CodeHelper: A Web-Based Lightweight IDE for E-Mentoring in Online Programming Courses

Xiao Liu
Department of Information Convergence Engineering
Pusan National University
Busan, Republic of Korea
liuxiao@pusan.ac.kr

Gyun Woo
Department of Information Convergence Engineering
Pusan National University
Busan, Republic of Korea
woogyun@pusan.ac.kr

***Abstract*—Many universities choose online courses instead of in-person during the COVID-19 pandemic. One of the limitations of online courses is that it is difficult to perform teacher-student interactions in online teaching environments. Especially in online programming courses, it is hard to find an appropriate approach to remotely guide students when they have trouble conducting program code. This paper presents a lightweight IDE named CodeHelper integrated with an online judge that allows the instructor or the teaching assistant (TA) to help students in a form of online pair programming. Students can share their code with the instructor or the TA in a sandbox programming environment generated by CodeHelper, and the instructor or the TA can guide students by marking the mistakes in the sandbox. CodeHelper allows real-time pair programming for both sides, which can reduce the time and the cost of distance learning. Besides, the source code shared in the sandbox can also be compiled for syntax check, executed with a user-defined input, and submitted to the online judge for automated assessment, which can let students check the correctness of their programs once their troubles have been solved. It is efficient to use CodeHelper for mentoring students in online programming courses.**

Keywords—Online programming courses, e-mentoring, pair programming

I. INTRODUCTION

Students commonly feel difficult to conduct correct program code when they firstly learning a new programming language. Most of the difficulties are solving the syntax errors in their code [1]. Based on the past teaching experience, students usually showed their incorrect code to the instructor or the teaching assistant (TA) in the laboratory session and asked for help. The teachers (instructor or TA) could give appropriate commend of how to solve the errors by reviewing the code. Such an approach worked fine in the in-person programming courses.

Due to the COVID-19 pandemic, lots of universities or institutions hold courses online instead of face-to-face. Such a change made many of the traditional teaching strategies ineffective since online courses can limit the teachers to interact with students. Especially for the programming courses where the teachers are unable to help students in time for solving the mistakes in students' code during the real-time online sessions.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2014-3-00035, Research on High Performance and Scalable Manycore Operating System).

Teachers had to use online meeting tools such as ZOOM to let the students share their screen, and then the teachers remotely controlled the students' computers to help them locate the mistakes since there was no appropriate tool at the first outbreak of the COVID-19. There are three major issues with using such an approach. First, since remote sharing demonstrates one's screen to all participants, students may feel embarrassed about sharing their incorrect code. Second, since remote sharing takes lots of network bandwidth, it is unable to guarantee that both teachers and students have the same qualified network connection in an online session. The remote sharing would not be performed well if one of the participant's networks is unstable. Third, the teachers can only inform the mistakes by whether audio or chat, instead of intuitively marking in the code.

To improve the effectiveness of programming education in online courses, CodeHelper is developed as an e-mentoring tool to assist the teachers to guide students in a form of pair programming. CodeHelper is developed using web technologies and supports text-based program code sharing for both teachers and students in real-time. CodeHelper also supports compilation for syntax check and program execution with user-defined input data. In addition, CodeHelper is integrated with an online programming judge system to implement program assessment. Such an implementation allows students to test their code immediately once the teacher helped them to solve their mistakes in CodeHelper.

Since CodeHelper allows a one-to-one code sharing between one student and one teacher, it can protect the student's privacy. Besides, the bandwidth of using CodeHelper is much less than remotely controlling one's computer since it only requires loading a web page. Besides, the teachers can mark the parts of incorrect code with suggestions as to more precise feedback.

CodeHelper is applied in a real-world online C++ programming course held in the Pusan National University during the first semester of the 2021 academic year. There are two scenarios of use cases collected from using CodeHelper in this course. The first scenario consists of the assistance that the teachers in helping the students who have trouble conducting code in laboratory sessions. The second scenario consists of the cases of using CodeHelper for pair programming between the teachers and students to conduct programs together. The two scenarios of use cases can help us to verify the effectiveness of CodeHelper and address its barriers in online programming courses.

The rest of this paper is organized as follows. Section 2 introduces the related studies based on pair programming and points their limitations. Section 3 describes the development of CodeHelper and the use cases. Section 4 discusses the benefits and the challenges of using CodeHelper. Section 5 concludes.

II. Related Work

A. *Pair Programming in Education*

Pair programming is a program development strategy that two developers work together for the same project [2]. One of the developers performs as a driver to conduct program code. The other acts as a navigator to review the conducted code in time. It is efficient that the mistakes conducted by the driver can be pointed by the navigator immediately.

