Rodrigo, M. M. T. et al. (Eds.) (2021). Proceedings of the 29th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Instruction Support System using Impasse Detector and Major Failure Diagnoser for Programming Exercises

Tomoki IKEGAME[a*], Yasuhiro NOGUCHI[b], Satoru KOGURE[b], Koichi YAMASHITA[c], Raiya YAMAMOTO[d], Tatsuhiro KONISHI[b] & Yukihiro ITOH[e]
[a]*Graduate School of General Science and Technology, Shizuoka University, Japan*
[b]*Faculty of Informatics, Shizuoka University, Japan*
[c]*Faculty of Business Administration, Tokoha University, Japan*
[d]*Faculty of Engineering, Sanyo-Onoda City University, Japan*
[e]*Shizuoka University, Japan*
*ikegame.tomoki.17@shizuoka.ac.jp

Abstract: In programming exercise classes, teachers or teaching assistants (hereinafter referred to as "TAs") circulate among the students to answer their questions. However, it is difficult to detect students who reach impasses, to instantly identify the part of the program in which students reach impasses, or to grasp the impasse status of the entire class. To solve these problems, we have developed a system with the ability to display impasses detected from students' exercise information on the source code and to aggregate and display the impasse for the entire class.

Keywords: Automatic impasse detection, programming exercise, monitoring

1. Introduction

In programming exercise classes, a small number of teachers and TAs circulate among the students to answer their questions. However, there are two problems, as shown below.

Problem 1. Difficulties in identifying and resolving the impasse for students
Since the students complete the exercise using their own PCs, it is difficult for the teachers and TAs to detect which student has reached an impasse. In addition, as it is difficult for even a skilled teacher to instantly identify which line in the program is causing the impasse just by looking, it takes time to support students.

Problem 2. Difficulties in grasping the impasse status of the entire class
Since individual teachers and TAs do not have time to share their support results, it is difficult for the entire class to grasp which line of the program caused impasses and what their learning items are.

Several systems collect and analyze students' source code and errors (Fu, Yin, Shimada, & Ogata, 2015; Shamsi & Elnagar, 2012). The same mistake in the source code can have various causes, such as simple typos, forgetting, or not understanding and reaching impasses. For teachers to understand the status of the students in the programming classes, it is important for them to know that the student has reached an impasse. The previous study, 3-Level Impasse Analyzer (Noguchi et al., 2020), provides a method to detect impasses. It analyzes the line of the program that caused an impasse (Lv.1), the line of the correct program corresponding to that line (Lv.2), and its learning items (Lv.3) from students' source code in terms of editing history and frequency of compilation errors. However, it is only capable of detecting text-based information and is not equipped with a framework for displaying the information in an easy-to-understand format.

In this study, we have developed a system that provides a solution to Problems 1 and 2 by displaying this information in a form that is easy for teachers and TAs to understand.

2. Implemented Functions and Mitigated Problems

2.1 Highlighting Students Who Reach Impasse in the Students' List

Those students who reached an impasse are highlighted in the students' list display screen of the system. This function mitigates Problem 1 since it can instantly find students who reach impasses without having to check each student individually.

2.2 GUI Display of Impasse Information for the Individual Students

Place a button to the left of the line number of the line on which students reached impasses in the student program browsing screen of the system. By clicking on this button, you will see what the impasse is and highlight the line of the student program and the correct program corresponding to that impasse, as shown in Figure 1.

Figure 1. Display of Information for the Individual Student (excerpt)

This function mitigates Problem 1 since it can visually obtain the impasse information Lv.1 and Lv.2 of the individual students.

2.3 GUI Display of Impasse Information for the Entire Class

Since the correct program for a task and the learning items corresponding to each of its lines are common to all students, the impasse information Lv.2 and Lv.3 can be aggregated for the entire class. The aggregated impasse information is displayed as shown in Figure 2.

