Simulated Collaboration to Understand Japanese Offshore Software Development in China

Takaya Yuizono* and Lihua Xuan*

*School of Knowledge Science, Japan Advanced Institute of Science and Technology, Japan
yuizono@jaist.ac.jp

Abstract – Two primary issues in Japanese offshore software development in China are how to manage specifications and intercommunication. To consider the software development using groupware technology in a laboratory setting, we proposed a simulated collaboration task. In the experiment, the Chinese, Japanese, and Japanese-Chinese groups corresponded with offshore development, domestic development, and cooperative development, which is a proposed future style, respectively. The laboratory experiments' results indicated that a well-collaborated team produced a good model chart; the Chinese participants experienced difficulty in using Japanese-language communication, making some of them self-assertive; and, in comparison to the Japanese participants, the Chinese participants tended to be satisfied with their results of the model chart, considering them neither good nor bad.

Keywords: offshore software development, software specification, laboratory simulation

1 INTRODUCTION

Nowadays, both Japan and China widely experience interactions in their trade and between their people. In trade, software development is representative of work that necessitates knowledge workers, making it an important collaborative undertaking. Japanese IT offshoring came into prominence in the 1980s, motivated by pressures to reduce labor costs and, to date, has been carried out primarily in China. Offshoring in recent years requires the management of the complete process of software development [1].

The largest amount of offshoring from Japan is sent priarily to China; hence, China receives the highest number of outsourcing jobs from Japan [2]. Offshoring from Japan to China is still prominent in the software development industry, although offshoring to India and Vietnam has also increased. As is well known, all offshoring projects were not always successful because of differences in language, culture, corporate climate, and work environment.

Groupware technologies, which support distributed software development, have the potential for IT offshoring because the major issues in offshore development involve software specification and human communication. In order to understand the software development process in different cultural settings, the time taken, expenses incurred, and detailed observations or logs maintained by an IT vendor are necessary [3]. In addition, evaluating a new type of offshoring may not be realistically acceptable to the vendor. Therefore, we design a collaboration task that simulates offshoring development and is availabile in a laboratory setting.

In the following section, we describe the collaboration model for offshore software development. In the third section, we explain the experiment and proposed collaboration task. In the fourth section, we present the results of the experiment and discuss them. In the last section, we summarize this paper and suggest a future direction.

2 SIMULATED COLLABORATION FOR OFFSHORE SOFTWARE DEVELOPMENT

2.1 Collaboration Task

We proposed a work model on the basis of the collaboration between the Japanese and Chinese in offshore software development, to investigate or explore the issues that exist in this collaboration.

The questionnaire on offshore software development was based on that of Nakahara and Fujino [4]. The results showed that, in the case of Japanese companies, about 20 percent indicated a communication problem, about 30 percent indicated problems with confirmation and modification of specifications, and about 20 percent indicated a communication problem in the case of offshore companies.

On the other hand, software development based on a specification is difficult to replicate in a laboratory experiment because the work is limited to only a few hours and the recruitment of programmers for the software development is unrealistic, since many students have yet to learn the essentials of programming. Therefore, we considered model charting as the collaborative work instead of program development. The model that simulates the collaboration in offshore software development is depicted in Figure 1.

In the collaboration task, the client and vendor are located at separate remote sites. The client sends a software specification to the vendor, and the vendor develops a model chart instead of developing a software program according to the specification.

The vendor consists of three persons—the team leader and two workers, who are not programmers—and they work in the same place. In the laboratory, the workers can pose any questions regarding the specification to the leader. When the leader is faced with a question that has not been resolved, he/she can question the client, who is in a different location.

