

*Practical Paper***Awareness Support in Group-based Software Engineering Education System****Kunihiko Chiken* and Atsuo Hazeyama****(Received 1 July 2003 and accepted in revised form 26 November 2003)*

Abstract We have been conducting software engineering education in the form of group learning in a university. We developed a group learning support system and applied it to an actual course. Our experience showed us that it was difficult for some groups to develop software smoothly, because development was often performed in a distributed environment. Thus we understood that awareness to ascertain the situation of the other members (including the teaching staff) was needed. In this paper, we describe how we considered the awareness support requirements in software engineering education, and provided functions to meet the requirements. We applied a group learning support system that provided awareness support to an actual class, and found an action pattern in the activity on the system after using awareness support functions. The results we obtained from questionnaires showed that the awareness support function was also useful in enabling the members to grasp other members' situations.

Keywords: group-based software engineering education, awareness support, group-based learning support system

1. Introduction

Information and communication technologies such as the Internet are permeating into our society rapidly. With the ever-increasing advancement of user needs, the demand for software system development is on the rise along with the increasing complexity of software systems. This has created a need for capable persons who can design and implement software systems; thus, the proper rearing of such persons has become a critical issue.

In business settings, software is generally developed by organizing projects that involve multiple persons. In software development projects that incorporate common goals, project members often engage in cooperative activities to achieve mutual agreement, division of work, and so on. We believe group-based software development learning, in which group members can join in such kinds of cooperative activities, can play an important role in university education.

As a result of this situation, a lot of group-based software engineering education courses have been and are now being conducted⁽¹⁾⁽²⁾⁽⁴⁾⁽⁶⁾⁽¹³⁾⁽¹⁴⁾. The goal of these courses is for students to acquire skills and knowledge with respect to software development through experience. In the courses, students plan and execute a series of software lifecycle process steps (analysis, design, implementation, and testing) performed in business settings for themselves as developers in the form of group learning. On the other hand, the

teaching staff (including the teacher and the teaching assistants) supports students' activities by playing several roles. In our course, we especially focused on the activities of the upstream processes of software development, and on collaborative and cooperative aspects among members⁽⁶⁾. In the exercises done in such courses, however, the workload of both the students and the teaching staff is quite high⁽¹⁾. Therefore, it is essential to provide an efficient learning environment. With this background in mind, we have been developing a group-based software engineering education support system (named "Waltz").

We categorize the main functions the system should provide for group-based collaborative software engineering education as follows:

- (1) Various types of document creation support
- (2) Communication and collaboration support
- (3) Efficient progress monitoring support
- (4) Support of metrication of activities and artifacts stored in the system, and knowledge management support

We have already provided Web-based document creation support by template and file uploading to solve problem (1), asynchronous or synchronous communication support by Bulletin Board System (hereafter referred to as BBS) and chat⁽⁷⁾, and cooperative work support in the inspection and problem resolution processes in acceptance testing between the student groups and the teaching staff to solve problem (2)⁽⁸⁾. We describe an overview of these facilities in Section 3.1. This paper proposes an approach to solve problem (3).

The students are required to spend a lot of time outside

* Graduate School of Education, Tokyo Gakugei University

of the classroom in the exercises. However, it is difficult for the group members to meet together in a face-to-face manner outside of the classroom. Therefore, they often perform their tasks in a distributed manner. In a distributed environment, however, it is more expensive to ascertain the progress among the group members than in a collocated environment. This may lead to delays in development work. It is also difficult for both the members of a group and the teaching staff to communicate in a synchronous manner outside of the classroom.

From the considerations above, we thought the system should provide awareness information necessary to ascertain the status of the members of a group and the teaching staff. By providing awareness information, a student can grasp the progress of both the other members in the same group and the teaching staff without unnecessary efforts. This would also enable the teaching staff to ascertain the status of student groups.

This paper proposes a system that provides awareness information for both members within a group and between the students and the teaching staff. We also show its usefulness from the results of its application to an actual course.

This paper is organized as follows: Section 2 describes related work on awareness support. Section 3 describes an overview of our group-based software engineering course and its support system. After that, we consider requirements for awareness support facilities in the support system in more detail. Section 4 describes the awareness support functions we implemented based on the requirements. Section 5 shows the evaluation results gained from applying them to an actual course. Finally Section 6 concludes this paper.

2. Related Work

Research on awareness support has mainly been conducted in the field of CSCW (Computer Supported Cooperative Work)⁽³⁾.

Kobylynski et al. classified the target of the awareness information into a physical environment and a virtual one⁽¹¹⁾. We focus on learning support in a virtual environment such as that described in Kobylynski et al. Jang et al. developed a system for supporting communication and coordination in a globally distributed design team, called TeamSCOPE. They classified awareness information provided by TeamSCOPE into four types: activity awareness, availability awareness, process awareness and perspective awareness⁽¹⁰⁾. Note that TeamSCOPE only supports activities within groups.

