Chen, W. et al. (Eds.) (2016). Proceedings of the 24th International Conference on Computers in Education. India: Asia-Pacific Society for Computers in Education

Student Behavior in Computer Simulation Practices by Pair Programming and Flip Teaching

Satoshi V. SUZUKI[a*], Sachio HIROKAWA[b], Shinji MUKOYAMA[c], Ryo UEHARA[d] & Hiroyuki OGATA[d]

[a]*Faculty of Liberal Arts, Osaka University of Economics and Law, Japan*
[b]*Research Institute for Information Technology, Kyushu University, Japan*
[c]*Graduate School of Science and Technology, Seikei University, Japan*
[d]*Faculty of Science and Technology, Seikei University, Japan*
*ssv@svslab.jp

Abstract: Recent education roles encourage willingness to learn individually, solve unfamiliar problems using knowledge acquired through information and communication technologies (ICTs), and collaborate with others. Various learning methods to cultivate this ability have been invented and researchers discussed learning effect on such methods with qualitative and quantitative analyses. One of the authors introduced pair programming as a method of peer learning and flip teaching, which consists of preliminary learning of basic programming and advanced learning practices based on peer activity in classroom lessons, into computer simulation practices for undergraduate students. With introducing class schedule design for flip teaching and development of peer learning preparation support system for determining the appropriate pair formation and seat allocation in the classroom utilizing a probabilistic combinatorial optimization algorithm, this study focuses on learning behavior of well-performing students, analyzing learning records and access logs on a learning management system and answers to a questionnaire administered after the practices. In this analysis, we attempted to discriminate behavior of well-performing students observed from the learning records, access logs, and questionnaire to discover best practices for improving the performance of medium to bottom-line students. Well-performing students tended to prepare for classroom lessons in good time, but their performance depended on the lesson content difficulty. A correlation was observed between the frequency of interaction among students and skill acquisition. We discuss how to improve learning environments using ICTs and collaborative learning methods based on the analysis.

Keywords: CSCL, peer learning, pair programming, flip teaching, learning analytics

1. Introduction

Various recent education opportunities, including higher education, must encourage willingness in students to learn individually, solve unfamiliar problems using knowledge acquired through information and communication technologies (ICTs), and collaborate with others. One of the suggested frameworks required skill sets in contemporary society is known as “21st-century skills” (Griffin, McGaw, & Care, 2012) and includes the skills of metacognition, communication and collaboration, ICT literacy, and understanding of social structures. Moreover, when applying acquired knowledge to solve problems in society, the skills to solve unfamiliar and ill-formed problems are required more than those for well-known and well-formed problems. In fact, learning environments that enable learners to acquire skills to solve “complex, unfamiliar, and non-routine” problems focusing on metacognitive processes, utilization of ICTs, and collaborative learning among diverse learners are developing in mathematical education (Mevarech & Kramarski, 2014). For this reason, providing learning environments that enable learners to learn individually and attempt to solve problems with acquired knowledge is an emergent problem in a dynamically changing contemporary society.

One of the authors introduced pair programming (PP) as a method of peer learning and flip teaching (FT) to enable students to acquire the basis of such skills into computer simulation practices for undergraduate students. PP is a method of collaborative software development in which one pair member assumes the role of driver, inputting the source code, and the other assumes the role of navigator, asking questions and suggesting programming ideas; these roles are then swapped (Beck & Andres, 2005; Williams & Kessler, 2002). FT is a method of blended learning that requires students to learn basic knowledge via video lessons to prepare for classroom lessons requiring students to solve advanced problems, providing opportunity to solve advanced problems with help (Bergmann & Sams, 2012; DeLozier & Rhodes, 2016). There are two purposes of FT dependent on a learning goal: mastery of knowledge and skills aiming the same goal, among students and acquisition of higher-order knowledge skills in classroom based on video lessons. Lucke (2014) claimed that there was no significant correlation between the time to spend to learn learning content via video lesson and learning performance of students, although the students tended to prefer the flip teaching class to traditional teaching. However, the analysis by Lucke (2014) only focused on comprehensive variables of student performance such as grade point average (GPA), total time spent on learning via video lesson through the course, total score of quizzes in the lessons and the score of final examination. Learning performance of students through flip teaching should be dependent on various aspects of learning behavior inside and outside the classroom. We attempted to discover student learning behavior in the computer simulation practices, analyzing students' behavior as observed on a learning management system (LMS) and through a questionnaire after the lessons. We aim to suggest ideas to improve learning environments considering middle and bottom-line students based on discovered best practices of well-performing learners.

In this paper, firstly, we review the studies of collaborative learning mainly focusing on peer learning and development of ICT-enhanced learning environment mainly for science and technology. Secondly, we explain the purpose, schedule and routine of computer simulation practices. Thirdly, we describe how to collect and analyze the data regarding to learning behavior of the students. Finally, we discuss best practices of well-performed learners based on the result of the analysis.