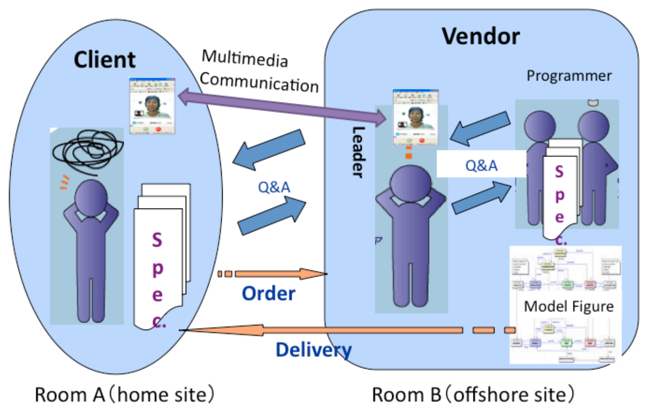

Figure 1: Collaboration model of offshore software development

The vendor prepares a system chart on the basis of the specification that is assumed to be ordered by the client. For this collaboration, workers can use a paper and pencil, but they must depict the chart as a final groupware product that supports a shared screen and chart-making function. The workers use a label and arrow representation for the charting. In the chart, a label object is used to represent an object in the specification and an arrow is used to represent the sending of an event; the arrow is often added to the string using a label object to explain the event. The leader can check the chart on the shared screen using the groupware.

In the task, two versions of specifications were prepared in order to mimic a problem caused by a specification change, which is considered a typical cause of confusion in software development [4]. The modified specification is an advanced version of the specification that was initially ordered by the client.

Three team patterns are considered because the characteristics of offshore software development are expected to be revealed in the comparison of these patterns. In the first pattern, all the people belonging to the vendor's side are Chinese. This simulates ordinary offshore software development and, hence, is labeled the "Chinese Group." In this case, the team leader is a Chinese who is fluent in Japanese, but the workers cannot speak Japanese though they can read Japanese sentences. In the second pattern, all the people are Japanese and are labeled the "Japanese Group," in order to simulate a domestic software development team for the comparison. The third pattern is considered the future style of offshore software development and is reflected by a collaboration of Japanese and Chinese participants and, hence, is labeled the "JC Group." In this pattern, the leader is Japanese and the workers are Chinese.

2.2 Software Specification for the Collaboration Task

The software specification was prepared before the collaboration task. The content of the specification was taken from a standard problem on stock management, for software specification research [5].

The theme for problem solving in the collaboration experiment was entitled "The store management system for a liquor company" and was developed to compare program design methods.

The structure of the specification was referred to as the guideline for the requirement specification [6], and the language description in Japanese follows Ooki et al's explanation [7]. The Japanese description reflects concurrent object processing and has no symbolic description that depends on a specific program design technique. Nowadays, a description with these characteristics suits the specification description for modem object-oriented development. A part of the specification is depicted in Figure 2.

In the specification, there are descriptions of the aim of the systems, glossaries, requirements, and specifications. In the first section, entitled "Introduction," the basic knowledge of the system is explained. In the next section, which is a subsection, the behaviors are explained according to the object. The objects named "Receptionist for stock," "Receptionist for delivery of goods," "Storage," and "Deficiency of stocks" are defined; accordingly, a behavior that sends an event to each object is defined.

1. Introduction
1.1 Goal: Model of Stock Management System
1.2 Scope: Management System of Liquor Shop
1.3 Glossary
Stored object: Commercial product in storage corresponding to each product.
Container: A container loaded with mixed products.
List of stored products: One item from the stored list and memorized correspondence between the stored product and container.
2. Requirements and Specification
2.1 Receptionist for the stock
2.1.1 When a container arrives
2.1.1.1 Making a new container
2.1.1.2 According to each branded product
a. Update the number of stocks to storage
b. Making an item list of the cargo.
2.2 Receptionist for the delivery of goods
2.2.1 When it receives a request for the delivery of goods
2.3 Storage
2.3.1 When it accepts the number of stocks from the receptionist.
2.3.1.1 The number of stocks become "the number of stocks + the number of entry stocks" and update the number to a deficiency of stocks.
2.3.1.2 When the delivery of goods necessarily occurrs because the deficiency of stocks is dissolved.
a. ~
b. ~
c. ~
2.4 Deficiency of stocks
~

Figure 2: Part of the specification with a stored system (underlined specification was added during the collaboration as a modification)

The parts of the sentence that were underlined were added when modifying the specification during the collaborative work to simulate a specification change, which often occurs in software development and causes difficulties in the development.