On the other hand, in the education domain this paper focuses on, there exist not only activities within a group of students but also interactions between the groups and the teaching staff. Because software development is knowl-

edge-intensive work and the developers (university students) are novices in our domain, we have found from several years of experience that interactions between students groups and the teaching staff play very important roles. We especially emphasize two types of verification activities: design inspection⁽⁹⁾ and acceptance testing⁽⁹⁾.

Inspection is a well-known quality assurance activity for all software artifacts in the software engineering field. It is an activity to detect defects in the artifacts created by pieces of work. Our developers are students and this is the first experience for most of them to develop software. The course has only three months to finish the project. It is therefore important to detect defects residing in the artifacts as early as possible. This is the reason we emphasize design inspection.

We aim at using software that student groups have developed in practice. This means quality assurance for the final product is important. Our experience has also taught us that students' capabilities in detecting problems are not high in general. This is the reason we emphasize acceptance testing, i.e., formal testing conducted to determine whether a system satisfies its acceptance criteria and thus enable the customer to determine whether to accept the system⁽⁹⁾.

We believe we should deal with not only awareness information in the fine-grained and ill-defined processes that TeamSCOPE provides, but also with awareness information in well-defined processes (e.g., the inspection process or problem management in acceptance testing).

Furthermore a mechanism is required for the teaching staff to easily ascertain the situation of all groups, which are running concurrently. The students want the teaching staff to respond to their requests as soon as possible because of severe time constraints. As such, we think not only awareness information within groups but also awareness information with which the teaching staff ascertains the situation of students groups and awareness information with which a student ascertains the situation of the teaching staff is required.

3. Requirements for Awareness Support in Group-based Software Engineering Education

This section describes an overview of our group-based software engineering course and its support system we have implemented thus far. For details of the course refer to⁽⁶⁾, and for details on the support system refer to⁽⁷⁾⁽⁸⁾. Then we consider requirements for awareness support facilities in the support system.

3.1 Overview of the Group-Based Software Engineering Course and its Support System

3.1.1 Course Overview

Our group-based software engineering course was conducted during the second semester of the third year undergraduate of the Department of Education Informatics in Tokyo Gakugei University. Students who wished to take this course were required to have taken the "Introduction to Software Engineering" course, which was conducted immediately prior to the semester.

Around 25 students took the group-based software engineering course. Since each group had 3-4 members, around six groups were organized. Each group selected one task from the (usually three) tasks the teacher presented. Each group was required to complete software development from scratch within around three months. They were also required to conduct each development process (analysis, design, coding, testing) on their own initiative and to complete product development on their own. Each group was furthermore required to do some sort of project management activities.

In industrial software development projects, various types of intermediate products are created as communication media for use among developers and between developers and clients. We call the intermediate products during the software development process "artifacts". They include requirements specifications, model diagrams in the analysis phase, a system architecture diagram, user interface design documents, functional design documents, data structure design documents in the design phase, and test specifications and test reports in the test phase. In addition, they are required to prepare management reports such as a development plan, progress reports, and meeting minutes.

The teaching staff is involved in the student groups as clients, supervisors, and mentors. More specifically, the teaching staff mainly involve the following activities⁽⁹⁾:

- * Serving as inspectors in the inspection process
Inspections are conducted in our exercises. In the exercises, a teacher and some teaching assistants play the role of inspectors in giving their comments on the artifacts of analysis and design phases. The teacher, playing the main moderator role, returns the results to the group after coordination if necessary. From the viewpoint of education in our inspection, the inspectors not only point out defects but also offer advice and/or suggestions for improvement.
- * Doing acceptance testing
After the student group finishes the system testing, the teaching staff does acceptance testing. When testers detect problems, they record them and notify them to the development group. The development group then investigates the reported problems, modifies the programs

accordingly, and reports the modifications to the tester. The tester confirms what the problems are and judges whether the proper modifications to the program have been made.

- * Providing informal communication such as questions and answers

The teaching staff answers questions raised by students, or provides advice and/or comments to management reports when problems are reported.

3.1.2 Support System

We have thus far provided the following functions to support the above-mentioned activities as a Web application.

(1) Support for student groups

The system provides the following main functions:

- * Template-based document creation support and sharing of the documents

The system provides templates for documents that can be represented only in text format, e.g., development plans and meeting minutes. Group members can view the documents.

- * File uploading function

Documents created in the analysis and design phases usually include not only texts but also diagrams, figures, and tables. We assume they are created with a word processor or with presentation software. The system provides a file uploading function that uploads the document files into the system, thus enabling the group members to view them.

- * Communication support

The system supports asynchronous communication through a normal BBS

(2) Support for the teaching staff

The teaching staff can view activities and the resulting artifacts across groups. Staff members can view messages written in the BBS of all groups and write messages into the BBS of all groups.