2. Related Work

2.1 Peer Learning

Peer learning is a popular form of collaborative learning in which two students interact to work on a learning activity. Many studies in psychology, cognitive science, and learning science suggest the advantage of peer learning. Miyake (1986) claims that the role of a task-doer and an observer in peer learning and changing these roles between learners promote detailed understanding of learning content in each learner. In addition, requiring explanation of learning content for each other also induce deep understanding of learning content of explainer (Okada & Simon, 1997). Moreover, externalization of thought of one of the pair induces awareness of overlooked problems in the learner who externalized her/his thought by being pointed out the problems by the other (Shirouzu, Miyake, & Masukawa, 2002). In fact, various education practices introduce its use. For example, some studies on development of learning support systems for academic writing introduce peer learning function (e.g. peer review of draft among learners) suggest that interaction between learners improves learners' writing quality (Cho & Schunn, 2007; Suzuki & Suzuki, 2013). As introduced in Section 2.2, PP is often regarded to productive method of team software development. However, PP in learning environment has potential to promote deep understanding for programming of students considering the discussion above.

2.2 Pair Programming as Peer Learning

The main advantage of PP, as explained in Section 1, is found in software engineering productivity, but it can be a method of peer learning. Fundamentally, PP is an important activity in extreme programming, an agile software development framework (Beck & Andres, 2005). One of its purposes is to promote communication among development team members (Williams & Kessler, 2002). One experiment implied that a novice–novice pair could improve productivity by PP over an expert–expert pair (Lui & Chan, 2006). Another experiment suggested that the improvement of productivity in PP depends on programming knowledge and skills of a programmer and complexity of PP tasks (Arisholm, Gallis, Dybå, & Sjøberg, 2007). A conversational analysis of novice–novice PP suggests that the pairs who can properly explain programming and frequently provide mutual explanations between them can improve the quality of their programs (Inoue, 2013). These studies suggest that novice–novice PP can promote understanding of programming, arranging the learning environment to promote mutual interaction between learner pairs.

2.3 Learning Environment Utilizing ICTs

Arranging learning environments to motivate learners to voluntarily acquire knowledge and skills and to promote interaction among learners using ICTs is an important problem. Recent studies of learning analytics aim to improve learning environments based on analysis of learning activities (Ferguson, 2012). Many educational practices have been introducing FT, as explained in Section 1, to promote voluntary learning activity. From the perspective of successful collaborative learning, group formation plays an important role. In fact, some studies have attempted to decide automatically on proper group formation using probabilistic optimization algorithms, clustering algorithms, and multi-agent simulation (Cruz & Isotani, 2014). This study also attempted to automatically decide on pair formation and seat allocation based on student learning performance using probabilistic optimization; we explain this procedure briefly to describe the analysis of learning behavior and to discuss the approach to improving learning environments including pair formation and seat allocation.

3. Class Design

One of the authors introduced PP and FT into computer simulation practices dealing with MATLAB programming and basic thoughts on computer simulation. The practices were part of a course called "Basis of the Simulation" for science and technology undergraduate students. Eighty-five students took the course, and two teachers, including one of the authors (with no teaching assistant), taught in the practices. The students had already studied the basis of C programming, but this was their first time learning MATLAB. We introduced Moodle[1] as an LMS in the practices. Considering the purpose of FT introduced in Section 1, FT in the practices was aimed at student mastery of MATLAB programming skills and application of the programming skills to computer simulation.

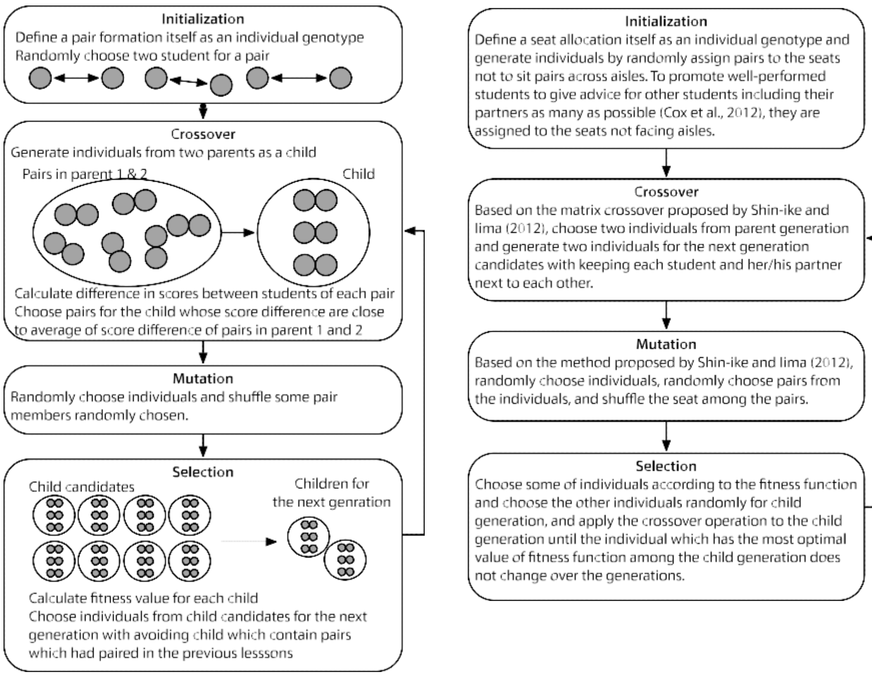
3.1 Probabilistic Optimization of Pair Combination and Seat Allocation in Classroom