Recently, there are studies [3-5] show that the effectiveness of pair programming can encourage students to learn programming skills in programming courses. Especially when the mentors with strong knowledge of programming can help the paired students efficiently [6]. In addition, since the industry adopts pair programming strategy in lots of real works, letting students grasp the skills of pair programming such as professional code review and collaborative development can benefit them to find jobs in the future.

B. *E-Mentoring Systems*

To apply pair programming in university programming courses efficiently, there are several studies focus on computer-based systems to provide a collaborative environment for programming practices [7]. The systems are developed whether as a plugin tool in an existing integrated development environment (IDE) or as a standalone platform. In a typical use case of such systems, a learner can use a computer asking for help tough a collaborative tool, and the mentor can assist the learner on a computer as well. These studies show that such an e-mentoring approach could improve the efficiency of help students learning programming skills in distance learning.

Codechella [8] is a real-time tutoring platform based on collaborative pair programming. It takes advantage of a visualization tool to generate an execution graph of students' programs, and if there are questions related to the program, the teachers can base on the graph to answer the questions via chat. However, a program should be tested based on input and output data for correctness check, and Codechella focuses on the logic implementation only.

SCEPPSys [9] is developed as a plugin of Eclipse IDE to provide a collaborative environment for distributed pair programming. Students can use SCEPPSys to share their code editor for collaborative development or reviewing others' code. However, the role of the instructor is an observer of the whole development progress, who is not involved in the pair programming. Students have to solve the problems on their own and the instructor cannot help them via SCEPPSys.

EdCode [10] is a platform that allows students to share their code with questions to the instructors and asking for help. The instructors can review the code and send answers back to students. EdCode provides both real-time and asynchronous pair programming in case of students may ask questions after the class session. However, EdCode only shares the text-based content (program code, questions, and answers) and either students or the instructors have to test the code by pasting it to their own IDE, which is inconvenient if one of the participants just happen to use a device that has no proper IDE installed.

CoVSCode [11] is a lightweight plugin of Visual Studio Code IDE developed for collaborative programming. It allows paired students to work together on the same project whether locally or remotely. The collaboration can be performed in real-time, and the notification is pushed to others once one of the participants modified the code. However, similar to the previously mentioned systems, CoVSCode also cannot provide the correctness check for the code. Besides, this study did not mention how the teachers were involved in the pair programming.

III. Methodology

A. *Development of CodeHelper*

CodeHelper is developed using web technologies and both the teachers and students can share program code through the web browser. In an online programming course, if there is a student who needs help from a teacher for solving his/her code errors, the teacher can create a session of CodeHelper and the student can participate in the session to share code. Fig. 1 shows the sequence diagram of how the CodeHelper session is created and works.

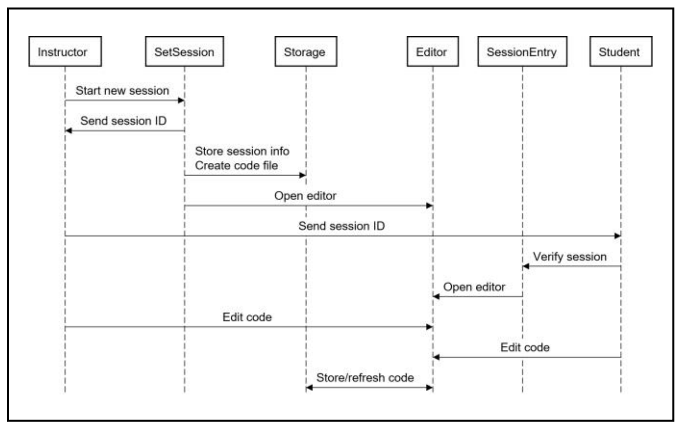

Fig. 1. Sequence diagram of creating a new session in CodeHelper

At first, the teacher creates a new session through the SetSession function. This function will store the session information including the teacher ID, the student ID, the session ID, the location of storing the shared code, the time of the session is created, and the status (open or closed) of the session.

After the storage, the SetSession function will open a code editor for the teacher with the session ID. Then the teacher can send the session ID to the student who asked for help.

Once the student received the session ID, he/she can enter the session by verifying the student and session ID through the SessionEntry function. After the verification, the SessionEntry function will open the code editor in the same session with the teacher.