3 EXPERIMENTS WITH THE COLLABORATION TASK

3.1 Experimental Procedure

There were 18 participants recruited for this research—eight Japanese and ten Chinese—and they were organized into six groups. All the groups assumed the role of vendors and each comprised one leader and two workers. A Japanese teacher assumed the role of the client in all the six experiments. In the case of the "Chinese group," all the members were Chinese; in the "Japanese group," all the members were Japanese; and in the "JC group," the leader was Japanese while the two workers were Chinese.

The procedure for the experiment was as follows. In the beginning, for about 15 minutes, the experimenter explained the collaboration task and how they should use the system. She described the specification and a chart of the system from a textbook on software engineering [8]. In the chart, a label object is used to represent an object in the specification and an arrow is used to represent the sending of an event; the arrow is often added to the string using a label object to explain the event. Then, the participants began the task. The total time for the task was fifty minutes. This time was set from the results of two experimenters, who completed the task in fifty minutes, using computers. At the beginning of the task, the participants were informed that the total time for the task was thirty minutes, and that it should be delivered with the specification of the storage system. The A3-sized paper and the pencil were provided for free usage at the same time. When it was twenty minutes into the task, the experimenter, who was the client, informed the participants of a modification to the specification and delivered the modified specification to them.

After completing the task, the participants answered a questionnaire that was based on a five-point scale and checked the relevant difficulty level with regard to their understanding of the specification. If they checked a low value, they were prompted to write a reason for this. Moreover, they were urged to underline the parts they did not understand and provide reasons or opinions for the same.

3.2 Environment

The experiments were carried out in two rooms—the faculty room and the research staff room—at the Graduate School of the Japan Advanced Institute of Science and Technology. Skype, a well-known software application that allows voice calls over the Internet, was utilized to communicate between the client and the leaders at the vendor site, primarily for questions and answers. The vendor used a groupware called KUSANAGI [9] to create a system chart. The groupware had a brainstorming tool, grouping tool, and arrow tool to support the grouping stage of the distributed and cooperative KJ Method [10]. A picture of the experimental setting at the vendor site is depicted in Figure 3. A screenshot of a modeling chart is shown in the next section.

In order to create the chart using KUSANAGI, the user labeled the objects or event explanations and depicted the event flow using arrows.

Figure 3: The experimental setting at the vendor site: On the left are the two workers and on the right is the team leader.

4 RESULTS AND DISCUSSION

4.1 Results of the Collaboration

The six charts of the storage system referred to the specification that was obtained from the experiment. The evalua-

Table 1: Evaluation of model charts and background of the group

Group name	Chinese Group-A	Chinese Group-B	Japanese Group-A	Japanese Group-B	JC Group-A	JC Group-B
Specification score (correct percentage)	12 (37.5%)	23 (71.9%)	17 (53.1%)	21 (65.6%)	4 (12.5%)	20 (62.5%)
Number of labels (correct percentage)	14 (78.6%)	24 (95.8%)	24 (83.3%)	27 (92.6%)	17 (35.3%)	26 (80.8%)
Number of arrows (correct percentage)	8 (100%)	21 (100%)	21 (66.7%)	20 (95.0%)	11 (36.4%)	27 (77.8%)
Knowledge of IT	1	3 (All)	3 (All)	3 (All)	3 (All)	2
Learning experience of the specification	1.7	4.0	3.7	2.7	1.7	1.7

tion points of the charts and the knowledge background along with which team prepared the charts are summarized in Table 1.

We evaluated the charts corresponding to the specification. The specification described in Section 2 has thirty-two lines. We checked the reflection of the contents on each line of the specification. We marked a circle when the line was adequately reflected in the chart, a triangle when the line was reflected to some degree in the chart, and a cross when the line was not reflected in the chart. Two persons performed this evaluation: one was a client and the other, an experimenter. We assigned 1 point to a circle-marked line, 0.5 points to a triangle-marked line, and 0 points to a cross-marked line. The total number of points from all lines implied the evaluation value of the chart. We named this value the "specification score." The marking between two evaluators indicated a high correlation coefficient: 0.75 from each line and 0.99 from each chart.