Proper pair formation and seat allocation of students promote the learning performance of students. Some methods to manually determine pair formation have been already suggested (Johnson, Johnson, & Holubec, 2009). The role of groups in collaborative learning mainly depends on tasks which the groups should work on (Barkley, Cross, & Major, 2005; Johnson et al., 2009), however, we focused on formation of the informal groups (Johnson et al., 2009) which work on the tasks which the groups can finish within each lesson. Additionally, changing pairs in PP can improve programming efficiency given several opportunities for PP (Williams & Kessler, 2002). Moreover, we pointed out in Section 2.2 that novice–novice PP can facilitate acquiring programming knowledge and skills. Besides programmer personality and temperaments (Sfetsos, Stamelos, Angelis, & Deligiannis, 2009), extreme programming knowledge and skill differences within a pair can increase troubles in programming activities, and those with inadequate programming knowledge and skills will have difficulty interacting to solving problems (Arisholm et al., 2007; Williams & Kessler, 2002). Taking the discussion above into account, pair formation was decided to minimize variance of differences within pairs in programming knowledge and skills using a genetic algorithm (GA) (Goldberg, 1989) shown in the left side of Figure 1, one of the popular methods of probabilistic combinatorial optimization, based on the results of prior C programming tests and quizzes before each lesson. We adopted pair formation itself as a genotype and developed crossover operation among the population of the parent generation based on

[1] http://moodle.org/

edge recombination crossover used to solve a travelling salesman problem (Whitley, Starkweather, & Fuquay, 1989).

Furtherclassroom seating behavioror influences learning performance. A correlation exists between classroom position and learning willingness and performance (Becker, Sommer, Bee, & Oxley, 1973). Seat arrangement with well-performing students allocated to the seats promotes teaching of students close to them (Cox, Cody, Fleming, & Miller, 2012). Taking these studies and Section 2.2 into account, we decided on the seat allocation to enable students to be seated in different positions in the classroom and to enable students who frequently taught reciprocally in the classroom to be seated close to each other. Seat allocation was also decided to use a GA shown in the right side of Figure 1, adopting seat allocation itself as a genotype and adopting crossover and mutation operation on matrices suggested by Shin-ike and Iima (2012) on population of parent generation.



Define a pair formation itself as an individual genotype
Randomly choose two student for a pair

Generate individuals from two parents as a child

Calculate difference in scores between students of each pair
Choose pairs for the child whose score difference are close to average of score difference of pairs in parent 1 and 2

Randomly choose individuals and shuffle some pair members randomly chosen.

Calculate fitness value for each child
Choose individuals from child candidates for the next generation with avoiding child which contain pairs which had paired in the previous lesssons

Define a seat allocation itself as an individual genotype and generate individuals by randomly assign pairs to the seats not to sit pairs across aisles. To promote well-performed students to give advice for other students including their partners as many as possible (Cox et al., 2012), they are assigned to the seats not facing aisles.

Based on the matrix crossover proposed by Shin-ike and Iima (2012), choose two individuals from parent generation and generate two individuals for the next generation candidates with keeping each student and her/his partner next to each other.

Based on the method proposed by Shin-ike and Iima (2012), randomly choose individuals, randomly choose pairs from the individuals, and shuffle the seat among the pairs.

Choose some of individuals according to the fitness function and choose the other individuals randomly for child generation, and apply the crossover operation to the child generation until the individual which has the most optimal value of fitness function among the child generation does not change over the generations.

Figure 1. Applying GA to find approximate solution of optimal pair formation (left) and seat allocation (right).

