

Development and Evaluation of a Presentation Software for Web Programming Language Teaching

Hidekazu KAMINISHI^a & Masao MUROTA^a

^a*Dept. of Human System Science, Graduate school of Decision Science and Technology,
Tokyo Institute of Technology, Japan
{hideka, murota}@mr.hum.titech.ac.jp*

Abstract: In this paper, we developed a presentation software for being used in web programming language course. It has functions to display sample source code of web programming language, edit it, and display its execution result in one slide. The reviews by nine teachers proved its efficiency in programming language teaching. By evaluations, good results were obtained for the functions that enable programming classes to be more interactive between the teacher and his/her students.

Keywords: Presentation Software, Computer Education, Web Programming Language, Development of Software

1. Introduction

1.1 Backgrounds

Presentation software and devices are commonly used. The most popular presentation tool is Microsoft PowerPoint [7]. However, various presentation tools that solve some weaknesses or problems of PowerPoint have been developed and used recently; for example, Tablet PC-based presentation system [1], presentation system with Mind Maps which introduces a spatial organization [5], or a system with index cards that are printed with slide content [8].

In a class of programming language, teachers often show a sample code with presentation tools, and then students work coding. To make students imitate sample code is a popular teaching method. Students do not often understand how the code works or imagine the execution result of the code, if the teacher shows only the code. Because of that, teachers should display the result of the sample code at the same time. However, this method has some problems;

Negative effect of split-attention by changing applications

Generally, teachers change two applications (presentation software and the programming environment tool) to display sample code (with explanation of code) and the results. This method separates the explanatory information from the results of executing codes, which is considered to cause the negative effect of split-attention [2] in students. Owing to this, the teaching method, frequently changing applications is not adequate.

Teacher's effort of preparing lectures

An experiment conducted with dual screen operation reduced the frequency of application change than those with single screen [3]. They tried to reduce the negative effect of the split-attention by using dual screen. They used tools of the popup object that corresponded with explanation and the executing programming steps. However, in that study, teachers have to set the popup object. This may be hard for teachers.

Lack of interactivity in the lecture with screenshots

The other method to reduce the frequency of change on a screen is to display both the sample code and the result in one screen at the same time. However, these screenshot cannot

be edited during the lecture. That is to say, modifying the sample code during a presentation is not easy. This is considered to be a barrier for interaction between the teacher and the students in a class.

1.2 Purpose of This Study

In this study, we developed a presentation software with the following features:

Reduction of the negative effect of split-attention

We aimed to reduce the effect of split-attention by displaying both a source code and its results of execution without changing application.

Support the interactivity between the teacher and his/her students

Sample code can be edited even when teachers are operating presentation. This function enables teachers to make lectures more interactive.

Reduction of the effort of preparing lectures

When teachers teach a programming course with presentation software, they have to edit both presentation slides and sample code, to be displayed in different application. This software enables teacher to edit them in one file at the same time.

Reduction of the operation between applications

With this presentation software, teachers can display both the sample code itself and the result of executing it. This function enables teachers to display them without worrying to change application for displaying both.

2. Feature and Function of the Software

2.1 Feature of this software

This presentation tool has an “environment” of editing or executing sample codes in a slide. The result of executing a sample code is displayed in the same screen.

When we make presentation slides, we edit text file with the Wiki-like notation. This notation was previously proposed [6], but now it is expanded with new functions.

An example of the layout of this tool is shown in Figure 1. “Code box” is an area to display or edit sample codes. “Execution box” is an area to display the result of executing the sample code.

Users of this software can display in execution box the result of the sample in code box. It is possible to edit this sample code without another application while displaying slides. It is possible to reflect the edited code in execution box with one click.

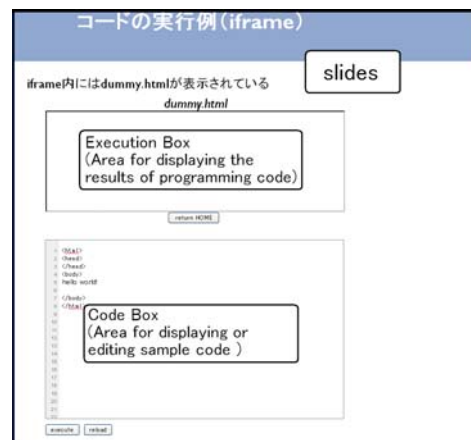


Figure 1: Layout of a slide and components

2.2 Components and functions of this software

Code box

We can display editable text areas using what is called “code box”.

We choose for each code box the what specific programming language to execute when we make presentation file. HTML/CSS/JavaScript are available.

Execution box

“Execution box” is the component which displays the result of executing the codes.

Execution box is made of the “iframe”, one of the HTML element. That is to say, HTML for showing results (HSR) is displayed there.

Executable code is executed and displayed in HSR.

Teachers who use this software have to prepare the HSR files before making slides. In Figure.1, “dummy.html” is displayed in execution box. It is an empty HTML file.

Execution button

Execution button (with the label “execute”) is displayed below the code box. When execution button is pressed, an executable code in the code box is executed, and the result is displayed in execution box.