In addition, we presented the number of labels and arrows in each chart in Table 1. As described in Section 3, the labels represented an object or an event, and the arrows represented an event flow. We checked the correctness of the labels and the arrows by referring to the specification. The procedure for the check is similar to the procedure for the specification score. The score for the evaluation value of labels and that for the evaluation value of arrows are calculated as correct percentage on the Table 1.

The knowledge background of the group was self-reported, indicated by the number of persons who answered "yes" to knowledge on information technology, and the average score from the five-scale questionnaire on software specification. When the value is five, it implies that the person has learned well, and when the value is one, it implies that the person did not learn at all.

In the results, the Chinese Group-B scored about seventy percent and the Japanese Group-B and the JC Group-B got more then sixty percent. The Japanese Group-A continuously scored about fifty percent. The Chinese Group-A scored about forty percent, and the JC Group-A, only about ten percent. The charts of the groups that scored more than sixty percent had many arrows and many labels compared to the other charts. The charts by the Chinese Group-A, which had a very low score, had fewer labels and fewer arrows, and the correctness of representation was inferior. The model chart prepared by the Chinese Group-B, which had a high score, is depicted in Figure 4, and the model chart by the JC Group-A, which had a low score, is depicted in Figure 5.

Before the experiment, we assumed that a knowledge background affects chart making. However, there are no such characteristics with regard to the knowledge background in the results. The experience with software technology did not always lead to good results. This may imply that the proposed task did not require software technology skills such as programming.

4.2 Results of the Questionnaire

The results of the questionnaires taken after the collaboration task are described below, and the results with the five-scale evaluation are shown in Table 2. A 5-point evaluation implied very good and 1 point implied very bad. From the questions, Q.8 and Q.9 were given to only the workers. We added a star to the left of the table in the case that showed a significant difference in the ANOVA-analysis between the value of the Japanese and Chinese answers.

Table 2 illustrates the interest level of the participants in the work task, levels of cooperation, communication within their groups, and ability to clearly pose questions to the client.

From the perspective of differences between the Japanese and Chinese, the Chinese tended to be more satisfied with their system charts than the Japanese. The Chinese, who belonged to the Chinese Group-A and the JC Group-A, which had low scores, responded that they were satisfied with their results. These results could lead to confusions in software development, so it is necessary to periodically check if the progress of the development is sound.

According to the questions to the workers, the Japanese participants communicated well with the leaders of their groups, but the Chinese participants felt that their communication was neither good nor bad. The Chinese colleagues could speak their mother tongue among themselves, but they were required to communicate in the Japanese language in the case of the JC Group, which is assumed to be the future style of offshore development. In other cases such as questions to a client, the Japanese participants felt that their communication was good, but the Chinese were indifferent about theirs. It is assumed that the differences in language clearly affected the conscious effort to create cooperative communication.

Table 2: Results of the five-scale questionnaire

Items	Value
Q1: Understandability of the collaboration task	3.3
Q2: Interest in the task	3.8
Q3: Pre-image of the system modeling	3.1
Q4: Understandability of specification	2.6
Q5: Satisfaction with the chart	2.5*
Q6: Is the work collaborative?	3.7
Q7: Do you communicate within a group?	3.6
Q8: Can you question your leader?	4.6*
Q9: Does your leader understand your question?	4.4*
Q10: Do you communicate well with the client?	3.3*
Q11: Can you pose a question to the client?	4.1
Q12: Does the client understand your question?	3.7*
Q13 Do you feel the barriers imposed by different cultures?	2.6

*Significant difference between the Japanese and Chinese with the ANOVA-test, $p < 0.05$.