3.2 *Lesson Schedule*

Lesson 1 Orientation and basic usage of MATLAB (in-class tasks)
- Take-home C programming test to diagnose programming knowledge and skills

Lesson 2 Basis of operation of vectors and matrices, drawing graphs (video lesson) / Solution of a quadratic equation, graphing a trigonometric function (in-class tasks)
Lesson 3 Matrix definition and modification (video lesson) / Dealing with matrix as table data, solution of Monty Hall problem with Monte Carlo technique (in-class tasks)
Lesson 4 Operation of matrices (video lesson) / Various expressions of Fibonacci sequence, linear transformation of figures (in-class tasks)
Lesson 5 Loop control, conditional branching, and computation time (video lesson) / Summation of sequence, solution of equation by bisection method, rectangle method of numerical integration (in-class tasks)
- No homework

Lesson 6 Comprehension test and questionnaire

Figure 2. Lesson schedule in the class.

Figure 2 shows the lesson schedule. Students were required to watch video lessons on basic MATLAB programming knowledge and to take a quiz about the lesson before each classroom lesson. The classroom consisted of a traditional row-and-column arrangement with a PC on each seat. We decided on pair formation and seat allocation as explained in Section 3.1 except for in Lessons 1 and 6. We assigned the pairs four in-class tasks, and pair members changed driver and navigator roles after each task. Students submitted worksheets on program execution results and discussed the results for each task via Moodle. One of the authors rated these worksheets. Students were allowed to talk with other students about the tasks with leaving their seats to promote student understanding of lesson content without excess intervention by teaching staffs.

4. Analysis of Student Behavior

We attempt to discuss student behavior based on an analysis of the relationship between comprehension test scores and learning behavior as observed in LMS access log and learning records and a follow-up questionnaire. Collecting data for the analysis from various aspects of learners is needed to deeply investigate tendency of the learners' behavior for improving learning environment (Ferguson, 2012). In fact, Aguiar et al. (2015) attempted to analyze demographic data, attendance and tardiness of classes, and academic achievement of high school students to detect candidates of dropouts. These studies of learning analytics imply that knowledge and skills of students acquired through the lessons can be predicted based on learning behavior such as attendance and tardiness of classes, academic achievement related to the class and in-class learning performance. To extract the relationship between the comprehension test scores and the learning behavior of the students, we attempted to find well-performing students using learning behavior data with a support vector machine (SVM). We also attempted to extract what kinds of learning behavior contribute to increased performance in well-performing students through feature selection, applying a feature word extraction algorithm originally developed for information retrieval (Sakai and Hirokawa, 2012). Additionally, we inspected the tendency in learning behavior of well-performing students observed in the answers of questionnaire by contingency table analysis.

4.1 LMS Access Logs

Diagnosis test score Score on take-home comprehension test on C programming after Lesson 1
Attendance Frequency of attendance
Tardiness Frequency of tardy
Tardy time Length of time between lesson start and login time on classroom PC
Quiz submission Frequency of quiz submission before each lesson
Seating as specified Frequency of sitting in the classroom as we specified (Some students could not obtain specified seats because of tardiness and their intention not to work with others)
Pairing as specified Frequency of pairing as we specified (same as above)
Quiz score Score on the quiz before each lesson
Score difference compared to partner Difference in scores on quizzes and in-class tasks within the pair
Self-evaluation of in-class tasks Self-evaluation of understanding in each in-class task (1: not at all–4: totally achieved. We adopted the simplified criteria based on Hahn, Mentz, and Mayer (2009) for lesson schedule constraint)
Self-evaluation of pair performance Self-evaluation of task performance as a pair in each in-class task (1: not at all–4: totally achieved; same as above)
Login outside classroom Login frequency on an LMS outside the classroom
In-class video access In-class video lesson access frequency
Video access outside classroom Video lesson access frequency outside classroom
In-class slide access Frequency of in-class access frequency of slides used in the lessons
Slide access outside classroom Frequency of access frequency of slides used in the lessons outside classroom
Quiz finish time Length of time between quiz submission and lesson onset time
In-class task finish time Length of time between finish time of each lesson and submission time of a worksheet on the fourth task of each lesson
Solo tasks Frequency of in-class task worksheet submission as solo work (Some students worked on these tasks alone because of their intention not to work with others.)
Pair performance difference between pair members Difference in self-evaluation of pair performance between the pair
Login outside the university Frequency of login on an LMS outside the university

Figure 3. Variables extracted from LMS access logs and learning records.

Taking into account the previous studies of learning analytics to predict learning performance based on student behavior (Aguiar et al., 2015; Ferguson, 2012), we extracted variables from access logs and learning records on Moodle on Figure 3. We calculated a median for the variables, coding 1 if each student obtained a variable higher than the median and 0 if each student obtained a variable lower than the median using the 78 students who attended Lesson 6.

4.2 Questionnaire After Comprehension Test

Figure 4 shows the content of a questionnaire after the comprehension test in Lesson 6. The answer was collected via Moodle and coded 1 if the students chose each option and 0 if they did not in **Q1**, **Q2**, **Q5**, and **Q6**. The answers of the "Yes" or "No" questions were coded 1 if the students chose "Yes" and 0 if they chose "No."

