Value Trace Problems for Code Reading Study of JavaScript Programming

Khin Thet Mon, Khaing Hsu Wai, Nobuo Funabiki, and Htoo Htoo Sandi Kyaw

***Abstract*—*JavaScript programming* takes critical roles in developing web application systems. Unfortunately, JavaScript programming is not fully educated in most universities, although its study may not be easy for students since the code is usually made by special composing styles. To assist self-study of JavaScript programming, currently, we are developing *JavaScript programming learning assistant system (JSPLAS)* by extending our *JPLAS* works for Java programming. In JPLAS, the *value trace problem (VTP)* is offered for code reading study by novice students. A VTP instance consists of a source code and a set of questions, where each question asks the actual value of an important variable or an output message in the source code. The correctness of any answer is marked through *string matching* with the correct one. In this paper, we study the VTP for code reading study of JavaScript programming. We generate 57 instances using source codes on basic grammar concepts and confirm the effectiveness through applications to 45 university students in Myanmar and Japan.**

***Index Terms*—JavaScript Programming, value trace problem, coding reading, grammar concepts.**

I. Introduction

Currently, *JavaScript* is the most popular programming language in the world [1]. *JavaScript programming* has been adopted in a variety of mobile/desktop application developments including game applications. *JavaScript programming* has several remarkable features compared with other major programming languages. Firstly, it is easy to get started with by abstracting away most of the complex details of programming and to focus on learning how to program. Secondly, it allows coding and testing on a Web browser without setting up any development environment. Thirdly, it becomes the essential web technology along with HTML and CSS to build interactive websites and it has many prewritten functionalities to help programmers develop complex systems easily.

In web application systems, *JavaScript* programs are used to detect the browser type used by a person and to customize the webpages depending on it. They are also used for security password creations, check forms, interactive games, and special effects. As a result, JavaScript programming has become common in building mobile applications and creating server- based applications.

Manuscript received November 20, 2021; revised January 19, 2022.

K. T. Mon, K. H. Wai, and N. Funabiki are with the Electrical and Communication Engineering Department, Okayama University, Okayama, Japan (e-mail: p3x78b2r@s.okayama-u.ac.jp, pjsu9uam@s.okayama-u.ac.jp, funabiki@okayama-u.ac.jp).

H. H. S. Kyaw is with Division of Advanced Information Technology & Computer Science, Tokyo University of Agriculture and Technology, Koganei, Japan (e-mail: fq5531@go.tuat.ac.jp).

Under the above-mentioned situations, a lot of people have now started to interest in learning *JavaScript programming*. Their learning styles will depend on their environments as well as their needs. For university students, the high-quality learning tools of *JavaScript programming* have been highly demanded, especially for self-study at home since many students have no opportunities of taking the courses at schools. Unfortunately, most universities are still now teaching *JavaScript programming* in a part of a web-programming course.

Heretofore, we have developed the *Java programming learning assistant system (JPLAS)* to assist self-study of Java programming [2]. Java is widely used in IT societies as the practical programming language. Students can learn *code reading* and *code writing* skills that are necessary to develop or create new programs. In *JPLAS*, we have the step-by-step programming study strategy to cultivate code reading and code writing skills of novice students. For this goal, *JPLAS* offers various types of exercise problems that have different difficulty levels to cover various students at various learning levels and stages.

The types of exercise problems in JPLAS include the grammar-concept understanding problem (GUP), the value trace problem (VTP), the element fill-in-blank problem (EFP), the code amendment problem (CAP), the code modifying problem (CMP) and the code writing problem (CWP). For any type of the problems in JPLAS, the answer from a student is marked automatically, to support programming self-studies by the students.

To offer self-study environments of *JavaScript programming* to novice students, we have started the project of *JavaScript programming learning assistant system (JSPLAS)*, by extending the works for JPLAS. In JPLAS, the *value trace problem (VTP)* is designed for novice students to study basic grammar concepts and programming skills through code reading [3]. A VTP instance consists of a source code, a set of questions, and their correct answer strings. Each question requests to answer the value of an important variable or an output message in the source code. The correctness of any student answer is marked through string matching with the stored correct string.

In this paper, we study the value trace problem (VTP) in JSPLAS, for code reading study of JavaScript programming by novice students. To study basic grammar concepts of JavaScript programming and source code implementations using them, we collect 57 source codes containing basic grammar concepts for generating the VTP instances. Then, we analyze the important variables, add their standard outputs in the codes, and prepare the correct answers, manually. After that, we generate the corresponding VTP

doi: 10.18178/ijiet.2022.12.5.1637

instances by applying the program for generating the answer interface.