These “execute code function” enable teachers to edit sample codes, and execute them in the same slide.

2.3 Consideration of the functions

Basically, the codes to be executed in this software are not limited, even if they include an unsafe code what is called infinite loop. This is because it is for teaching purpose, and due to the fact that since teachers have enough programming language skills. Besides, it is also for the purpose of teaching the unsafe code to students.

The result of execution “overwrites” results, not “appends” results. This is because in appended mode, students may be confused what code causes the given results. Therefore, we choose the overwritten mode in this software.

3. Software Reviews by Teachers

3.1 Procedure of Experiment

To prove the efficiency of this software, we conducted experiments for review. Subjects were nine people who have experience teaching computer classes in universities. The experiment includes micro teaching session and usability test, with pre/post questionnaire. These procedures are shown in Figure 2.

Before starting the experiments, an experimenter explained subjects how to use this tool, and let them use it for several minutes. After that, subjects were requested to fill in a pre-questionnaire.

In micro teaching session, the experimenter taught a section of a programming course to a subject. The content of the class was “Introduction to CSS”, which includes the grammar of CSS with displaying examples. The experimenter did not display only one sample, but also showed another related sample (for example, examiner replaced class selector of CSS with other one in a sample code) by the function of editing codes.

The course target was university students whose major are not computer science. We chose this content because sample source codes are collaborated with HTML and CSS. Subjects were provided with a paper handout of slides and a pen. They can use a computer with this software that has the same slides as the teacher does. They can watch slides on the computer and try to use this software, for example, type source code and display its execution results.

In usability test, subjects were requested to repeat teacher’s operations in micro teaching session with this software.

Later, subjects were requested to fill in a post-questionnaire.

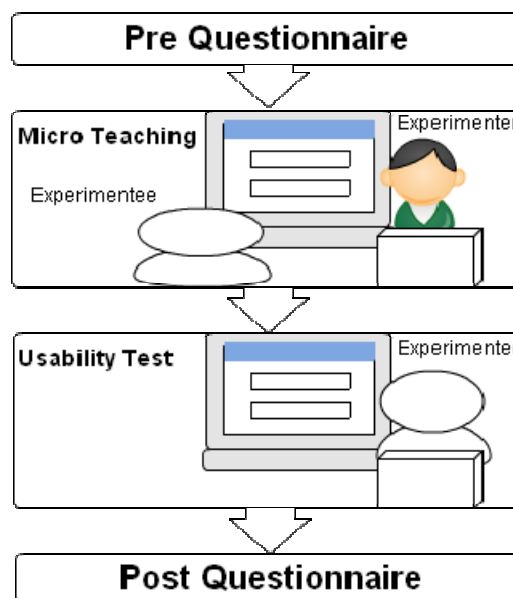


Figure 2: The procedure of an Experiment

3.2 Results of questionnaire

In the pre-questionnaire, all subjects used presentation tools to show slides and application software to give their lectures. They had to switch the presentation tool and application software to display. The score of the question on whether or not the changing application causes interruption for lecture was 3.67 (SD: 1.05), which has a wide deviation.

Results of post-questionnaire (scored by 5 points: 5 are very good) are shown in Figure 3.

Questions 1-4, related to visual issues, scored low, because of the small size of characters.

Questions 6-8 and 10, those that are related to the main purposes of this paper, scored over 4 points in average. They are appreciable results to evaluate. However, Question 9 did not score high (3.89 in average) compared with other questions. This result is considered to be due to the lack of displaying area. In the case that three or more boxes are included in one slide, to avoid the use of small characters, we deal with the shortage of displaying area by scrolling up and down. Consequently, there is no way but to use scrolling slides in order to deal with this problem.

Free comments on post-questionnaire are shown bellow.

- Display slide without scrolling (5 persons)
- Show error message (3 persons)
- Support other language (2 persons)

3.3 Considerations

Displaying of Slides

In programming lecture, characters in screen tend to be small. This experiment is not an exception, the results related to visual issues in questionnaire, did not score well.

When implementing new functions in screen (execution box in this case), the area to display conventional content (sample code in this case) become inevitably small. To solve this problem, we should use larger, more high-resolution screen or use two or more screens.