(3) Cooperative process support between student groups and teaching staff

- * Inspection process support

The system supports activities in the inspection process as follows:

- The groups register the artifacts for system analysis and design, select those designated as inspection targets, and then request that they be inspected. After that, the group members and the teaching staff are notified via e-mail that the inspection is going to start.
- Each inspector checks the artifacts, comments on them, and judges whether modification is needed for each artifact.
- After all inspectors finish giving their comments, the teacher holds a meeting if necessary. In the meeting, the

teacher refers to all inspectors' comments and decides on a final judgment, i.e., "re-inspection is necessary", "accept with modifications" or "accept", and may provide additional comments on one or more artifacts.

- After the teacher makes a final judgment on all artifacts specified for the inspection session, the system records the status of the inspection session automatically and the result is notified to the group and the teaching staff.

This is one inspection cycle. This process is repeated until the teacher accepts all artifacts.

- * Problem resolution process support in acceptance testing
This function provides support in reporting a problem detected during the acceptance testing to the development group, notifying the result, and reporting whether the problem has been fixed or not. Specific functions are as follows:

- Web-based template for submitting a problem report and having developers return it to the person who submitted it.
- Notification via e-mail when counter-action is initiated
- Tracking progress of each problem report

3.2 Browsing Awareness

In collaborative software development, the developers have to proceed in developing their own work while at the same time keeping track of the progress the group members have made, based on discussions with group members, and/or checking the artifacts group members have created. It is also important for the teaching staff to ascertain the progress of all groups. However, much more effort is needed to ascertain the progress that has been made in a distributed development environment than in a collocated development environment. Therefore, it is necessary to reduce the efforts and costs as much as possible.

We therefore provide awareness information to show who accessed each artifact (not only documents but also BBS messages and inspection comments by the teaching staff) that a group created. We call this awareness information "Browsing Awareness".

3.3 Discussion Awareness

In collaborative work, communication is generally required in various scenes, such as decision-making, the making of announcements or requests, question-and-answer sessions, and so on. This also applies to collaborative software development. We aim at supporting collaborative software development in a distributed environment.

Text or speech exists as a means of conversation in a cyber space. Students in this course do their tasks with computers in various places, e.g., computer rooms, library

in university, or at their home. The system should be standard, i.e., it should be able to be used without any special hardware or software. Accordingly, the system should run on a standard Web server such as Apache, a standard mail server such as SMTP, and a standard Web browser such as Internet Explorer. We decided to adopt text as the means of conversation. Note that we do not insist text is the optimal choice. It is important to persistently manage communication contents so that one can review important decisions and discussions that led to them.

TeamSCOPE provided an electronic bulletin board system called "Message Board", which took advantage of information sharing. However, the results obtained in applying it showed that the team members mainly used e-mail to communicate with each other because of its usability⁽⁶⁾. We provided a standard BBS up until the 2001 class. However, the results we had obtained previously also showed that many groups had used e-mail because of the convenience it provided. Today, most university students in Japan have a mobile phone, the function of which they most often use is the e-mail function. We hope to take advantage of this function in improving communications in the upstream and downstream processes of software development.

When communication is established by using e-mail, managing the information is left to each person. Thus, the information may be scattered. Since e-mail messages cannot be conveyed to members who do not participate in the communication, it is impossible to share opinions and ideas with them. Using mailing lists may solve this problem, but doing so makes it impossible to relate the communication contents to the artifacts.

Thus, we put e-mail messages into the BBS of our support system, which is a kind of repository of software artifacts. We can manage the messages both written in the BBS directly and submitted via e-mail in our support system in an integrated manner. With this function, the members can share the communication contents exchanged by e-mail, so both members who did not participate in the communication and the teaching staff can ascertain what discussions were made. We furthermore think that in software development communication, it is necessary to provide a repository, which manages both the artifacts and the communication contents, and to provide a mechanism that users can associate the artifacts with the message in communication.

As we described in Section 3.2, in group-based collaborative software development, it is necessary for all members of a group to ascertain shared artifacts and to show the facts that they accessed. As the messages in the BBS are objects for browsing awareness, we can ascertain the degree of access for the messages in the BBS.

We call the awareness information needed to ascertain

the group members' opinions and ideas "Discussion Awareness".

3.4 Inspection Progress Awareness

As mentioned before, the teacher places emphasis on the design inspection in the exercises. As such, much effort is needed on the part of both the teaching staff and the groups.

The teaching staff performs the inspections for all groups that are concurrently running. Furthermore, inspections are often repeated several times until an artifact is approved, and so it is difficult for the inspectors to ascertain which artifacts they have given comments to. As all inspectors give comments independently, the teacher must play the role of a moderator in reviewing all the comments of all inspectors to check them for consistency. He must also urge people who arrived late to give their comments. We therefore provided progress-monitoring functions to promote these tasks.

4. Awareness Support Functions


This section describes awareness support functions based on the requirements considered in the previous section.


4.1 Browsing Awareness


The intermediate artifacts created in group-based software development, i.e., the documents, the messages of the BBS, the inspection comments, the problem reports, and so on, have to be recognized by all members of the group. It takes more time and cost to confirm whether group members checked them in a distributed environment than in a collocated environment. It is also unreasonable to assume that the teaching staff will have enough time to ask each student to check whom they have contacted and what comments they have made. Therefore, as Fig.1 shows, we provided a browsing awareness function, which shows whether group members accessed the artifacts from the access log to them. The attributes are the number of accesses, the oldest access time, and the last access time for each member.