Q1 Did you work on the pair tasks when there were large gaps in programming knowledge and skills between you and your partner?
1. You worked on the task without collaborating with your partner.
2. You unilaterally taught the task solution to your partner.
3. You collaborated with your partner and reciprocally taught to each other, and you deeply understand the lesson content.
4. You collaborated with your partner and reciprocally taught to each other, but some of the lesson content was too hard for you to understand.
5. It seemed that there were no large gaps in programming knowledge and skills.
Q2 Did you work on the pair tasks when there were small gaps in programming knowledge and skills between you and your partner?
1. (The same as option 1. in Q1.)
2. (The same as option 2. in Q1.)
3. (The same as option 3. in Q1.)
4. (The same as option 4. in Q1.)
5. It seemed that there were no small gaps in programming knowledge and skills.
Q3 Did you talk with ex-partners from previous lessons about the tasks?
1. Yes 2. No
Q4 Did you talk with friends that you ordinary talk to about the tasks?
1. Yes 2. No
Q5 How did you feel about your specified seat in the classroom?
1. Your seat seemed relatively fixed in a certain location of the classroom.
2. There were variations in the location of your seat in the classroom.
3. You cannot answer this question because you rarely had a seat specified in class.
Q6 How did you feel about the students around your seat in the classroom?
1. There were many students familiar to you aside from your partner.
2. There were many students unfamiliar to you aside from your partner.
3. You cannot answer this question because you rarely had a seat specified in class.
Q7 Did you talk with other students near your seat?
1. Yes 2. No
Q8 Did you talk with other students far from your seat?
1. Yes 2. No

Figure 4. Questionnaire after comprehension test in Lesson 6.

4.3 Discrimination of Well-Performing Students Using SVM

We defined well-performing students as students those scores were higher than the median of the score in all students. Thus, the score of each student was coded 1 if the score was higher than the median (34 students), and 0 if the score is lower than the median (38 students). Four students whose scores were is equal to the median was were omitted for the analysis.

First, we constructed a 74 (students) × 61 (variables) matrix with coded data described in Section 4.1 and 4.2. Next, we regarded this matrix as a document matrix with each student corresponding to an individual document and each variable to an indexed term to apply the method of feature word extraction algorithm (Sakai and Hirokawa, 2012). The matrix was handled on GETA[2] for high-speed computation in the same way as Sakai and Hirokawa (2012). Based on the matrix, we attempted to classify the well-performing students and the ill-performing students using SVM^{perf} (Joachims, 2006) for model construction, classification, and default parameter setting. While using SVM, linear kernel was adopted according to default settings of SVM^{perf}. We conducted five-fold cross validation and obtained the following values: precision was 0.9173, recall was 0.9492, F-measure was 0.9306, and accuracy was 0.8710. The model is adequately useful for the classification of the students because these values were sufficiently high.

We applied the feature word extraction algorithm (Sakai and Hirokawa, 2012) to the matrix in the following procedure, 1. The matrix was constructed with all variables, and SVM was applied to the matrix to construct a model for classification. 2. The matrix was constructed with each variable w_i (i=1, 2,…, 61) and calculated the positive and negative weight (distance from estimated separated hyperplane to each instance in the SVM model) of the variable *weight*(w_i), 3. The variables were chosen for model construction by SVM, and the optimal number of variables was calculated. We compared the following six criteria to choose variables considering the values from cross validation which was the most clearly discriminate well-performed students with manipulating the number of variables to choose: *weight*(w_i),

[2] http://geta.nii.ac.jp/

$weight(w_i)df(w_i)$, $weight(w_i)\log df(w_i)$, and the absolute values of these variables ($df(w_i)$ is a frequency of 1 in all students for the variable w_i) based on precision, recall, F-measure and accuracy. As a result, we adopted $|weight(w_i)df(w_i)|$ as the criterion, and we chose $2N$ positive and negative variables whose $|weight(w_i)df(w_i)|$ was ranked in the top $2N$ of all positive and negative variables (when N=5, precision was 0.9729, recall was 0.8905, F-measure was 0.9245, and accuracy was 0.8702).

Figure 5. Positive (+) and negative (−) variables chose in the analysis.

1. − **Quiz score** before Lesson 5
2. + **Diagnosis test score**
3. − **Tardy time**
4. + **Performance difference between pair members** in Lesson 4
5. − **Score differences between pair members** in Lesson 3
6. + **Quiz score** before Lesson 3
7. + **Score difference between pair members** in Lesson 4
8. − **Performance difference between pair members** in Lesson 5
9. + **Performance difference between pair members** in Lesson 3
10. + **Quiz score** before Lesson 4

We obtained the positive (+) and negative (−) variables shown in Figure 5. The list in Figure 5 is sorted in descending order of $|weight(w_i)df(w_i)|$. Positive variables (+) indicate that the higher the student variables were, the higher their comprehension test scores were. Negative variables (−) indicate that the lower the student variables were, the higher their comprehension test scores were. For example, "− **Quiz score** before Lesson 5" indicates that, the lower the quiz score before Lesson 5 was, the higher the comprehension test score was.