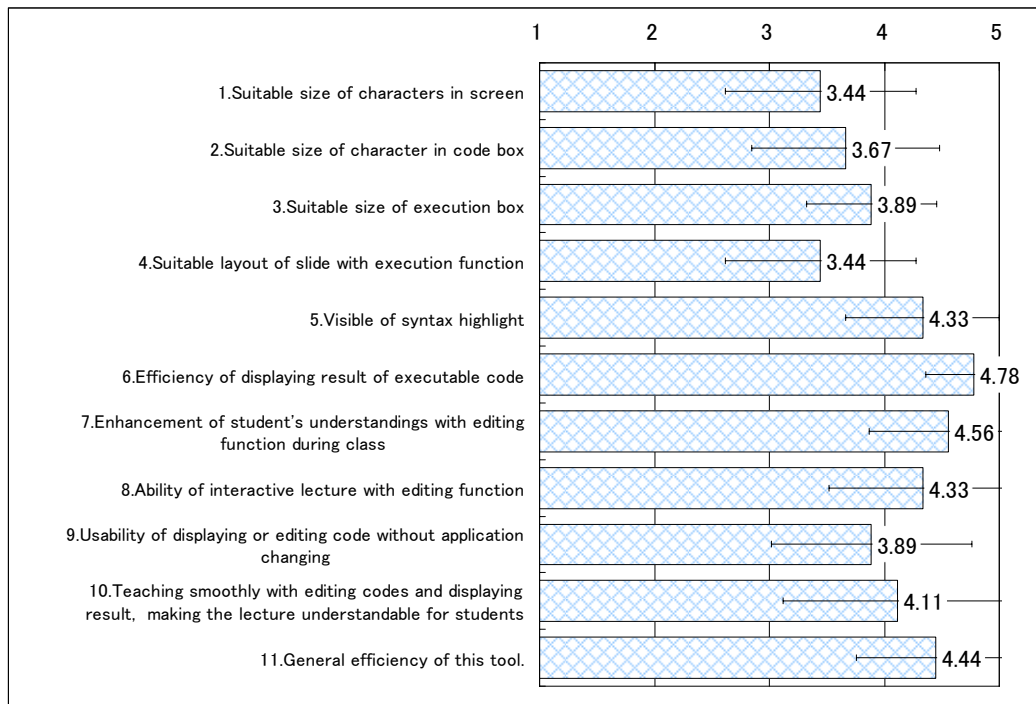


Figure 3 : The result of a post-questionnaire by 5 points (5: very good)

But this is a solution based on hardware. It is difficult to solve it via software.

Functions of Execution and Editing

Because question 10 scored high, teachers evaluated that this software is good for reduction of split-attention. And question 8, related to the teacher-student interactivity, also scored high, so this software is evaluated as an interactive tool between the teacher and his/her students.

Free comments

A problem of scrolling is mentioned before.

Function of displaying error message may be solved by a debug tool. Firebug, that is an add-on of Firefox, is a tool of notifying error of HTML/CSS/JavaScript, et cetera [4]. But at present, it appears unclear messages with this software, so some modifications are required.

To support other languages, we may use “ideone API”. It is an online compiler and debugging tool which allows to compile and run code online in more than 40 programming languages[9]. These results are able to be displayed in web page.

4. Conclusion and future study

4.1 Conclusion

We developed a presentation software for being used in web programming language courses. It contains the function to display sample codes of web programming language, editing these codes, and display the executed results of these codes.

According to teachers’ review, this tool is efficient for interactive lectures between teachers and students. However, scrolling slides derived from the shortage of displaying area did not evaluate good.

4.2 Future study

Evaluate the impact of students’ split-attention

Though this software has been evaluated by teachers, evaluation from the point of view of students is not enough. It remains unclear whether or not this software reduce the students’ split-attention. We plan to assess those on an experimental lecture.

Implementation of the additional function

This web application has also the potential to work on the net. With web server, students will be able to post the code to be checked by teacher.

This software is based on the presentation software of the same authors [6]. That is why it is easy to adopt multi-screen environment.

References

- [1] Anderson, R., Anderson, R., Simon B., Wolfman, S. A., VanDeGrift, T. & Yasuhara, K. (2004). Experiences with a tablet PC based lecture presentation system in computer science courses. *Proc. of CSE '04* (pp.56-60): ACM Press.
- [2] Ayres, P., & Sweller, J. (2005). The Split-Attention Principle in Multimedia Learning. *The Cambridge Handbook of Multimedia Learning*, 135-146, Cambridge University Press.
- [3] Chang, Y., Chang, T., Hsu, J. & Yu, P. (2009). The Impact of Split-Attention Effect on Dual-Screen Learning Environment for Programming Language Instruction. *Proc. of ED-MEDIA 2009* (pp.2110-2115). Honolulu, Hawaii: AACE.
- [4] Hewitt, J. Firebug. <http://getfirebug.com/>. (Referred on May 2, 2010)
- [5] Holman, D., Stojadinović, P., Karrer, T. & Borchers, J., (2008). Fly: an organic presentation tool. In *extended abstract of CHI '06: Conference on Human Factors in Computing Systems* (pp.863-868). Montreal, Quebec: ACM Press.
- [6] Kaminishi, H. & Murota, M. (2009). Development of Multi-Screen Presentation Software. *Proc. of ED-MEDIA 2009* (pp.3934-3939). Honolulu, Hawaii: AACE.
- [7] Levasseur, D. G., & Sawyer, K. J. (2006). Pedagogy Meets PowerPoint: A Research Review of the Effects of Computer-Generated Slides in the Classroom. *The Review of Communication* 6, 1-2, (pp.101-103).
- [8] Nelson, L., Ichimura, S., Pedersen, E. R. & Adams, L. (1999). Pelette : A Paper Interface for Giving Presentations. *Proc. of the CHI 99 Conference on Human Factors in Computing Systems*, ACM Press, (pp.354-361).
- [9] Sphere Research Labs, ideone.com, <http://ideone.com/>. (Referred on May 2, 2010)