For evaluations, we assign the generated 57 VTP instances to students in Myanmar and Japan who are studying JavaScript programming and confirm the effectiveness of the proposal in studying basic JavaScript programming. Here, we divide the instances in two groups, namely, 32 in VTP-I and 25 in VTP- II, to help the students solve them step-by-step.

The rest of this paper is organized as follows: Section II reminds the importance of JavaScript programming. Section III reviews our preliminary works to this study. Section IV presents value trace problems (VTPs) for JavaScript programming. Section V evaluates them through applications to students. Section VI introduces related works in literature. Finally, Section VII concludes this paper with future works.

II. Importance of JavaScript

JavaScript is often abbreviated as *JS*. *JS* is a scripting language that has been commonly used to implement dynamic behaviors of web pages in *web application systems*. Currently, *JS* is the dominant client-side scripting language for *web client programming*, where 97% of the websites use it. Therefore, *JS* is one of the powerful and flexible programming languages, which is essential for developing web applications.

A. Web Client Programming

JS is one of the core technologies of web client programming. *JS* can be used to create dynamic and interactive web pages by combining with *Hypertext Markup Language (HTML)* and *Cascading Style Sheet (CSS)*. *HTML* is the fundamental building block of any web page. It can be used to create content of webpages. It can control the layout of the content and gives the structure for the web page design. To stylize the website, *CSS* can apply style to the web page elements. On the other words, CSS describes the style of the *HTML* content, such as layout setting, background colors, and fonts. It targets various screen sizes to make web pages responsive. Therefore, it can be used to primarily handle the "look and feel" of a web page. *JS* is used for interactivity of web pages. It is interacted with them following the *document object model (DOM)*. The way of how *JS* works is that it can add interactivity to a web page and handle complex functions and features. Therefore, *JS* is the programmatic code which enhances functionality.

B. Popularity of JavaScript

JS is primarily used in the web browser, enabling developers to create webpage content, manipulate data, draw graphics, and interact with the device running the browser through various APIs. *JS* becomes one of the world's most used languages and there are several reasons why *JS* is popular nowadays.

As the growth of the Internet, web application systems have become major players of computer systems, which makes web client programming using *JS* essential in web applications. *JS* does not need any environment setup and so it is flexible for the developers to use. Besides, *JS* can run everywhere, including devices like mobile phones, tablets, and laptops and on the client-side as well as the server-side.

This makes *JS* a universal language. Along with HTML, CSS, it's a core component of web technology as it provides interactivity to web pages in the browser. Moreover, *JS* offers a wide range of frameworks and libraries that help developers create complex applications.

According to the GitHub and Stack Overflow Survey, it depicts that *JS* is the most popular programming language and it has maintained the top-ranked position for several years [1].

C. Features of JavaScript Programming

JS is one of the core technologies in the web client programming. *JS* can be used to dynamically change/update contents, control multimedia contents, animate images in the web page.

There are a lot of remarkable features in JS. First, JS is rich in libraries. The JS libraries contain various functions to perform specific tasks on the web page. Since these libraries are very powerful and can ease the programmer's workloads for creating a website or a web application [4].

Moreover, JS libraries are easy to learn, and can boost the speed of the web page to accomplish a specific task. As JS is an interpreted scripting language, it does not need the compilation step. The script written inside JS is processed line by line. An interpreter in the browser reads over the JS code, interprets each line, and runs it.

More modern browsers use a technology known as Just-In-Time (JIT) compilation, which compiles JS to executable bytecode, just as it is about to run. Besides, JS is the lightweight language, which is meant for client-side execution. This lightweight nature of JS is the great feature.

Another interesting feature of JS is that it supports dynamic typing, which means the data types of the variables are defined based on the stored values. In some programming languages such as Java, a particular variable will store a certain type of the data, whereas in JavaScript, we do not have to provide the data type for declaring a variable. In JavaScript, we just need to use "var" or "let" keyword before the variable name to declare a variable without worrying about its type [5].

JS is a prototype-based scripting language. This means JS uses prototypes instead of classes or inheritance. In programming languages like Java, we create classes first, and then, create objects for those classes.

However, in JavaScript programming, we can define the object prototype first. Then, more objects can be created using this object prototype. JS is commonly used for implementing client-side validation functions, because every website has a form in which users enter values, and to make sure that users enter the correct value, we must put proper validations in place, both on the client-side and on the server-side.

III. Preliminary Works

In this section, we review our preliminary works on the value trace problem (VTP) [6].

A. Design Goals of Value Trace Problems

The design goals of the VTP are as follows:

1) a variety of source codes effective in programming study are depicted with full forms to novice students,
2) students can correctly answer the questions only by carefully reading and understanding the codes, and
3) any student answer can be marked through string matching automatically.