4.4 Contingency Table Analysis

We attempted to examine the tendency of questionnaire answers after the comprehension test in Lesson 6 by summarizing the answers to each question in a contingency table. Student group U_d indicates the students whose **diagnosis test scores** in Section 4.1 were coded as 1, and student group L_d indicates the students whose **diagnosis test score** were coded as 0. Student group U_c is equivalent to a group of the well-performing students, and student group L_c is equivalent to a group of the ill-performing students explained in Section 4.3. Student group (X, Y) indicates the group of students who belong to both X and Y (for example, the students in the group (U_d, U_c) belong to both U_d and U_c). Table 1 shows the tables that indicate a clear answer tendency, and the results of Fisher's exact test suggested the tendency of significance in the difference of students' answers depending on the student groups (**Q4**: p=. 071; **Q5**: p=. 054). The results slightly imply that the students in the L_d group tended not to talk about the tasks with friends in class, and the students in the L_c group tended not to have a seat specified in class.

Table 1: Number of students who chose each option in **Q4** and **Q5**.

Q4 answers	(U_d, U_c)	(U_d, L_c)	(L_d, U_c)	(L_d, L_c)
Yes	18	15	6	13
No	3	1	4	7

Q5 answers	(U_d, U_c)	(U_d, L_c)	(L_d, U_c)	(L_d, L_c)
1.	4	3	5	1
2.	16	9	5	16
3.	1	4	0	3

5. Discussion

From the results of the search for well-performing students in Section 4.3, well-performing students showed good performance in the diagnosis test and arrived to the classroom early before the beginning of the lesson. This result suggests that students well-prepared with programming knowledge and skills who attended each lesson from the beginning should perform well. Additionally, the results of contingency table analysis of **Q5** imply that some ill-performing students tend to sit against specifications because of tardiness and unwillingness to working on peer learning tasks. The results of contingency table analysis of **Q4** suggest that students who performed poorly on the diagnosis test tended to avoid interaction with others. This can be because such students were reluctant to interact with others because of lack of programming knowledge and skills and verbalization of the programming question. These results imply that the environment in which students acquire sufficient programming knowledge and skills to attend this course is crucial. Moreover, successful classes including FT should be designed to enable students to avoid tardiness and finish preparation for class learning with video lessons (Bergmann & Sams, 2012). One of the important class design features for FT is to separate in-class content for students who finished class preparation and for students who arrived late and/or did not prepare. Motivating students to sufficiently prepare for class by improving class design should be important for successful FT practices.

Analysis results also implied that learning behavior of well-performing students should depend on learning content in each lesson. In preparation for in-class tasks, well-performing students showed better performance in Lessons 3 and 4 and worse performance in Lesson 5 than did ill-performing students. Compared to their partners, well-performing students showed better in-class task performance in Lesson 4 and worse performance in Lesson 3 than did ill-performing students. In self-evaluation of pair performance compared to the partner, well-performing students showed better performance in Lesson 3 and worse performance in Lesson 5. We examined differences in **In-class task finish time** among lessons using one-way ANOVA (within-participant), and found a significant main effect (F (3, 231) =19.93, p<.001, η^2=0.206); the finish time in Lessons 2 (M=10.62, SD=6.76) and 5 (M=8.73, SD=11.79) were longer than in Lessons 3 (M=2.83, SD=3.31) and 4 (M=4.56, SD=4.66) according to multiple comparisons using Holm's method. This means that the in-class tasks in Lessons 2 and 5 are harder for students than those in Lessons 3 and 4 and suggests that the difficulty of in-class tasks affected the difference in learning behavior of well-performing students. Well-performing students tended to show better performance in preparation for in-class tasks and self-evaluation of pair performance. Nevertheless, a different tendency of in-class task performance in Lessons 3 and 4 was observed in well-performing students. While inductive reasoning of execution results was mainly required for students in Lesson 3, utilizing knowledge of linear algebra was mainly required in Lesson 4. The difference of tendency in the in-class task performance represents a similar type of required skills. The results also suggest that consistency in learning content design is needed because different lesson content policies can confuse understanding and self-evaluation of learners.