インスペクションコメント閲覧状況

グループメンバ	閲覧回数	最古閲覧日時	最新閲覧日時
澤本和憲	2	2002-11-21 18:07:35	2002-11-21 19:08:10
土屋理江	5	2002-11-21 18:07:20	2002-11-21 18:45:58
羽鳥裕美	3	2002-11-21 18:08:03	2002-11-23 21:35:55
齊藤浩一	4	2002-11-21 07:47:05	2002-11-26 17:05:59


**Name of
members**


**No. of
access**


**The oldest
access time**



**The last
access time**

Figure 1 Browsing awareness

4.2 Discussion Awareness

We provided a function to put the contents of e-mail messages into the BBS of our system. Fig.2 shows the function model.

When one member sends an e-mail message to a designated address (1), the message is inserted into the message database of our system through the mail interface (2). The message is then resent to the mail server through the mail interface (3). All the group members are then notified of the message via e-mail sever in a synchronous manner (4). The message inserted to the message database is displayed on the BBS interface (5). By browsing the BBS interface, all the group members can view the message (6). When one member submits a message to the BBS (7), the message is inserted into the BBS database (8) so that all the group members can view it (9). At the same time, the message itself is sent to the mail server through the mail interface (10), and then all the group members are notified of it via e-mail (11).

With this function, the developers can share the contents of the messages when they communicate with the

group via the e-mail system. This function enables the developers to take advantage of both the usability of the e-mail system and the information sharing function of the BBS. We also provided a function to designate an artifact which is associated with a message below the message field of the BBS.

The messages in the BBS are the objects for the browsing awareness. Therefore the system shows who accessed each message and when. As each message has an ID, when a user replies to a message sent from the system by the e-mail system, the system can update the access counter to the original message.

4.3 Inspection Progress Awareness

We provided a function that showed the judgment of each inspector for each artifact as shown in Fig.3. Using this function, the teacher can determine whether all inspectors have finished giving comments or not, and also ascertain the inspectors who are late at a glance. The students can ascertain the progress of the inspection at a glance without browsing the results of each artifact.

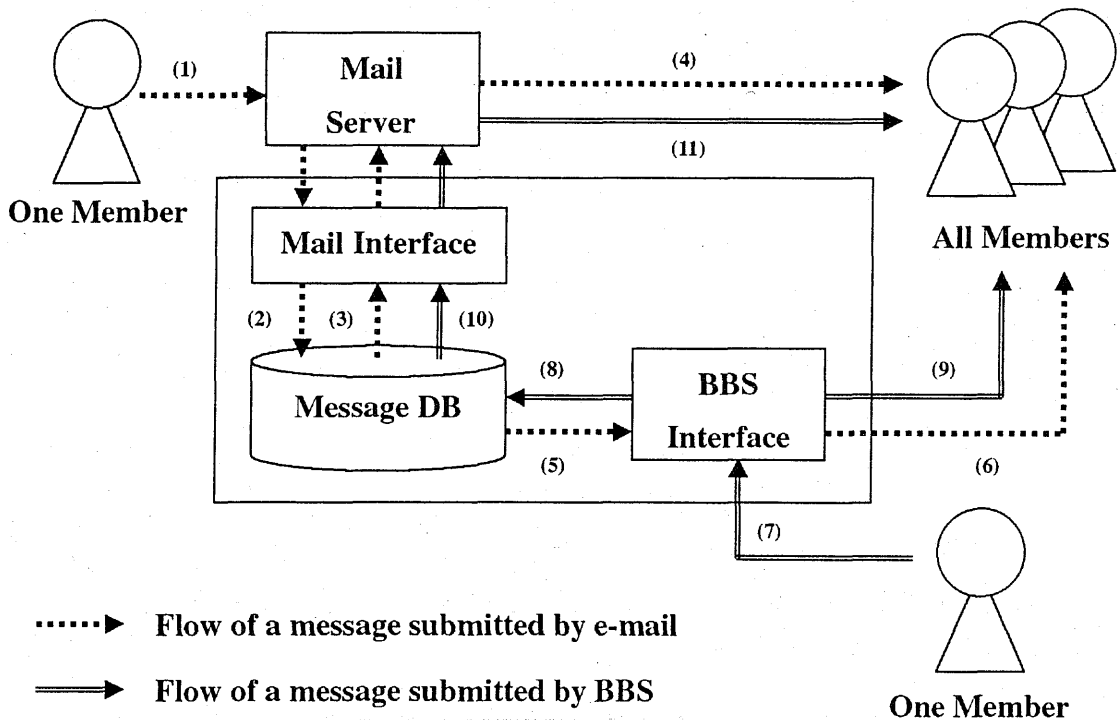


Figure 2 Discussion awareness

5. Evaluation

We applied the system in an actual software engineering course in the 2002 school year for third year undergraduate students in Tokyo Gakugei University's Department of Education Informatics. The 22 students who participated in the course were divided into six groups, and each group was supervised by three members of the teaching staff. This section describes evaluation on each function that provides the awareness information described in the previous section, based on the access logs for each function and the results obtained from questionnaires. The questionnaires listed four ranks, "Very useful", "Useful", "Not very useful" and "Useless", and reasons for choosing them. The response rate for the questionnaires was around 73 percent.