Doubts on the specification were highlighted; after the analysis of the experiment, it was found that twenty out of the thirty-two lines were questioned by the participants owing to their lack of comprehension. The common doubts posed by both the Japanese and Chinese participants comprised five lines that included the repetition of words such as "container" or the "deficiency of stocks," phrases that were ambiguous such as "because the deficiency of stocks is dissolved" or "when a delivery of goods necessarily occurred,"

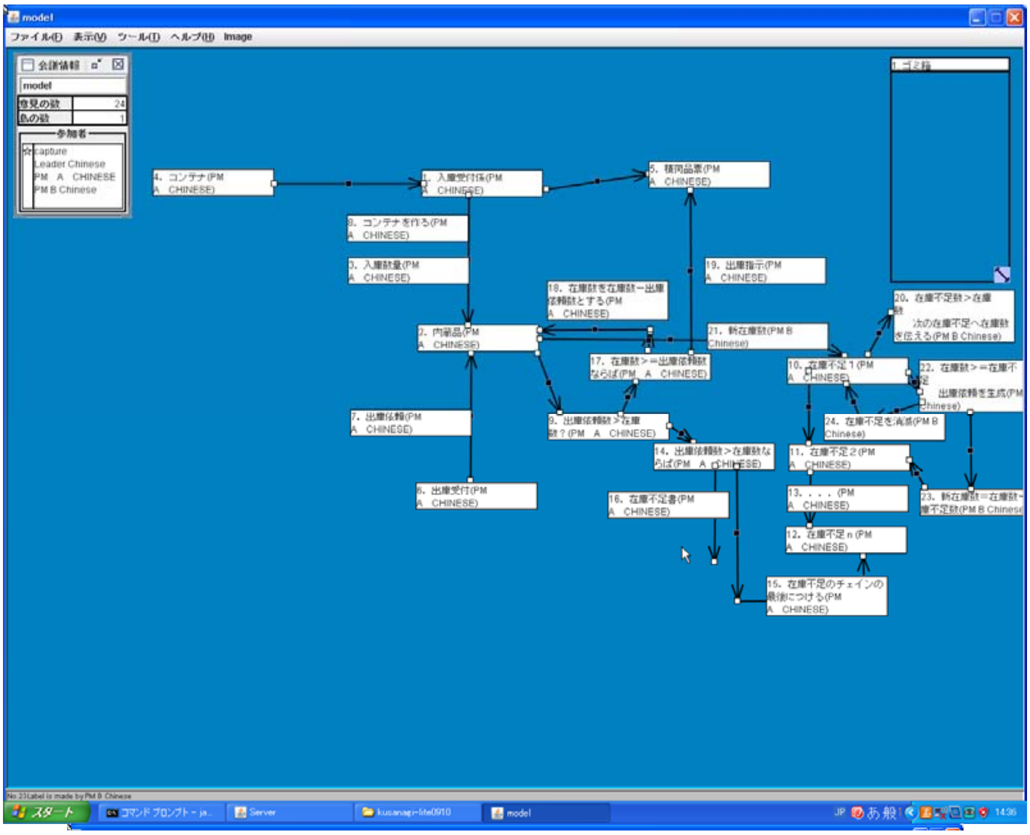

Figure 4: Screenshot of a Good Model Chart created by the Chinese Group-B

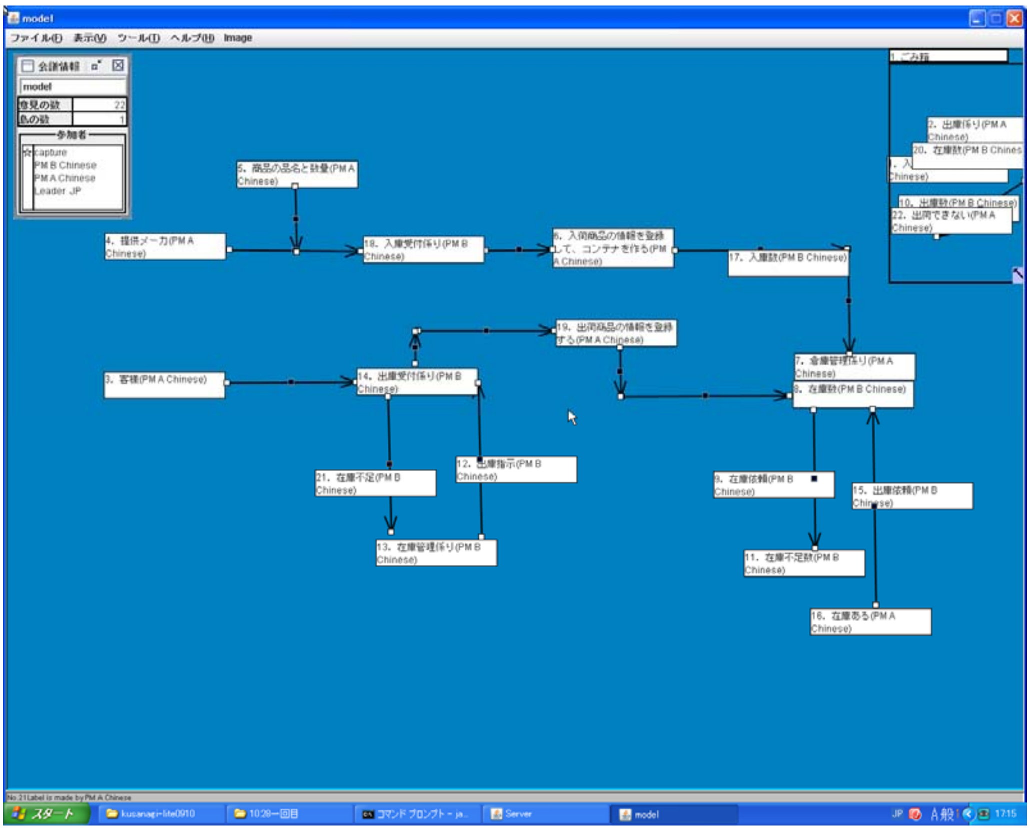

Figure 5: Screenshot of a Poor Model Chart created by the JC Group-A

Table 3: Video observation of the collaboration.

Group	States of collaboration	States of communication
Chinese Group-A	The two workers participated individually in the work. One worked on the paper and then on the system. The other worked directly on the system.	Opinions were exchanged between the three persons. One worker was self-assertive.
Chinese Group-B	The two workers did not divide the work. One worked on the paper and the other on the system, on the basis of the results of the paper.	The three people collaboratively worked on each line of the specification. They spontaneously exchanged their opinions.
Japanese Group-A	Neither worker was given a share of the work. The leader controlled the work sharing, and the workers only worked on the system.	There were a few sets of good communication between the three participants. The leader identified the orders to the workers for each line of the specification.
Japanese Group-B	The two workers were each assigned a share of the work. One worked on the paper and the other on the system, on the basis of the results from the paper.	One worker was silent while the other actively and successfully communicated with the leader.
