or seeing from one rule, there is no conditional clause with an attribute, which the rule uses, in the other rule (excess of condition). Or each rule has a conditional clause whose attribute is not used in the other rule (replacement of condition). The flaw of Type2 occurs only when both the rules have at least one clause whose attribute and attribute value are same.

Table 1: Priority of flaws

Priority	Combination of the three rules	Flaw type
1	The conditional parts of the three rules are same, but there are some rules whose classes are different from others.	Type1
2	In two rules out of the three rules, the conditional parts are same, but the classes are different each other. And there is a lack, excess or replacement of condition in the remaining rule.	Type1 and Type2
3	In two rules out of the three rules, the conditional parts are same, but the classes are different each other. And there is not a lack, excess or replacement of condition in the remaining rule.	Type1
4	Two rules out of the three rules are same. And there is a lack, excess or replacement of condition in the remaining rule.	Type2
5	Two rules out of the three rules are same. And there is not a lack or excess or replacement of condition in the remaining rule.	Type3
6	There are no rules with the same conditional part. But there is a pair of rules which have a lack, excess or replacement of condition each other.	Type2
7	All the three rules have no same clause in the conditional part each other.	

(Type3) Lack of rule:
The same rule (Both the conditional part and class are same) is in two types of knowledge set, but it is not in the rest of the knowledge set.

4.2 Priority of flaw

For defining the priority of flaw, we consider the following three fundamental policies:

(1) Give priority to the flaw which has contradiction.

(2) Give priority to the flaw which has many rules whose conditional parts are same.

(3) Give priority to the flaw which has rules with a lack, excess or replacement of condition.

We also think that there is higher importance in the order of (1)-(3). The reason is as follows. Firstly the contradiction makes a user, who uses this knowledge, confused because he/she does not know which rule's decision to adopt. Therefore Policy (1) comes first. Policy (2) considers how many people think the condition important. Policy (3) considers whether or not the system can provide more concrete suggestion. We thought that it is easier for the two users to think from important rules. When the users meet more important rules which affect broader cases after correcting some trivial rules, they may have to revise again those corrected rules. Therefore Policy (2) comes before Policy (3). From the above discussion we define the priority of flaws as in Table 1. We also show the rules which only one user has as Priority 7. This is because other users may consult this rule for acquiring new knowledge.

4.3 Content of question

In our model, the system selects a user to show a question based on the correctness of the user's knowledge.

Table 2: Contents of questions to the flaws

Flaw type	Content of question
When the conditional parts of the three rules are same	
Type1	"Only your decision is different from others."
Type1	"All the members' decisions are different each other."
When two (User A and User B) rules are same	
Type3	"Only you do not have this rule." (to User C)
Type2	(If lack of condition) "Your rule lacks of attribute X. Your rule may be generalized excessively." (to User C)
Type2	(If excess of condition) "Only your rule uses attribute X. Your rule may be specialized excessively." (to User C)
Type2	(If replacement of condition) "You considers attribute X as a critical one. But attribute Y may be a critical one." (to User C)
When two (User A and User B) rules' conditional parts are same but their classes are different	
Type1	"Your decision is different from User B (A)." (to User A(B))
Type2	(If lack of condition) "Attribute X may not influence the decision." (to User A(B))
Type2	(If excess of condition) "There is a possibility that you cannot make a decision unless you consider attribute X." (to User A(B))
Type2	(If replacement of condition) "There is a possibility that critical attribute is not attribute X but attribute Y." (to User A(B))
When no rule's conditional part is same	
Type2	(If lack of condition) "Your rule does not consider attribute X which is used in User A's rule. Please check your rules each other."
Type2	(If excess of condition) "You use attribute X in your rule. User A does not consider the attribute. Please check your rules each other."
Type2	(If replacement of condition) "You consider attribute X as a critical one. But, User A considers attribute Y as a critical one. Please check your rules each other."

This selection is conducted by the majority vote. When the system selects a rule respectively from the three kinds of knowledge set, if the two rules are same and the other rule is different from them, the system sees the different rule as an incorrect one. The system shows a question to the owner of the rule (User A) for solving the flaw. It also notifies the other user (User B) that the question is shown to User A. If the owner is the system, it shows a question to both the users.