As explained in Section 3.1, we attempted to determine pair formation focusing on the knowledge and skill differences in programming between each student pair and seat allocation focusing on the history of seat positions in previous lessons and friendship among the students using GA. The analysis suggested that the knowledge and skill differences in programming between each student pair can influence learning performance in computer simulation. On the other hand, little influence of seat position was observed from the analysis of the questionnaire. It is difficult to find policies to modify the method to determine the pair formation and the seat allocation through the result of the analysis in this study. As well as calculation performance evaluation of the algorithms suggested in this paper by comparing other probabilistic optimization algorithms suggested by Cruz and Isotani (2014), improving the algorithms with considering what kinds of data and evaluation function for optimization are effective for pair formation and seat allocation should be investigated through learning analysis.

The discussion above contains some speculations, because we could not record the detailed process of interaction among students. Collecting data to comprehend student behavior in detail is needed, such as questions with free descriptive answers and student interviews. In addition, Inoue (2013) analyzed within-pair conversation in PP as peer learning in detail. Similarly, analysis of interaction not only with the PP partner but also other students is required for further discussion. Furthermore, various classroom settings should be taken into account in class design. We used traditional row-and-column seat arrangement in this study; however, various class designs are considered for facilitating peer learning, including the use of ICTs and seat arrangements (Prensky, 2010). Possibilities to improve classroom environment and the lesson content design should be discussed further based on student behavior analysis.

Furthermore, the results of the analysis and the appropriate method to determine pair/group formation and seat allocation should be depend on the number of teaching staff, the number of students, the number of lessons conducting flip teaching, learning content, and students' readiness and motivation for the learning content as well as the classroom environment. Despite the increasing number of practices of flip teaching (Bergmann & Sams, 2012; DeLozier & Rhodes, 2016), there were few attempts to measure learning effect in such practices with a learning analytics approach (for example, Lucke, 2014). Evidence-based improvement of the learning environment is needed with the discussion on various aspects of data observable in the learning environment utilizing ICTs.

To improve the learning environment, especially for ill-performed students based on best practices of well-performed learners, we attempted to discover general tendency of learning behavior of well-performing students from discrimination of their data using SVM. However, there can be various learning processes among well-performing students as well as ill-performing students. Recent studies suggest a skeptical view on the existence of learning styles and the effect of adaptive learning environment, according to the learning styles (for example, Cuevas, 2015). Nevertheless, investigating several possibilities to succeed and fail in learning via flip teaching and peer learning based on student learning behavior should be valuable for finding hypotheses of relationship between learning behavior and learning performance of the students in flip teaching.

6. Conclusion

This study reported computer simulation practices adopting PP and LT, analyzed the students’ learning behavior based on LMS records and learning performance, and discovered the learning behavior of well-performing students. Allowing for learning environment design to encourage a willingness to learn voluntarily especially for ill-performed students, to learn utilizing ICTs, and to collaborate with others in learning activities, learning environments should be explored further based on the analysis of learning behavior of students.

Acknowledgements

We would like to express our gratitude to Kohei Yuge (Seikei University) and Kiiko Katsuno-Matsumoto (Seikei University) for the arrangement of the class, and Japan Society for Educational Technology SIG-03 (Development and Practice of Education and Learning Management Systems) for providing the opportunity to collaborate among the authors and deeply discuss this study. The research is partially supported by “Research and Development on Fundamental and Utilization Technologies for Social Big Data” (178A03), the Commissioned Research of the National Institute of Information and Communications Technology (NICT), Japan; Grant-in-Aid for Scientific Research (S), 16H06304; the Education Enhancement Program (EEP) of Kyushu University; and Grant-in-Aid for Scientific Research (C), 24501147.

References

Aguiar, E., Lakkaraju, H., Bhanpuri, N., Miller, D., Yuhas, B., & Addison, K. L. (2015). Who, when, and why: A machine learning approach to prioritizing students at risk of not graduating high school on time. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge* (*LAK '15*) (pp. 93–102). Poughkeepsie, NY, USA. doi: 10.1145/2723576.2723619

Arisholm, E., Gallis, H., Dybå, T., & Sjøberg, D. I. K. (2007). Evaluating pair programming with respect to system complexity and programmer expertise. *IEEE Transactions on Software Engineering*, *33*(2), 65–86. doi: 10.1109/TSE.2007.17

Barkley, E. F., Cross, K. P., & Major, C. H. (2005). *Collaborative learning techniques: A handbook for college faculty*. San Francisco, CA, USA: Jossey-Bass.

Beck, K., & Andres, C. (2005). *Extreme programming explained: Embrace change* (2nd ed.). Boston, MA, USA: Addison-Wesley.

Becker, F. D., Sommer, R., Bee, J., & Oxley, B. (1973). College classroom ecology. *Sociometry*, *36*(4), 514–525.

Bergmann, J., & Sams, A. (2012). *Flip your classroom: Reach every student in every class every day*. Alexandria, VA, USA: International Society for Technology in Education.

