

An Implementation of Teacher Service Functions in Desktop-Version Java Programming Learning Assistant System

Su Sandy WINT^{*1}, Nobuo FUNABIKI^{*2}, Khin Khin ZAW^{*3}, Nobuya ISHIHARA^{*4}

Graduate School of Natural Science and Technology
Department of Electrical and Communication Engineering
Okayama University, Okayama, Japan

Email: susandywint@s.okayama-u.ac.jp, funabiki@okayama-u.ac.jp

Abstract: To advance the Java Programming study, we have developed *Java Programming Learning Assistant System (JPLAS)* as a web-based application system. Unfortunately, JPLAS is not suitable for developing countries where the Internet access is often poor and frequent power shortages occur at schools and homes. In this paper, we implement *teacher service functions* in the *desktop-version JPLAS (D-JPLAS)* to allow the use of JPLAS without the Internet access. For *student service functions*, we adopt the existing implementation. To ease the usage by people who do not have much knowledge to manage the web application system, *D-JPLAS* does not require the server and database by managing all the data in the file system. The simple evaluation of the implementation confirms the usefulness of D-JPLAS in Java programming educations with JPLAS.

Keywords: JPLAS, desktop, Java, teacher service functions, Swing GUI

1. Introduction

Nowadays, the object-oriented programming language, *Java*, has been widely used in real world applications such as Web sites, enterprise applications, IoT (Internet of Thing), and cloud systems. To enhance the Java programming study, we have developed *JPLAS* [1]- [4] and been using it in our university. However, as a Web application system, the use of JPLAS may have some difficulties in these cases: the stable Internet access is not provided due to cost and frequent power shortage reasons, and skilled persons are not be available to manage the Internet, the Web server, and the database system.

To avoid those difficulties in the online JPLAS, in this paper, we implement the *desktop-version JPLAS (D-JPLAS)*, which can be the efficient solution for developing countries where the Internet access is often poor at schools and homes. Besides, D-JPLAS does not use the server or database so that it can be easily used for those who have less knowledge.

D-JPLAS has *teacher service functions* and *student service functions*. With *teacher service functions*, a teacher can generate new problems and assignments for students, collect answers from the students, and save and analyze them, on offline using the desktop PC. With *student service functions*, a student can answer the problems and assignments, on offline. The files for assignments and answers should be exchanged between the teacher and the students by using flash devices, file servers, or emails. In this paper, we implement teacher service functions, whereas we adopt the existing implementation for the student service functions [5][6].

2. Implementation Teacher Service Functions

In this section, we present the implementation of the teacher service functions in D-JPLAS. Java with Swing GUI [7] are used in the implementation.

2.1 Eight Problem Types in JPLAS

To cover various students at different levels, currently JPLAS provides the eight problem types: 1) element fill-in-blank problem (EF) requests to fill in the missing elements in the given problem code, 2) value trace problem (VT) requests to trace values of important variables, 3) code completion problem (CC) requests to complete the code by filling missing elements, 4) code amendment problem (CA) requests to fix the wrong source code according to test code, 5) code extension problem (CE) requests to modify the source code by extending the given source code, 6) statement fill-in-blank problem (SF) requests to fill the statement given in the problem code, 7) code writing (CW) requests to write source code mentioned in test code, 8) code clone detection problem (CD) requests to optimize the copied code in the source code.

2.2 File System Directory Layout

To manage various files for generated problems, assignments, answers from students, and their analysis results, the following file system directly layout is adopted in D-JPLAS.

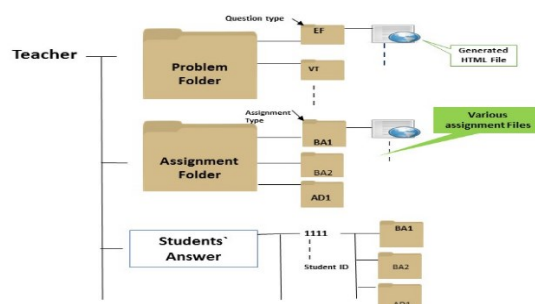


Figure 1 File system directory layout.

2.3 Problem Generation

To generate a new problem, the teacher needs the following procedure: 1) select the problem type, 2) set the problem name, difficulty level, statement, and comment, 3) prepare the source code, 4) run the problem generation program if available, and 5) store the generated files to the file system. Figure 2 shows the interface and the generated HTML interface for EF.