A question is created according to the type of the flaw and the combination of the rules which causes the flaw. We designed the contents of questions as in Table 2. This table shows the questions when one user's rule is not correct. Firstly, we explain the case when the conditional parts of the three rules are same. In this case, when two rules' classes are same and the other rule's class is different from them, the system tells the user whose decision is different that the user's decision is different from others. When all the rules' classes are different, the system tells all the users that all the users' decisions are different.

Secondly, we explain the case when only two rules' conditional parts are same. The question differs between the case that those two rules' classes are same and the case that they are different. When they are same, and if the flaw is Type3, the system tells the user who does not have the rule that only the user does not have the rule. If the flaw is Type2, the system provides a question to the user whose rule is different from others' (we call this user "target user" in this paragraph). The content differs by lack, excess, or replacement of condition. If the target user's rule has a lack of condition to other users' rules, he/she may generalize the rule excessively. This means that he/she does not know the missing attribute influences the decision, or even if he/she knows that, he/she thinks the attribute is not so critical. If the target user's rule has an excess of condition to other users' rules, he/she may specialize the rule excessively. This means that he/she considers very rare cases, or thinks indecisive attributes as critical ones.

When the two rules' classes are different, this becomes a flaw (Type1). Firstly the system tells both the users that their decisions are different each other. If the flaw is also Type2, the system provides a question based on the idea that the rule whose conditional part is different from the other two rules (we call the owner of this rule "target user" in this paragraph) serves as a useful reference for revising those two rules. The content differs by excess, lack, or replacement of condition. If the target user's rule has a lack of condition to other users' rules, firstly the system tells the target user of that fact. Then it tells the other users that their extra attribute may not influence the decision. This is because the decision becomes difficult due to the consideration of the extra attribute and in the result their decisions are divided. If the target user's rule has an excess of condition to other users' rules, the system tells the other users that their missing attribute may influence the decision. This is because their decisions may be divided because they miss the critical attribute.

When the three rules' conditional parts are different each other, flaws (Type2) can happen. In this case, because the system cannot predict whose rule is wrong, it just tells all the users that their rules include a lack, excess, or replacement of condition to others'.

4.4 Examples of flaw

Examples of flaw are shown in Figure 3-(a)-(c). The combination of the three users' Rule1s is not a flaw because all the rules are same. The combination of Rule2s is a flaw (Type1) because the conditional parts of the three rules are same but only User A's rule is different in class. The combination of Rule3s is a flaw (Type3) because only user A does not have the rule "if (wt, high) (tide type, middle) then (good catch)". The combination of Rule4s is a flaw (Type2) (also the case that two rules are same). In this case, User B and the system reach a consensus in that "if (wt, high)(tide, neap) then (bad catch)." However, User A tries to judge this case by also considering the bait. In this case, if the rule of User B and the system is correct, incorporating the bait makes the rule overspecialized. The combination of Rule5s is a flaw (Type2) (also the case that two rules' conditional parts are same, but their classes are different). To the condition (wt, low), UserB's decision is "good catch" and the system's decision is "bad catch." User A tries to consider this case including the attribute "difference of water temperature from the previous day." There is a possibility that we cannot judge only from the water temperature of the day and we can judge when considering also the difference of water temperature from the previous day.

5. EVALUATION

5.1 Objective of evaluation and target domain

We think that the followings are major issues for the evaluation of our model:

(1) Validity of flaw definition and question

(2) Validity of flaw priority

(3) Performance of each user's knowledge EC

We conducted this evaluation by asking real experts to use our system for real problems. Due to the space limitation of the paper, we show the result of Issue (1) and Issue (3). To Issue (2), we confirmed its validity from the order of the users' rule modification and the answers to the questionnaire about the usefulness of the flaw order. For Issue (1), we counted the number of flaws which were left after the experiment, and asked the users the reason they had not modified the rules for each left flaw. For Issue (3), we conducted two kinds of experiment: an experiment in which the users used the system with the flaw detection and an experiment in which the users used the system without the flaw detection. We checked the number of modifications of the rules and the improvement of the rules' quality. Finally, we analyzed the content of the discussion during the knowledge EC for acquiring the further findings. In our research, we conducted the experiment in the domain of fishing because it is one of the typical domains in which experts need to conduct knowledge EC. Among many types of fishing, we target gure-fishing ("gure" is a a name of fish in Japanese. It is called "girella" in English.) around Akashi channel in Japan.