5.1 Activity analysis by using the support system

Table 1 shows the actions the support system provides for students, other than enabling them to view the awareness information.

Table 1 Actions system provides for student

View Plan Report	View Inspection Comment
Create Plan Report	View Group Progress Report
View Meeting Minutes	Create Group Progress Report
Create Meeting Minutes	View Personal Progress Report
View Test Report	Create Personal Progress Report
Create Test Report	View Uploaded File
View Problem Report	Upload File
Create Problem Report	View BBS Tree
View Module Diagram	View BBS Comment
Create Module Diagram	Create BBS Comment

We defined a session as the period of time starting when a user logged into the support system and performed certain actions with it to the time he/she logged out of it. We investigated the "session ratios" defined as the number of times each action is performed to the total number of sessions, because we wanted to know how frequently each action was performed. Fig.4 shows the ratio of the number of sessions conducted that involve one of the relevant actions to the total number of sessions conducted. The figure shows that the usage ratio of BBS was high, especially

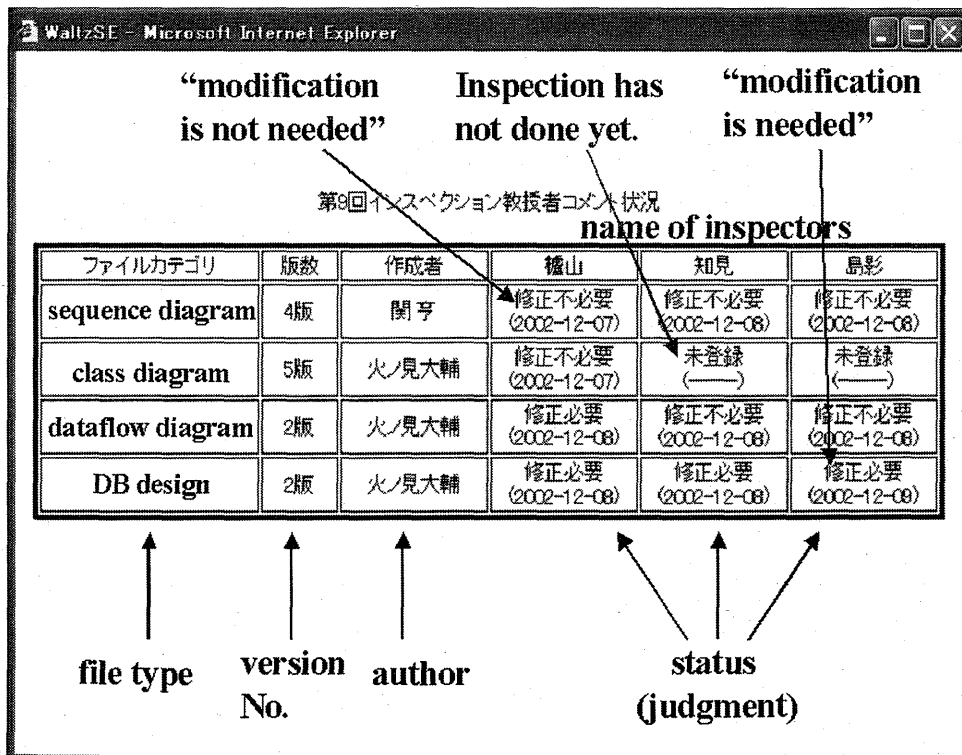


Figure 3 Inspection progress awareness

for the “View BBS Tree”, and “Create BBS Comment” categories. This suggests communication via the BBS is the main activity with this system.

Recently, mobile phones are permeating into our daily life, and with them we can communicate with people anywhere at anytime⁽¹²⁾. University students in Japan are likely to communicate with each other by using the e-mail function on their mobile phone in particularly. The target developers of our study are students. According to usage of the discussion awareness, which enabled submission from e-mail to BBS, around 20 percent of the total messages submitted came from e-mail in the group that used e-mail most. Around 95 percent of the messages posted from e-mail were posted through the e-mail function on a mobile phone. Thus, it can be said that using the e-mail function of a mobile phone almost always enables communication with respect to software development. We implemented discussion awareness that combines e-mail with BBS. This discussion awareness enabled student groups to manage the communication contents in the course of software development.