Cho, K., & Schunn, C. D. (2007). Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers and Education*, *48*(3), 409–426.

Cox, J., Cody, J., Fleming, J., & Miller, M. (2012). Seat assignment contribution to student performance in an information technology classroom. In *2012 ASEE Northeast Section Conference*. Lowell, MA, USA.

Cruz, W. M., & Isotani, S. (2014). Group formation algorithms in collaborative learning contexts: A systematic mapping of the literature. In *Proceedings of the 20th International Conference on Collaboration and Technology* (*CRIWG 2014*) (pp. 199–214). Santiago, Chile: Springer International Publishing. doi: 10.1007/978-3-319-10166-8_18

Cuevas, J. (2015). Is learning styles-based instruction effective? A comprehensive analysis of recent research on learning styles. *Theory and Research in Education*, *13*(3), 308–333. doi: 10.1177/1477878515606621

DeLozier, S. J., & Rhodes, M. G. (2016). Flipped classrooms: A review of key ideas and recommendations for practice. *Educational Psychology Review*, online, 1–11. doi: 10.1007/s10648-015-9356-9

Ferguson, R. (2012). Learning analytics: Drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, *4*(5/6), 304–317. doi: 10.1504/IJTEL.2012.051816

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA, USA: Addison-Wesley.

Griffin, P. E., McGaw, B., & Care, E. (2012). *Assessment and teaching of 21st century skills*. Dordrecht, the Netherlands: Springer.

Hahn, J. H., Mentz, E., & Meyer, L. (2009). Assessment strategies for pair programming. Journal of Information Technology Education, *8*, 278–289.

Inoue, T. (2013). Investigating the relation between behavior and result in pair programming: Talk and work leads to success. *The Journal of Information and Systems in Education*, *12*(1), 39–49. doi: 10.12937/ejsise.12.39

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD '06*) (pp. 217–226). Philadelphia, PA, USA. doi: 10.1145/1150402.1150429

Johnson, D. W., Johnson, R. T., & Holubec, E. J. (2009). *Circles of learning: Cooperation in the classroom* (6th ed.). Edina, MN, USA: Interaction Book Co.

Lucke, T. (2014). Using learning analytics to evaluate the effectiveness of the flipped classroom approach. In *Proceedings of the 2014 Australalasian Association for Engineering Education Conference* (pp. 1–9). Wellington, New Zealand.

Lui, K. M., & Chan, K. C. C. (2006). Pair programming productivity: Novice-novice vs. expert-expert. *International Journal of Human-Computer Studies*, *64*(9), 915–925.

Mevarech, Z., & Kramarski, B. (2014). *Critical maths for innovative societies: the role of metacognitive pedagogies*. Paris, France: OECD.

Miyake, N. (1986). Constructive interaction and the iterative process of understanding. *Cognitive Science, 10*(2), 151–177.

Okada, T., & Simon, H. A. (1997). Collaborative discovery in a scientific domain. *Cognitive Science*, *21*(2), 109–146.

Prensky, M. (2010). *Teaching digital natives: Partnering for real learning*. Thousand Oaks, CA, USA: Corwin.

Sakai, T., & Hirokawa, S. (2012). Feature words that classify problem sentence in scientific article. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications and Services* (*IIWAS '12*) (pp. 360–367). Bali, Indonesia. doi: 10.1145/2428736.2428803

Sfetsos, P., Stamelos, I., Angelis, L., & Deligiannis, I. (2009). An experimental investigation of personality types impact on pair effectiveness in pair programming. *Empirical Software Engineering*, *14*(2), 187–226.

Shin-ike, K., & Iima, H. (2012). A method for determining classroom seating arrangements by using a genetic algorithm. In *Proceedings of the 20th International Conference on Computers in Education* (*ICCE 2012*). Singapore, Singapore.

Shirouzu, H., Miyake, N., & Masukawa, H. (2002). Cognitively-active externalization for situated reflection. *Cognitive Science*, *26*(4), 469–501.

Suzuki, S. V., & Suzuki, H. (2013). Improving academic essays by writing and reading peer annotations on source documents. In *Proceedings of the 10th International Conference on Computer-supported Collaborative Learning* (*CSCL 2013*) (Vol. 2, pp. 363–364). Madison, WI, USA.

Topping, K., & Ehly, S. (Eds.). (1998). *Peer-assisted learning*. Mahwah, NJ, USA: Lawrence Erlbaum Associates.

Whitley, L. D., Starkweather, T., & Fuquay, D. (1989). Scheduling problems and travelling salesman: The genetic edge recombination operator. In *Proceedings of the 3rd International Conference on Genetic Algorithms* (pp. 133–140). Fairfax, VA, USA.

Williams, L., & Kessler, R. (2002). *Pair programming illuminated*. Boston, MA, USA: Addison-Wesley.