JC Group-A	Each of the two workers handled a share of the work. The two first worked on the paper and then on the system.	There were exchanges of opinion between the leader and workers, but only a few between the workers. One worker was self-assertive, while the other asserted negative words.
JC Group-B	Each of the two workers was assigned a share of the work. The two directly worked on the system.	There was communication between the three participants. One worker was self-assertive, and the other asserted less qualitative opinions.

and technical terms such as “chain.” In addition, for the Chinese, their characteristic doubts stemmed from the definition of words, such as “container,” “stored object,” and “deficiency of stocks”; ambiguous representations, such as the “next deficiency of stocks” and “oneself”; and the obscurity of outputs, such as “output a document about deficiency of stocks” and “create a request for delivery of goods.”

The abovementioned Chinese participants had a disadvantage because of the use of Japanese, which is not their mother tongue. The Japanese participants should pay attention to clear communication in the Japanese language to ensure good collaboration.

4.3 Observation of Collaboration

We described a state of collaboration in Table 3 by using a video analysis of the participant’s cooperativeness and communication.

Drawing attention to the Chinese workers, the self-assertiveness of some of them stands out. This might reflect a cultural habit that the Chinese are more individualistic than the Japanese, who prefer homogeneity. Naturally, the cooperative Chinese group created the good chart.

The Chinese Group-A, JC Group-A, and JC Group-B had less learning experience with software specification. From these groups, the JC Group-B, who created a good chart that earned more than sixty percent, worked by having good communication between the three members and by sharing the work on the computer screen. On the other hand, the Chinese Group-A and JC Group-A had to balance the work by creating parts of the chart individually first and then combining them.

