Case Studies of Flipped Classroom Using Adaptive Learning System

Haruki UENO[1]*, Hiroto YAMAKAWA[2] and Hiroshi KOMATSUGAWA[1]
[1] Graduate School of Photonics Science, Chitose Institute of Science and Technology, Japan
[2] Faculty of Science and Technology, Chitose Institute of Science and Technology, Japan
e-mail: ueno@kklab.spub.chitose.ac.jp

In this study, we evaluated the reliability of our model for a flipped classroom through two case studies. In the first case study, we have examined that learners can promote other competencies such as skills for logical thinking or mathematical thinking. Our results show that our model can be applied to introduce not only programming classes but also to logical thinking such as "Algorithm and Programming". In the other case study, we have examined that the model works effectively without depending on teachers' skill, applying the model to a programming class performed by another teacher. The results indicate the reliability of our model without depending on teachers' skill of a series of case studies.
Key words: Flipped Classroom, Adaptive Learning System

1. INTRODUCTION

Learners need to acquire knowledges in academic disciplines that require advanced expertise, and it is necessary to confirm the degree of their knowledge in various situations of lectures. In a previous study (Ueno et al., 2017), we proposed a learning model for a flipped classroom using an adaptive learning system that provided learning quizzes corresponding to learners' understanding degree of knowledge. Further, the learning effectiveness of our model was evaluated through a case study of a C programming class. This class is appropriate for developing learner's competency such as skills for programming. In the present study, our purpose is confirmation of the reliability of our model. We have evaluated the reliability of our model through two case studies. One is the application of our model to not programming classes but classes for logical thinking, using the exercise of Algorithm and Programming. The other is confirmation of the usefulness of our model in case of applying in other teacher's programming classes.

2. MODEL

2.1 Learning Design

We assume that several learning objectives are applied to a class and several lessons in the class are needed to master one objective. For instance, we have fifteen lessons in the C programming class, and three lessons are needed to master the ability for using functions in the C programming language. We set the learning objective of first lesson as understanding knowledge concept, that of second lesson as knowledge utilization and that of third lesson as application of knowledge. We define the period of the lessons for mastering the given learning objective as a "learning unit". Before each lesson, learners are recommended to do preparatory learning using learning materials in our system. At the beginning of each lesson, learners are assigned to take tests for checking the degree of understanding for the preparatory learning.

2.2 Learning Material

The learning materials for preparatory learning are textbooks and quizzes. The textbooks are explanatory notes of a knowledge in three lessons. The quizzes are structured in accordance with the understanding degree of the learning objective. Note that the quiz format comprises a problem, an answer and explanation. We set basic and advanced level to understanding knowledge concepts and application of knowledge and set basic, intermediate and advanced level to utilization of knowledge. The learning quizzes are classified into seven levels, and according to the levels, the objective of understanding knowledge concept is set to level 1-2, that of knowledge utilization is set to level 3-5, and that of application of knowledge is set to level 6-7.

2.3 System

Our implemented learning system mainly provides two functions: (i) an adaptive test function (ATE)

and (ii) an adaptive training function (ATR). The quizzes are shared and provided in both ATR and ATE. ATE is a test function to check learners' understanding degree of knowledge for preparatory learning. The ATE provides tests in which problems are automatically selected from quizzes and changes quizzes adaptively on basis of the learner's response of correct or incorrect answers using a logistic function of Item Response Theory. In the end of the test, the ATE presents a learner with the understanding degree that is defined in seven levels. The ATR is a learning function where learners can solve quizzes of seven levels out of classroom through the knowledge map.