Once both participants entered the session of CodeHelper, they can use the session in two ways. The first is the student can paste the code into the editor and the teacher can mark the incorrect part of the code and give suggestions of how to fix it. The second is the teacher can share a sample code through the editor to the student to explain the asked question.

To implement efficient code sharing in CodeHelper, a messenger module is developed for real-time code update and notification of code marking. Fig. 2 shows its sequence diagram.

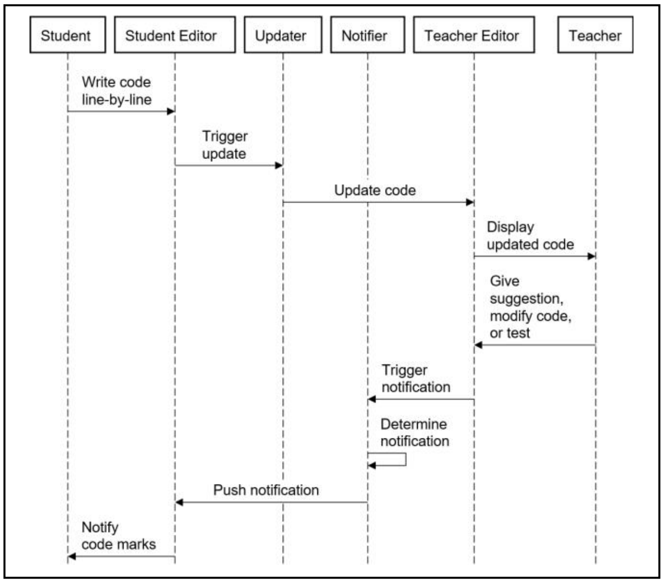

Fig. 2. Sequence diagram of automatical code update and notification

The code update is automatically executed after every line of code typed by the student. The Student Editor will call the Updater function to update the code in the Teacher Editor. Besides, the teacher can perform the actions such as marking the incorrect part of the code, giving suggestions, or testing the program for compilation and execution with input data. For each of the actions, the Notifier function will determine the type of the actions and push the corresponding notification as an alert box in the Student Editor. The student can check the changes such as marked code, compiled result, and the output of execution by confirming the notification.

The real-time code update and the notification of code marking and testing are available for both participants, which means students can send questions and test programs, meanwhile, the teachers can receive the result in real-time and through notification as well.

As demonstrated in the web page of CodeHelper in Fig. 3, the teachers and students can execute actions such as saving and sending code (1), refreshing code editor (2), asking and answering questions (3), compiling the program (4) to check the syntax (5), making sample input data (6) to execute the program (7), check the output of the execution (8), submitting the code to the online judge (9), downloading source code file (10) and closing the CodeHelper session (11).

Fig. 4 shows the sequence diagram of how CodeHelper handles the actions with different modules. Once the Save/send button was clicked, the Action handler firstly stores the code in the editor into the system database. In the second, the handler calls the messenger module for broadcasting the changes.

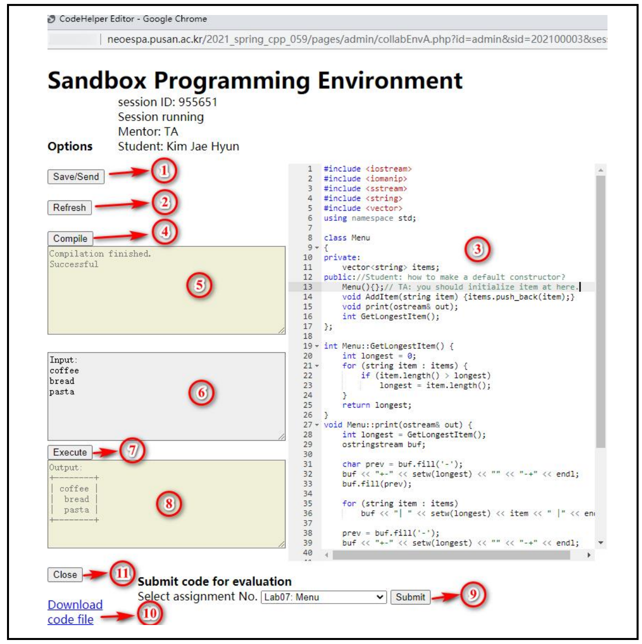

Fig. 3. The web page for using CodeHelper

Once the Refresh button is clicked, the handler firstly reads the stored code from the database in case of any changes of code from the other participant. In the second, the handler sends back the code to the editor.