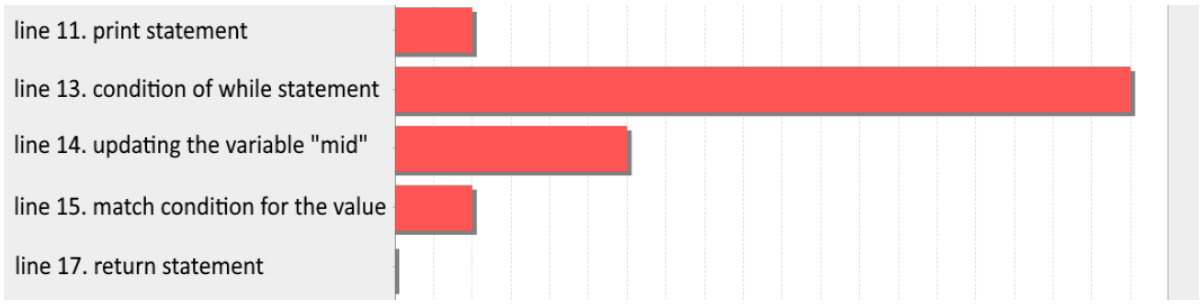

Figure 2. Display of Impasse Information for the Entire Class (excerpt)

Each item is the line that caused an impasse and the corresponding learning item, and the horizontal axis represents the number of people. This function makes it possible to obtain visual information on the parts of the correct program on which the entire class reached impasses and what the learning items are.

3. Evaluation

To evaluate the effectiveness of the system, a preliminary evaluation was conducted on one teacher and three students with TA experience. Using the system containing previously collected information on students' exercises, subjects were asked to perform tasks such as finding students who reached impasses, identifying the location of the impasse, and checking the impasse status for the entire class, and to describe what kind of guidance should be provided. As a result of the evaluation, four subjects wrote that they would do the tasks based on the information presented by the system.

In addition, we gave the subjects questionnaires. The questionnaires were designed using a 4-point evaluation scale, in which 4 is good, and 1 is poor. Questions 1 and 2 correspond to Problem 1, and Question 3 corresponds to Problem 2. The results are shown in Table 1.

Question 1. Is it easy to find the students who reached impasses?
Question 2. Is it easy to identify lines of programs that caused impasses?
Question 3. Is it easy to grasp the impasse status of the entire class?

Table 1. *Questionnaire Results*

	Question 1	Question 2	Question 3
Subject A	4	3	4
Subject B	4	4	3
Subject C	4	4	4
Subject D	4	4	2
Average	4.00	3.75	3.25

From Table 1, we can see that for every problem mitigation, we received a rating of 3 or higher. Therefore, the system developed in this study is effective in mitigating Problems 1 and 2. However, in the open-ended section of the questionnaire, some subjects indicated that they would be able to better understand students' status if they could see information on students' past impasses.

4. Conclusion

In this study, we have developed the system to solve two problems related to students' impasses during programming exercises. The evaluation results showed the effectiveness of the system.

Future works include the display of information on past impasses and support for online classes, which are becoming increasingly popular.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP19K12265 and JP18K11566.

References

Fu, X., Yin, C., Shimada, A. & Ogata, H. (2015). Error log analysis in C programming language courses. *Proceedings of the 23rd International Conference on Computers in Education (ICCE 2015)*. pp. 641-650.

Noguchi, Y., Ayabe, K., Yamashita, K., Kogure, S., Yamamoto, R., Konishi, T. & Itoh, Y. (2020). Experimental design of automated extraction for 3-level tutoring support information in programming exercises. *Proceedings of the 28th International Conference on Computers in Education (ICCE 2020)*. pp. 255-260.

Shamsi, F.A. & Elnagar, A. (2012). An intelligent assessment tool for students' Java submissions in introductory programming courses. *Journal of Intelligent Learning Systems and Application, 4*(1)*,* 59-69.