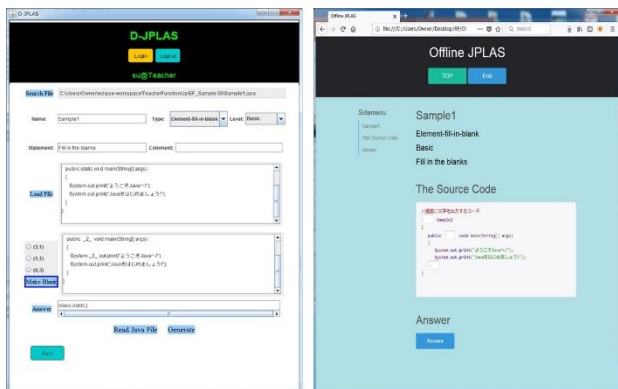


Figure 2 Problem generation interface and example generated EF interface.

2.4 Assignment Generation

A new assignment can be generated by: 1) set the problem name, difficulty level, statement, and comment, 2) select the problems, 3) store the generated files to the file system, 4) compress the folder containing the files into the zip file. Figure 3 shows the interface for compressing the folder and the file system directory.

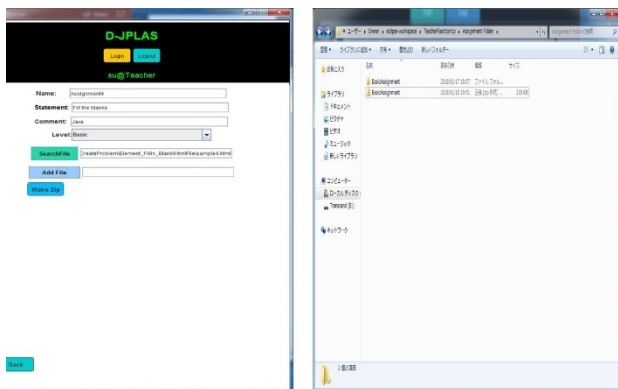


Figure 3 Folder compression and file system directory.

2.5 Answer File Management and Analysis

The submitted answer files from the students are stored into the corresponding folder. The answer files are extracted into CSV files and will analyze them to know how many problems as well as assignments that the students can solve.

3. Evaluation

To evaluate the usefulness of D-JPLAS in the Java programming educations with JPLAS, we applied it to three teachers in our department or in other schools. They commented that D-JPLAS is very useful because it can be used without the Internet access. Moreover, the teachers remarked that it is easy to generate problem files as well as assignment files with D-JPLAS. And there is no difficulty in handling file management system. Therefore, D-JPLAS makes teachers convenient, easy and less time not only in generating files but also in creating files for the teacher service functions.

From now on, we will ask teachers in several universities in Japan and Myanmar to use D-JPLAS for their Java Programming courses. Then, we will improve functions in D-JPLAS to meet their requests.

4. Conclusion

This paper presented the implementation of teacher service functions in **D-JPLAS**. In future works, we will implement the answer file management and analysis function for teacher service functions in **D-JPLAS**. Then, it will be applied to Java programming courses in Japan and Myanmar.

References

- (1) N. Funabiki, Y. Matsushima, T. Nakanishi, K. Watanabe, and N. Amano, "A Java programming learning assistant system using test-driven development method," IAENG Int. J. Computer Science, vol. 40, no. 1, pp. 38-46, Feb. 2013.
- (2) K. K. Zaw, N. Funabiki, and W.-C. Kao, "A proposal of value trace problem for algorithm code reading in Java programming learning assistant system," Inf. Eng. Express, pp., Sep. 2015.
- (3) N. Ishihara, N. Funabiki, and W.-C. Kao, "A proposal of statement fill-in-blank problem using program dependence graph in Java programming learning assistant system," Inform. Eng. Express, vol. 1, no. 3, pp. 19-28, Sep. 2015.
- (4) N. Funabiki, Tana, K. K. Zaw, N. Ishihara, and W.-C. Kao, "A graph-based blank element selection algorithm for fill-in-blank problems in Java programming learning assistant system," IAENG Int. J. Computer Science, vol. 44, no. 2, pp. 247-260, May 2017.
- (5) N. Funabiki, H. Masaoka, N. Ishihara, I-W. Lai, and W.-C. Kao, "Offline answering function for fill-in-blank problems in Java programming learning assistant system," in Proc. ICCE-TW, pp. 324-325, May 2016.
- (6) N. Funabiki, Y.-X. Wang, N. Ishihara, and W.-C. Kao, "An offline answering function for code writing problem in Java programming learning assistant system," in Proc. ICCE-TW, pp. 245-246, June 2017.
- (7) Swing GUI, <https://docs.oracle.com/javase/tutorial/uiswing/index.html>.