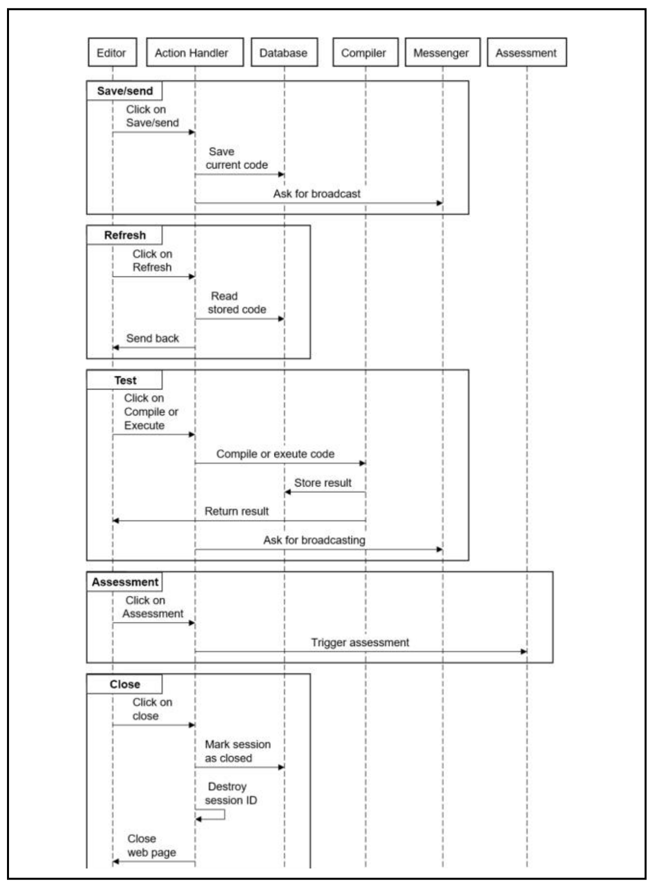

Fig. 4. Sequence diagram of action handling in CodeHelper

For code testing, the code should be compiled and executed. Once the Compile button is clicked, the code in the editor will be sent to the compiler in the back-end and the result of the compilation will be sent back to the editor. Once the Execution button is clicked, the code with the input data typed by the participants will be executed and the output will be sent back to the editor after execution. For both compilation and execution, the result will be informed to the other participant through notification.

Once the Submit button for assessment is clicked, the code in the editor will be sent to the online judge for the correctness check. Since this action invokes the judging system as an external module, the feedback of the correctness check will be displayed on the newly opened assessment web page.

Once the Close button is clicked, the handler firstly marks the current session as closed in the database and then destroys the session ID in the web cookie. In the second, the handler closes the editor and exits CodeHelper.

B. *Use Cases*

CodeHelper is applied in a C++ programming course held in PNU during the first semester of the 2021 academic year. This course is held online due to the COVID-19 pandemic. Till this study is presented, this course is still ongoing, and CodeHelper has only be used with a few students. The use cases are classified as two sets by scenarios: mentoring and pair programming.

In the first scenario, the TA used CodeHelper to guide the students who had errors in their code and seeking for help in the online session of this course. There are ten students who asked for help through CodeHelper in two online sessions. Once the students shared their code with questions through CodeHelper, the TA marked the incorrect parts of the code with suggestions and send the feedback to the students in time.

Lots of time is reduced by using CodeHelper comparing with the situations where CodeHelper has not been invoked and the teachers had to copy the students' code from chatting tools, paste it to teachers' own editor, review the code, and then give the feedback to students through chatting tool again.

After the midterm exam, the teacher assisted students to solve the exam problems for knowledge review. There are five problems in the exam and each of them requires students to conduct an independent program to solve. The second scenario is that several students were recruited as volunteers and used CodeHelper to conduct program code as pair programming.

For each exam problem, the instructor worked with one student in a CodeHelper session. The convenient code sharing occurred for every line of code typed by the student in real-time. The instructor observed the code conduction of the student meanwhile assisted the student if he/she had troubles and gave appropriate comments immediately if the student made mistakes.

For both scenarios, the teachers and students used CodeHelper for code sharing, compilation, execution, and eventually submitting to the online judge for assessment. Since all the development processes were conducted in CodeHelper, they reduced lots of time for switching tools.

IV. Discussion

A. *Benefits of CodeHelper in Online Programming Courses*

Table I compares CodeHelper with other similar systems introduced in Section 2. The system features compared in Table I present the effectiveness of each system for e-mentoring in online programming courses.

The teachers are involved in the use of CodeHelper which can help students locate and solve their problems quickly. Such an e-mentoring approach is more efficient than letting students learn by themselves.