5.2 Experimental method

The knowledge acquired in this experiment is whether we can get good catch or bad catch under a specific condition of weather and bait. Attributes and attribute values were set as in Figure 2. Class is either "good catch" or "bad catch." The range of the attribute values were set by considering the real weather data recorded for one year. We obtained the weather data from Hyogo Fisheries Technology Institute and Kobe City Fishing Park. Six anglers whose skills were almost same provided 66 cases as fishing results. After the inductive learning, 19 rules were created.

Table 3: The number of left flaws out of the detected flaws and the solution ratio

Flaw type	Pair 1	Pair 2	Pair 3	All pairs	solution ratio(%)
1	0/0	0/2	0/1	0/3	100
2	0/55	1/63	0/33	1/151	99.3
3	0/38	4/29	0/26	4/93	95.7
IR	7/50	3/57	0/33	10/140	92.9

IR:Independent rule

12 experts including an editor of a commercial fishing magazine and a champion of the all-Japan gure-fishing tournament participated in this experiment. Two users who are in the same room discuss the issues in real time for knowledge EC. Flaws and questions are displayed in their own PCs. Two PCs share rules through the network. We made six pairs from 12 users. Three pairs used the system with the flaw detection. The other three pairs used the system without the flaw detection.

5.3 Validity of flaw definition and question

Table 3 shows the number of the detected flaws and that of the left flaws without rule modifications in each type of flaw including the rules only one user independently has. We also show the ratio of solving the flaws (solution ratio) in Table 3. From this table, we can see that the ratio of the left flaw is less than 5%. We asked the users to select a reason that they did not solve the flaw from the following options. It is asked in each unsolved flaw. We also show the number of answers bellows.

(1) I did not modify the rule because the difference of condition or decision is not so important. (0 times)

(2) It was a trouble to modify the rule. (0 times)

(3) I forgot to modify the rule. (0 times)

(4) We did not modify the rule because our opinions are different. (3 times)

(5) It is difficult to judge by myself. (19 times)

(6) I think the system's question is wrong and this flaw should be left. (0 times)

(7) Other. (8 times)

All the reasons the users selected "Other" are "Our rules are same, but we were not confident to keep the rule for the group." From the results of the questionnaire, we can see that all the reasons are caused by the content of the rules and not caused by the definition of flaw or the contents of the system's questions. From the results of the above two kinds of evaluation, we think that the definition of flaw and the contents of questions for solving them is valid. We also notice from these results that the rules which only one user has are left and the most of the above reasons are that the users were not confident to keep the rule for the group. We can see that the users used the function of separately storing the knowledge which are public for the group and the knowledge which are personal only for the user.

Table 4: The number of rule operations.

(a) With the flaw detection

	User 1	User 2	User 3	User 4	User 5	User 6
change	4	3	13	2	2	7
deletion	3	7	11	11	4	0
addition	14	18	20	12	25	24

(b) Without the flaw detection

	User 7	User 8	User 9	User 10	User 11	User 12
change	0	2	0	4	1	2
deletion	1	0	20	5	1	5
addition	6	0	7	1	0	2

Table 5: Precision of rules created by each user

	User 1	User 3	User 5	User 7	User 9	User 11
before	0.54	0.43	0.43	0.46	—	0.54
after	0.65	0.71	0.54	0.50	0.54	0.61

5.4 Effectiveness of knowledge EC

The number of times of rule change, deletion and addition are shown in Table 4. We can see that the number of rule operations is larger in the case with the flaw detection than in the case without the flaw detection. This means that the system provided more clues for acquiring or improving knowledge to the users. However, we cannot judge whether the users could improve their knowledge by using the system with the flaw detection. If we want to know this, we have to use both the knowledge (the rules before using the system and the rules after using the system) in real situations.