2.4 Case Study

The scheme of its "learning units" is shown in Fig 1, and we suppose that each one of them consists of three lessons. The learning objectives of each lesson corresponds to those of levels 1-2, 3-5, and 6-7, successively. As mentioned above, before each lesson, learners are recommended to do preparatory learning and for this step they could use the function of ATR. At the beginning of each lesson, learners are assigned to take tests for checking the degree of understanding for the preparatory learning. Our model's major characteristic lies in the capability of managing various learning situations through the iterated learning process by using the system. All learners begin their preparatory learning in the first step of the "learning unit" shown in Fig 1, using ATR out of classrooms. In the middle step, some learners also do the preparatory learning for the second lesson, but others may review the first lesson because of their lack of knowledge. In our learning model, this learning phase are allowed by the use of ATR. Learners are adaptively recommended to do their exercises and gain total knowledge in the "learning unit" through our implemented system.

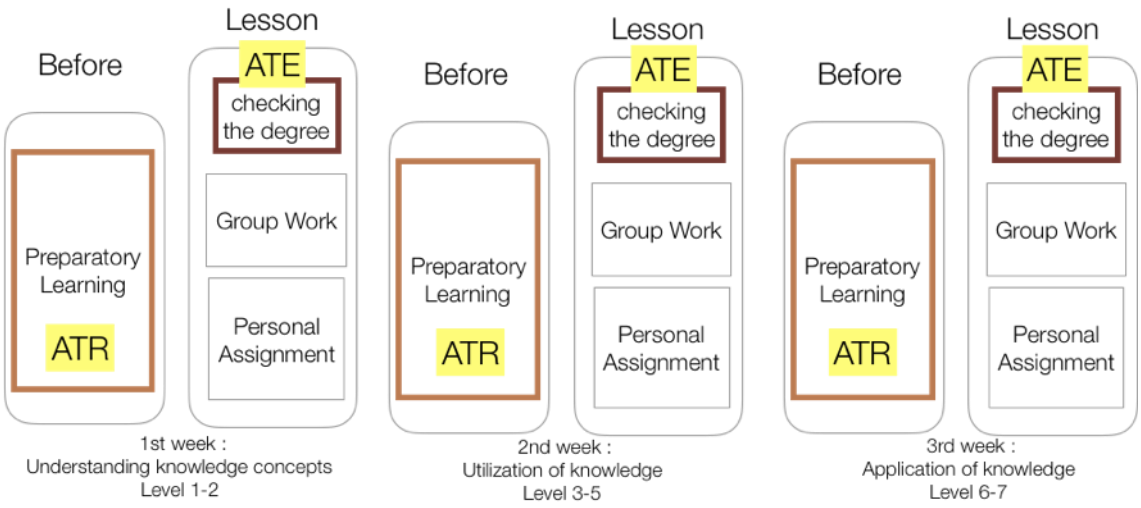

Fig 1. A Model for a Flipped Classroom

3. EVALUATION

3.1 A Class for Logical Thinking

In the first case study, we have examined that learners can promote other competencies such as skills for logical thinking or mathematical thinking. In "Algorithm and Programming" class, learners are recommended to require basic knowledge of algorithm and construct programs using the knowledge logically for 15 lessons. The 15 lessons were classified into 5 "learning units". The objectives of each "learning unit" were "Flowchart" in the first 3 lessons; "Stack and Queue" in the 4th to 6th lessons; "Lists" in the 7th to 9th lessons; "Recursion" in the 10th to 12th lessons; and "Sorts" in the 13th to 15th lessons. We show the results of the learners' numbers at the low, middle and high level through the learning unit of three weeks in Fig 2. The results show that the understanding degree of each learning objective was improved at the end of each "learning unit". The change of learning degree obtained in the present study takes the same tendency as for C programming class performed in the previous study. Our results show that the proposed model can be applied to introduce not only programming classes but also to logical thinking such as "Algorithm and Programming".