We performed an experiment to measure the effectiveness of discussion awareness. In the experiment, one party comprised student groups of the 2002 “Exercises of software engineering” course. Another party comprised four groups, each containing 3-4 students of the 2003 “Introduction to software engineering” course. We asked the latter

party to hold discussions on the system analysis and design phases for a task carried out with a normal BBS (i.e., one that does not have a function for providing information to users via e-mail). Fig.5 shows the average response time to messages by three response means; e-mail replies to messages submitted to the Waltz BBS, replies from the Waltz BBS to messages submitted to the Waltz BBS, and replies from a normal BBS to the messages submitted to the normal BBS. The figure shows that the response time by normal BBS is much longer than that with Waltz that supports

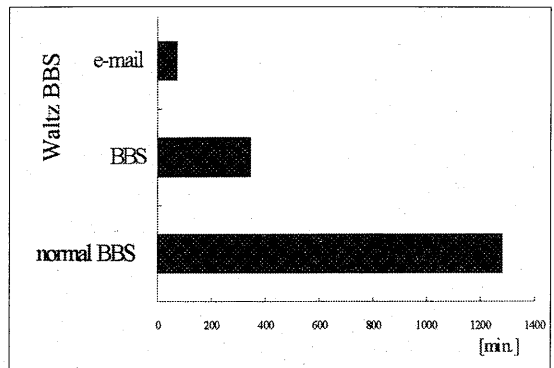


Figure 5 Average response time to a message using Waltz BBS and normal BBS

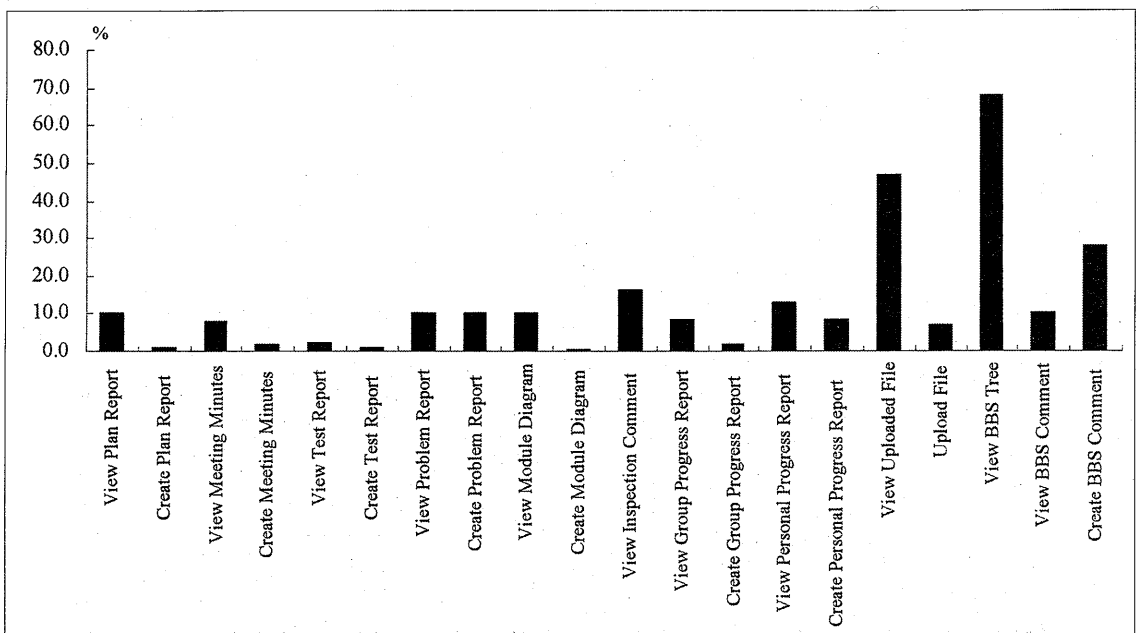


Figure 4 Ratio of each function involved in the session

discussion awareness. It also shows that in Waltz BBS the response time from e-mail was much shorter than that from BBS. From these results, we can say that discussion awareness enabled the communication to be performed quickly.

Tables 2, and 3 show users' (students and the teaching staff) actions and their ratio after using the browsing awareness. Table 4 shows users' (students and the teaching staff) actions and their ratio after using the inspection progress awareness in the same session.

The tables show the five most frequently performed actions and the ratio among them. Since the browsing awareness information was presented with BBS threads, we exclude it from our analysis. In the inspection comments, uploaded files and problem reports, we found that the users tended to access the contents themselves soon after referring to the awareness information, i.e., the action patterns of viewing from awareness information to the contents themselves. Awareness information caused both the students and the teaching staff to use the original functions.

In the browsing awareness, there was little difference with respect to the patterns between the students and the teaching staff. In the "Inspection Progress Awareness", in many cases the students tended to access the inspection comments after referring to the awareness information. On

the other hand, the teaching staff tended to use BBS or to view the uploaded files after referring to the awareness information. The teaching staff plays the inspector role. We think they checked the progress of other inspectors based on this awareness information and then viewed the files uploaded for inspection. BBS, especially "View BBS Tree" and "Create BBS Comment" are often used after accessing awareness information provided by both the browsing awareness and the inspection progress awareness. This result suggests awareness information triggers communication.