For this evaluation, we asked one expert of gure fishing to go fishing for two years and give us the fishing results. We used these fishing results as test data for evaluating the quality of the rules. There are 14 "good catch" days and 17 "bad catch" days. We deleted randomly three days from 17 "bad catch" days. Thus, we used 14 "good catch" days and 14 "bad catch" days as test data. We used the rules of User 1, 3, 5, 7, 9 and 11. We applied the rules before using the system and the rules after using the system to the above test data. We compared the matched rule's class with the real case's class in each day.

The precision is shown in Table 5. We can see that the precision has become higher for every user (except User 9). User 9 firstly could not understand the difference between rules and cases and wrote many incomplete rules. We cannot calculate the precision before using the system because there are cases which do not match any rule. We can also see that the improvement of precision is higher for User 1, 3 and 5 than for User 7 and 11. The precision after using the system of User 1, 3 and 5 tends to be higher than that of User 7, 9 and 11. We can say that the users who used the system with the flaw detection could improve the quality of the knowledge better than the users who used the system without the flaw detection. The quality of the final knowledge of the users with the flaw detection also became better than that of the users without the flaw detection.

Table 6: The number of rule operations by the discussion type

Discussion pattern	Pair 1 User 1	Pair 1 User 2	Pair 2 User 3	Pair 2 User 4	Pair 3 User 5	Pair 3 User 6
1	15	19	19	13	23	25
2	2	5	11	2	3	2
3	0	1	6	0	0	0
4	0	0	0	0	8	0
5	1	3	7	7	4	3

From the results of the two experiments in this subsection, we can see that the proposed method activated the users' discussions, and in the result they tried to improve their knowledge more times, and finally the quality of the knowledge became better.

5.5 Analysis of discussions

In this subsection, we analyze the users' discussions for acquiring further findings. We found that discussions when users modify their rules are categorized as follows:

Pattern 1: The users notice the error of the rules by discussing each other.

Pattern 2: The user does not have the experience, but he/she understands the other user's explanation.

Pattern 3: The user does not understand the other user's explanation, but he/she has confidence with the other user and modifies his/her rule.

Pattern 4: The users disagree with each other, but one user gives in and modifies his/her rule.

Pattern 5: The user reconsiders the rule and modifies it by himself/herself.

We analyzed the video of the discussions, and categorized all the discussions when modifying the rule according to the above five patterns. Table 6 shows the result. The human relationships of the pairs are as follows. In Pair 1, they met for the first time. User 1 has more advanced skills than User 2. In Pair 2, their relationship is the master and pupil. User 3 is a pupil and User 4 is a master. In Pair 3, they are friends and their skills are almost same.

From Pattern 2 in Table 6, we can see the difference of the users' skills. In Pair 1 and Pair 2, the number of times of modifying rules differs between the users. From Pattern 3 in Table 6, we can see the influence of the relationship of the master and pupil. User 3 modified the rules many times even if he did not understand the other's explanation. In Pair 3, User 6 tried to find a reason to each rule learned from cases and explained it to the other. But his explanation sometimes seemed like a stretch. We found many scenes where User 5 got confused. User 5 is more reserved on discussions than User 6. This is why Pattern 4 occurred for User 5. From this result, we can see that the performance of the knowledge EC is influenced by the members' relationships like confidence and the members' personalities. We have to carefully select members for the knowledge EC or show a guideline for the discussion to the members.

6. SUMMARY AND FUTURE WORKS

In this research, we proposed a model which supports two users' discussion for externalizing tacit knowledge and combining the expressed knowledge for SECI model. In our model, the computer creates knowledge by inductive learning of cases and detects flaws among this knowledge and the two kinds of knowledge expressed by the two users respectively. It provides a question for solving the flaw to the appropriate user. The two users do a discussion in real-time from this question. We applied our model to the knowledge management in the gure-fishing domain. We asked some gure-fishing experts to do a discussion by using this system. From the result, we can see that the users tried to improve the knowledge more by using our system. We also see that the the knowledge improved by the system works better for the real situation. The current problem of our model is that the knowledge representation in the model is only if-then rule. The future work of this model is to extending other types of representation by introducing basic natural language processing techniques.

7. REFERENCES

[1] Drucker, P.F., et al.: Harvard Business Review on Knowledge Management, *Harvard Business School Press*, 1998.

[2] Nonaka, I.: A Dynamic Theory of Organizational Knowledge Creation, *Organization Science*, Vol.5, No.1, pp. 14–37, 1994.