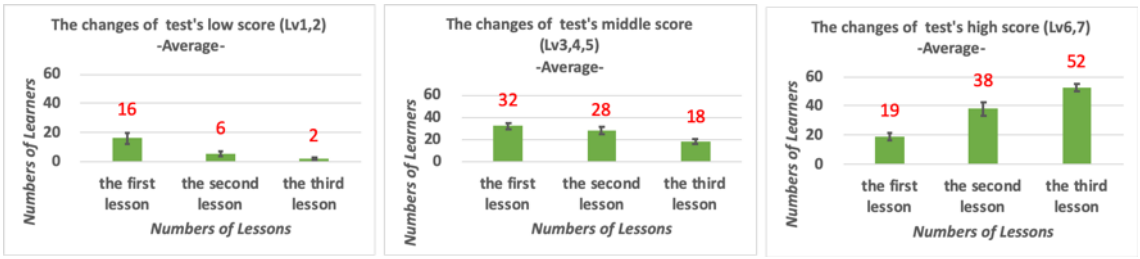

Fig 2. Changes in the tests' score of "Algorithm and Programming" class.

3.2 A Class performed by Another Teacher

In the other case study, we have examined that the model works effectively without depending on teachers' skill, applying the model to Java programming class performed by another teacher. In the class, the 13 lessons were classified into four "learning units". The objectives of each "learning unit" were "Basic Grammar" in the first 3 lessons; "Class" in the 4th to 6th lessons; "Class Design" in the 7th to 9th lessons; the midterm examination in the 10th lesson; and "Library and Exception" in the 11th, to 13th lessons. We show the results of the learners' numbers at the low, middle, and high level through the learning unit of three weeks in Fig 3. The results also show that the understanding degree of each learning objective was improved at the end of each "learning unit". The change of learning degree obtained in the present study also takes the same tendency as for the result of a series of case studies. Our results show that the proposed model can be applied to Java programming class performed by another teacher. Further, we performed midterm and final examinations for Java programming class that was identical to that performed two years ago, which had not been adapted our model. We found that the average score of the midterm examination increased from 66.6 (n = 68) to 76.4 (n = 74) and that the average score of the final examination increased from 80.2 (n = 66) to 86.7 (n = 73). We found a middle correlation (r = .533) between the midterm examination and the ATE results and a low correlation (r = .385) between the final examination and the ATE results. Our results indicate the reliability of the model without depending on teachers' skill of a series of case studies.

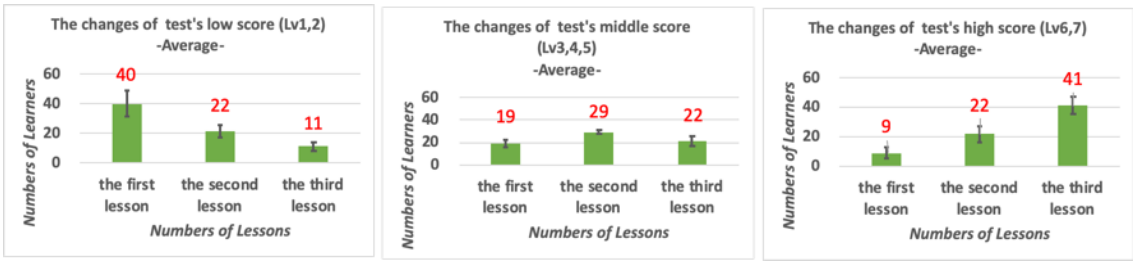

Fig 3. Changes in the tests' score of "Java Programming" class.

3. CONCLUSION

We evaluated the reliability of our model for a flipped classroom through two case studies. In the first case study, our results show that the proposed model can be applied to introduce not only programming classes but also to logical thinking such as "Algorithm and Programming". In the other case study, our results indicate the reliability of the model without depending on teachers' skill of a series of case studies.

Acknowledgements

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (C) Number JP17K00492.

References

[1] Ueno H., Kato T., Yoshida F., Tsukada N., Tateno H., Fukamachi K., Yamakawa H., & Komatsugawa H. (2017). A Model of Flipped Classroom Using an Adaptive Learning System. Work-in-Progress Poster, The 25th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education 1-3.