5.2 Questionnaire Results on Usefulness of Awareness Functions

We asked both the students and the teaching staff to reply to a questionnaire on discussion awareness, browsing awareness, and inspection progress awareness after finishing the exercises. Table 5 shows the results we obtained.

The results on usefulness of "Browsing Awareness" for the BBS messages, inspection comments and problem reports showed that more than 70 percent of the respondents chose "Very useful" or "Useful" for all functions. They gave various reasons in their comments. For example, regarding the BBS, several students commented that they

Table 2 Major actions and usage ratio after using browsing awareness (students, unit: %)

Browsing Awareness (students)		
Inspection comment	Uploaded File	Problem Report
View Inspection Comment (45.2)	View Uploaded File (33.8)	View BBS Tree (36.4)
View BBS Tree (13.0)	View Problem Report (16.2)	View Problem Report (28.5)
View Module Composition (9.2)	View Inspection Comment (16.2)	View BBS Comment (16.4)
Make Module Composition (9.0)	View BBS Tree (12.4)	View Uploaded File (11.5)
View BBS Comment (8.4)	View BBS Comment (7.1)	Make BBS Comment (3.1)

Table 3 Major actions and usage ratio after using browsing awareness (teaching staff, unit: %)

Browsing Awareness (teaching staff)		
Inspection comment	Uploaded File	Problem Report
View Uploaded File (40.6)	View Uploaded File (48.4)	View Problem Report (38.1)
View Inspection Comment (19.4)	View Problem Report (15.5)	View BBS Tree (20.2)
View BBS Tree (14.9)	View BBS Tree (13.5)	Make Problem Report (11.9)
View BBS Comment (8.5)	View BBS Comment (11.5)	View BBS Comment (11.8)
View Test Report (5.2)	Make Problem Report (4.0)	Make BBS Comment (7.7)

Table 4 Major actions and usage ratio after using inspection progress awareness (unit: %)

Inspection Progress Awareness	
Students	Teaching staff
View Inspection Comment (42.6)	View BBS Tree (28.4)
View BBS Tree (20.0)	View Uploaded File (27.1)
View BBS Comment (10.6)	View BBS Comment (24.9)
View Uploaded File (8.3)	Create BBS Comment (7.4)
View Module Diagram (5.0)	View Personal Progress Report (5.2)

AWARENESS SUPPORT IN GROUP-BASED SOFTWARE ENGINEERING EDUCATION SYSTEM

Table 5 Questionnaire results on usefulness of each type of awareness (unit: %)

	Discussion Awareness	Browsing Awareness				Inspection Progress Awareness
		BBS Message	Inspection comment	Uploaded File	Problem Report	
Very useful	56.25	43.75	43.75	25.00	55.00	43.75
Useful	18.75	31.25	31.25	43.75	30.00	31.25
Not very useful	6.25	18.75	25.00	25.00	10.00	25.00
Useless	18.75	6.25	0.00	6.25	5.00	0.00

could confirm whether other members had accessed the contents they had posted. On the other hand, since the messages posted to the BBS of our system were also forwarded to the e-mail address of the group members, one member pointed out a problem, i.e., that the browsing information with respect to those who had checked the contents submitted to the BBS by means of e-mail was not reflected in the awareness information. Regarding inspection process support, some people wrote, "As the inspection comments have to be recognized, I can impel the members who do not refer to the comments to do so." or "When I modify the artifacts, I can consult with the other members who referred to the inspection comments". Regarding the shared artifacts, some commented that they were able to determine the situations for all members because all of them had to recognize the artifacts before starting the inspection.

We originally intended to provide functions (viewing the awareness information) to ascertain whether other members had accessed the objects or not. However, one student commented, "I used the functions to check whether I had previously accessed the artifacts, because a lot of artifacts were created and I could not remember which ones I had accessed". From such comments, we found the following: in software development, many kinds of intermediate artifacts are created, such as the messages submitted to the BBS, the comments of the inspection, the system analysis and/or design specifications, test specifications, and problem reports. It is therefore difficult for persons to grasp not only whether other members have accessed the artifacts, but also whether they themselves have accessed the artifacts. We found these awareness support functions were useful in enabling users to grasp not only their own situation but that of other members as well.

The results on usefulness of "Discussion Awareness" showed that 75 percent of the respondents thought it was "Very useful" or "Useful". Some students commented on the reasons for their choices, e.g., "I could communicate with the other members even when I was out", and "I could read the comments submitted by e-mail on the group learning support system afterward". One of the teaching staff members commented: "I could ascertain the contents of the e-mail communication among the students as well as messages written on the BBS directly".

The results on usefulness of the "Inspection Progress

Awareness" showed that 75 percent of the people involving the students and the teaching staff felt the function was "Very useful" or "Useful". One teacher in the teaching staff commented, "It is important for me to determine the progress that the inspectors on my staff have made in the inspection and to check their comments, because I have to verify them before making a final decision and passing it on to the group". Many students provided similar comments, i.e., that it was useful to ascertain all the results at a glance.