[3] Davenport, T. and Prusak, L.: Working Knowledge, *Harvard Business School Press*, 1998.

[4] Wenger, E., McDermott, R. and Snyder, W.M.: Cultivating Communities of Practice, *Harvard Business School Press*, 2002.

[5] Jackson, P.: Introduction to Expert Systems, *Addison-Wesley Publishing Company*, 1986.

[6] Grudin, J.: CSCW, *Comm. of the ACM*, Vol.34, No.12, pp. 30-34, 1991.

[7] Quinlan, J.: C4.5 Programs for Machine Learning, *Morgan Kaufmann Publishers*, 1993.

[8] Kawakita, J.: The KJ Method – a Scientific Approach to Problem Solving. Technical report, Kawakita Research Institute, 1975.

[9] Eshelman, L. and McDermotto, J.: MOLE: A Knowledge Acquisition Tool That Uses Its Head, *Proc. of AAAI'86*, pp. 950–955, 1986.

[10] Kahn, G.: Strategies for Knowledge Acquisition, *IEEE Trans. on PAMI*, No. 5, pp. 511–522, 1985.

[11] Boose, J.: A Knowledge Acquisition Program for Expert Systems based on Personal Construct Psychology, *Journal of Man-Machine Studies*, Vol. 23, pp. 495–525, 1986.

[12] Dabija, V.D., Tsujino, K. and Nishida, S.: Learning to Learn Decision Trees, *Proc. of AAAI'92*, pp.88-95, 1992.

[13] Lebowitz, M.: Integrated Learning: Controlling Explanation, *Cognitive Science*, Vol. 10, pp. 219–240, 1986.

[14] Towell, G.G., Shavlik, J.W. and Noordewier, M.: Refinement of Approximate Domain Theories by Knowledge-based Neural Networks, *Proc. of AAAI'90*, pp. 861–866, 1990.

[15] Pazzani, M.J. and Kibler, D.: The Utility of Knowledge in Inductive Learning, *Machine Learning*, Vol.9, pp.57-94, 1992.

[16] Boose, J.H. and Bradshaw, J.: Expertise Transfer and Complex Prblems, *Intl. Journal of Man-Machine Studies*, Vol. 26, No. 1, pp. 3–28, 1987.

[17] Ueda, H.: Knowledge Acquisition Support Groupware GRAPE, IPSJ SIG Technical Reports of HI, Vol.1991 No.18, pp.171-178, 1991.

[18] Winograd, T. and Flores, F.: Understanding Computers and Cognition, *Addison-Wesley Publishing Company, Inc*, 1986.

[19] Conklin, J. and Begeman, M.L.: gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *Proc. of CSCW'88*, pp.140-152, 1988.

[20] Shepherd, A., Mayer, N. and Kuchinsky, A.: Strudel - An Extensible Electronic Conversation Toolkit, *Proc. of CSCW'90*, pp. 93–104, 1990.

[21] Medina-Mora, R., et al.: The Action Workflow Approach to Workflow Management Technology, *Proc. of CSCW'92*, pp. 281-288, 1992.

[22] Nunamaker, J.M., et al.:Electronic Meeting Systems to Support Group Work, *Comm. of the ACM*, Vol.34, No.7, pp.40-61, 1991.

[23] Farnham, S., et al.: Structured Online Interactions: Improving the Decision-Making of Small Discussion Groups, in *Proc. of CSCW'00*, pp. 299–308, 2000.

[24] Gregg, F. and Stefik, M.: Cognoter, Theory and Practice of a Colaborative Tool, *Proc. of CSCW'86*, pp.7-15, 1986.

[25] Kohda, Y., et al.: Group Idea Processing System: GrIPS, Journal of JSAI, Vol. 8, No. 5, pp. 65–74, 1993.

[26] Moran, T.P., van Melle, W. and Chiu, P.: Tailorable Domain Objects as Meeting Tools for an Electronic Whiteboard, *Proc. of CSCW'98*, pp.295-304, 1998.

[27] Cox, D. and Greenberg, S.: Supporting Collaborative Interpretation in Distributed Groupware, *Proc. of CSCW'00*, pp.289-298, 2000.