The web-based design of CodeHelper does not require any installation of extra software or plugin. It simplifies the use of CodeHelper so that the participation of the mentoring has no restriction of devices and locations for both teacher and students.

In addition, owing to the connection to the online judge, the program code in CodeHelper can be automatically tested for correctness. It reduces much time for the participants to check the code in local computers.

TABLE I. Comparison of Functionalities among Similar Systems

System name	**System features**		
	Teacher involved	***Web-based***	***Correctness check***
Codechella [8]	Yes	Yes	No
SCEPPSys [9]	No	No	No
EdCode [10]	Yes	Yes	No
CoVSCode [11]	No	No	No
CodeHelper	Yes	Yes	Yes

B. *Challenges of using CodeHelper*

Currently, the teacher is able to assist only one student through CodeHelper at one time. Other students have to wait till the previous session ends then participate in a new session one by one. Such a disadvantage may reduce the interest of students to use CodeHelper.

CodeHelper relies on the teacher's responses to every request from students. In the aforementioned use cases, there was only one TA who participated as a teacher for the mentoring. Such a lack of TA will be too busy to handle all requests if there are many students ask for help at the same time.

V. Conclusion

This paper adopts the concept of pair programming in education and conducted a web-based e-mentoring IDE called CodeHelper. The two scenarios of using CodeHelper proved that it is effective to help students to solve their problems and guide them to conduct correct code in an online programming course.

As the online courses are ongoing and the world is still in the middle of the COVID-19 pandemic, it is necessary to improve CodeHelper to support one-to-many assistance for helping more students at one time. Besides, it is necessary to consider whether increase the number of TAs or grant the students who have better programming skills as teachers to make sure every help can be answered in the course session.

In the future, a questionnaire is going to be offered regarding the user experience of CodeHelper to analyze the perceptions of teachers and students on such an e-mentoring approach. The constructive opinions from teachers and students can be helpful for improving the effectiveness and the practicality of CodeHelper.

REFERENCES

[1] Y. Bosse and M. A. Gerosa, “Why is programming so difficult to learn? Patterns of Difficulties Related to Programming Learning Mid-Stage,” ACM SIGSOFT Software Engineering Notes, vol. 41(6), pp. 1-6, 2016.

[2] L. Williams and R. R. Kessler, Pair programming illuminated, Addison-Wesley Professional, 2003.

[3] F. J. Rodríguez, K. M. Price, and K. E. Boyer, “Exploring the pair programming process: Characteristics of effective collaboration,” In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, March 2017, pp. 507-512.

[4] K. M. Ying, L. G. Pezzullo, M. Ahmed, K. Crompton, J. Blanchard, and K. E. Boyer, “In their own words: Gender differences in student perceptions of pair programming,” In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, February 2019, pp. 1053-1059.

[5] Ö. Demir and S. S. Seferoglu, “A Comparison of solo and pair programming in terms of flow experience, coding quality, and coding achievement,” *Journal of Educational Computing Research*, vol. *58*(8), pp. 1448-1466, 2021.

[6] N. Aottiwerch and U. Kokaew, “The analysis of matching learners in pair programming using K-means,” In *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, April 2018, pp. 362-366.

[7] L. Silva, A. J. Mendes, and A. Gomes, “Computer-supported collaborative learning in programming education: A systematic literature review,” In *2020 IEEE Global Engineering Education Conference (EDUCON)*, April 2020, pp. 1086-1095.

[8] P. J. Guo, J. White, and R. Zanelatto, 2015, “Codechella: Multi-user program visualizations for real-time tutoring and collaborative learning,” In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, October 2015, pp. 79-87.

[9] S. Xinogalos, M. Satratzemi, A. Chatzigeorgiou, D. Tsompanoudi, “Student perceptions on the benefits and shortcomings of distributed pair programming assignments,” In *2017 IEEE Global Engineering Education Conference (EDUCON)*, April 2017, pp. 1513-1521.

[10] Y. Chen, J. Herskovitz, G. Matute, A. Wang, S. W. Lee, W. S. Lasecki, and S. Oney, “EdCode: Towards Personalized Support at Scale for Remote Assistance in CS Education,” In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, August 2020, pp. 1-5.

[11] H. Fan, K. Li, X. Li, T. Song, W. Zhang, Y. Shi, and B. Du, “CoVSCode: A Novel Real-Time Collaborative Programming Environment for Lightweight IDE,” Applied Sciences, Vol. *9*(21), Art. no. 4642.