6. Conclusion

We considered the awareness information necessary for group-based software engineering education, and proposed three awareness functions. Then we provided these three awareness support functions in a group learning support system. We applied the support system to an actual course, and evaluated the functions by means of log analysis and questionnaires. The log analysis results showed that smooth communication was achieved and that the awareness information triggered for certain activities, i.e. the three awareness functions was useful for both the teaching staff and the students. The questionnaires revealed that many students considered the awareness functions to be very useful. Thus we found that the functions were effective in group-based software engineering education.

We will investigate a comparative study by groups with and without the system and clarify effectiveness of the awareness functions.

Acknowledgments

This study was supported by a Grant-in-Aid for Encouragement of Young Scientists (No. A 12780120) from Japan's Ministry of Education, Science, Sports and Culture. The authors wish to thank anonymous reviewers for their comments to improve this paper.

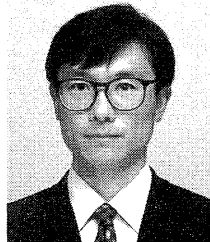
References

- (1) Brown, J.: "Bloodshot Eyes: Workload Issues in Computer Science Project Courses", Proc. of the 7th Asia-Pacific Software Engineering Conference (APSEC2000), IEEE Computer Society Press, pp. 46-53, Singapore (2000)

- (2) Bruegge, B., Dutoit, A. H., Kobylinski, R. and Teubner, G.: "Transatlantic Project Courses in a University Environment", Proc. of the 7th Asia-Pacific Software Engineering Conference (APSEC2000), IEEE Computer Society Press, pp. 30-37, Singapore (2000)
- (3) Dourish, P. and Bellotti V.: "Awareness and Coordination in Shared Workspaces", Proc. of the Computer Supported Cooperative Work, ACM Press, pp. 107-114 (1992)
- (4) Drummond, S., Boldyreff, C. and Munro, M.: "Software Engineering Group Project Work: Past, Present and Future"
- (5) Gilb, T. and Graham, D.: "Software Inspection", Prentice Hall (1993)
- (6) Hazeyama, A.: "An Education Class on Design and Implementation of an Information System in a University and Its Evaluation", Proc. of the 24th Annual International Computer Software and Applications Conference (COMPSAC2000), IEEE Computer Society Press, pp. 21 - 27, Taipei, Taiwan (2000)
- (7) Hazeyama, A., Osada, K., Miyadera, Y. and Yokoyama, S.: "Education Support System of Business Application Software Development", Transactions of Japanese Society for Information and Systems in Education, Vol. 17, No. 3, pp. 251-262 (In Japanese) (2000)
- (8) Hazeyama, A. and Osada, K.: "Web-based Software Engineering Education Environment Supporting Collaboration between the Teaching staff and the Student Groups", Proc. of the First International Symposium on Cyber Worlds: Theories and Practices (CW2002), IEEE Computer Society Press, pp. 323-330, Tokyo, Japan (2002)
- (9) Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: (1990)
- (10) Jang, C. Y., Steinfield, C. and Pfaff, B.: "Supporting Awareness among Virtual Teams in a Web-based Collaborative System: The TeamSCOPE System", ACM SIGGROUP Bulletin, Vol. 21, No. 3, pp. 28-34 (2001)
- (11) Kobylinski, R., Creighton, O., Dutoit, A. H. and Bruegge, B.: "Building Awareness in Global Software Engineering: Using Issues as Context", Proc. of the International Workshop on Global Software Development (2002)
- (12) Maeda, F.: "Multi-Group Awareness on Mobile Communication Environment", IPSJ SIG Technical Report GW-17-1 (In Japanese) (1996)
- (13) Rein, G. L.: "Teaching IS Design and Development in a Group Learning Setting", Proc. of Computer Supported Collaborative Learning (1995)
- (14) Shoenig, S.: "Supporting a Software Engineering Course with Lotus Note," Proc. of the International Conference on Software Engineering Education and Practice (SEE&P1998), IEEE Computer Society Press (1998)



Kunihiro Chiken received the degree of B.A. in Educational Informatics from Tokyo Gakugei University, Japan in 2003. He is now a master course student at Tokyo Gakugei University. His research interests are software engineering and computer supported cooperative work.



Atsuo Hazeyama received the degree of B.E. from Waseda University, Japan in 1985 and the degree of M.E. in Computer Science from the University of Electro-Communications, Japan in 1987. He received his Dr. Eng. from Shinshu University, Japan in March 1999. He joined NEC Corporation in 1987. He is now an Associate Professor at the Department of Mathematics and Information Science, Tokyo Gakugei University. His research interests are collaborative software development support, collaborative learning support, and knowledge management. He was a secretary of SIG-KBSE/IEICE during 1996-1998, and a member of the editorial committee of the IEICE Transactions 1999-2003. He is now a member of the editorial committee of the IPSJ Journal. He is a member of the IEEE Computer Society, the ACM, the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), the Japanese Society for Information and Systems in Education (JSiSE), and the Japanese Society for Artificial Intelligence (JSAI).