Using these observations, the group that scored the lowest points had smaller amounts of work sharing and some of them were self-assertive; that is to say, they did not collaborate in the task. This implies that good collaboration favors the success of a proposed task.

In this experiment, we allowed individual work on paper but urged the usage of the shared environment and groupware technology to encourage shared work. This can be explored by combining the consideration of effectiveness and shared work in software development.

5 CONCLUSION

In this paper, we design a laboratory experiment to simulate offshore software development between the Japanese and Chinese. We performed an experiment in which the participants prepared a system chart instead of program development, using the prepared specification. We considered three types of collaborations, such as an offshore type, domestic type, and a more collaborative type.

The results of these experiments were as follows:

(1) The most collaborative team produced a good model chart.

(2) The Chinese participants faced difficulty in using the Japanese language to communicate, and some of them were self-assertive.

(3) The Chinese participants, compared to the Japanese, tended to be satisfied with their results of the model chart, considering them neither good nor bad.

In the future, we will consider an interface that elicits a collaborative mind or considers the cultural habitat. And, it should be expected to execute the experiments with not natural language but also formal specification language like UML [4]. In addition, more investigations on collaborations in the software development process will be issued.

REFERENCES

[1] H. Tuji, T. Moriyasu and T. Mori, "Evolution of Offshore Software Development and Tacit Knowledge of Engineering," IPSJ Magazine, Vol.49, No.5, pp. 551-557 (2008) (in Japanese).

[2] MIC in Japan, "Report of Progress of Offshoring and its Effect" (2007) (in Japanese).

[3] D. Perry, N. Staudenmayer, and L. Votta, Jr., "Understanding and Improving Time Usage in Software Development," In Trends in Software. Vol. 5, Software Process, A. Wolf and A. Fuggetta, Eds. John Wiley and Sons, New York (1995).

[4] T. Nakahara and H. Fujino, "The Application of UML in the Offshore Project," Journal of the Society of Project Management, Vol. 10, No. 6, pp. 9-14 (2008) (in Japanese).

[5] T. Yamazaki, "Explanation of the Programming Design Method with the Open Problem," IPSJ Magazine, Vol. 25, No. 9, p. 934 (1984) (in Japanese).

[6] Japan Users Association of Information Systems, "A Guideline to Requirement Specification" (UVC) (2007) (in Japanese).

[7] Y. Ooki et al., "Description of Stock Management System by Concurrent Prolog," IPSJ Magazine, Vol. 26, No. 5, pp. 470-476 (1985) (in Japanese).

[8] T. Nakadokoro, "Software Engineering 2nd Edit," pp. 42-46, Asakura-syoten (2004) (in Japanese).

[9] T. Yuizono and J. Munemori, "Groupware for a Knowledge Creative Process of a Research Group and its Application to Externalization and Combination Steps," Journal of IPSJ, Vol. 48, No. 1, pp. 30-42 (2007).

[10] J. Munemori and Y. Nagasawa, "GUNGEN: groupware for a new idea generation support system," Inf. and Soft. Technol., Vol. 38, No. 3, pp. 213-220 (1996).

ACKNOWLEDGEMENTS

This research was partially supported by The Ministry of Education, Culture, Sports, Science and Technology (MEXT) and the Grant-in-Aid for Young Scientists (B) 21700133, 2010.

(Received August 25, 2010)
(Revised April 26, 2011)

Takaya Yuizono received the B.E., M.E., and Dr. of Engineering from Kagoshima University, in 1994, 1996, 1999, respectively. He was a research associate in Kagoshima University, a lecturer and an associate professor in Shimane University, respectively. He has been an associate professor at School of Knowledge Science, Japan Advanced Institute of Science and Technology since 2005. His research interests include in groupware, computer-supported cooperative work and knowledge medium. He received KES2005 Best paper award in 2005. He is a member of ACM, IEEE, IPSJ and IEICE.

Lihua Xuan received the bachelor of Japanese from Dalian Nationalities University in 2007 and master degree of School of Knowledge Science from Japan Advanced Institute of Science and Technology in 2010. Now work in Tagami EX Co.,Ltd. Her interests were cross-cultural collaboration in business.