To compose a VTP instance, a source code, a set of questions with the answer forms (blanks), and the correct answers are necessary. A question asks the messages or the values from the code to the standard output. Thus, to make an efficient VTP, the corresponding standard output statements should be added into the original source code, in order to display the important messages or the values of the key variables in the source code. Unfortunately, the selection of important messages or key variables are manually selected in our works.

B. Concept of the Value Trace Problems

A VTP instance can be generated by the following procedure:

1) to select a source code that is suitable for studying basic grammar concepts or a fundamental data structure/algorithm,
2) to find important messages and key variables in the code,
3) to add the corresponding standard output statements of them,
4) to prepare the questions of asking the messages/values at the standard output and their correct answers,
5) to put together the source code, the questions, and the correct answers into one text file,
6) to run the program with the text file to generate the HTML/CSS/JavaScript files for the answer interface on a browser, and
7) to register the generated VTP instance in the assignment to students.

IV. Value Trace Problems in JSPLAS

In this section, we present the 57 VTP instances for studying basic grammar concepts of JavaScript programming in JSPLAS.

A. VTP Instances for Basic Grammar Concepts

As we discussed before, we divide them into two groups, 32 for VTP-I and 25 for VTP-II. Tables I and II show the grammar concept, the number of lines (LOC) in the source code, the number of answer forms to be filled in, and the average correct answer rate of the students for VTP-I and VTP-II respectively.

Unfortunately, the selection of important messages or key variables are manually selected in our works.

B. Examples of VTP Instances

Here, we show an example of the generated VTPs in this study. Source code shows the adopted source code for studying "JavaScript Data Type Usage" in VTP instance ID=1 of the basic grammar of VTP-I. Question shows the corresponding question with five answer forms. The correct answers to them are undefined, null, 100, true and Hello. A student needs to read the source code carefully to understand it and fill in the forms correctly.

Fig. 1 illustrates the user interface for solving VTP instance ID=25 of VTP-II. This instance intends studying "Arrow Functions". After filling all answer forms, the student has to click the blue “Answer” button. If the answer is not correct, the background color of the form will change into red. Otherwise, the background color will be white. The student can submit the answers repeatedly until all the answers to be correct.

Source Code

```
main( ) {
    var undeclaredVar;
    var obj = null;
    var num = 100;
    var inProgress = true;
    var Greeting = "Hello";

    console.log(undeclaredVar); // undefined
    console.log(obj); // null
    console.log(num); // number
    console.log(inProgress); // boolean
    console.log(Greeting);    // string
}
```

Questions

What is the value of undeclaredVar? _1_
What is the value of obj? _2_
What is the value of num (approximately two decimal place)? _3_
What is the value of inProgress? _4_
What is the value of Greeting? _5_

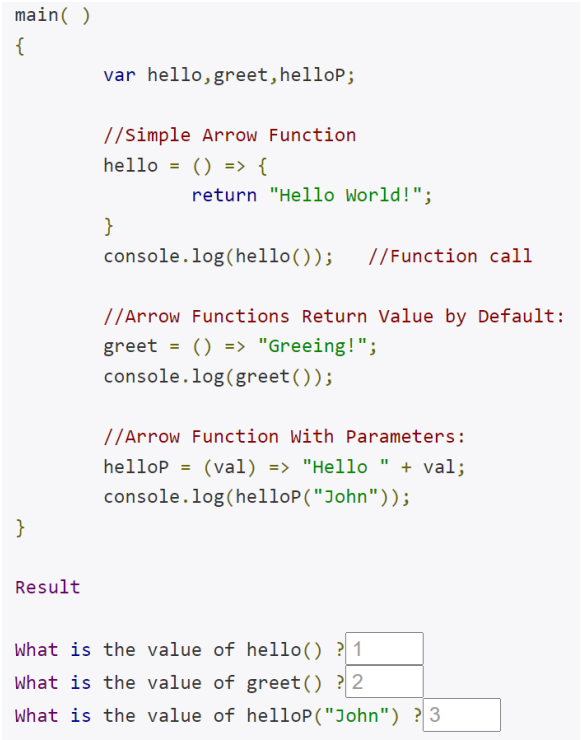

Fig. 1. User interface of VTP instance.

V. Evaluation

In this section, we evaluate the generated 57 VTP instances through applications to 45 university students in

Myanmar and Japan. Among them, 11 third-year undergraduate students in Myanmar have studied JavaScript programming as one subject in the web programming course. On the other hand, 34 first- year master students in Japan first studied JavaScript programming by reading the provided references for the evaluations before solving them. It is noted that one student did not reply to the answers for VTP-I.

A. Individual Student Result for VTPs

Fig. 2 and 3 show the number of students for each correct answer rate range for VTP-I and VTP-II, respectively. For VTP-I, six students among 44 solved with 100% correct rate. There are 25 students who achieved 95% to 99% correct rate. Among the remaining 13 students, eight did 90% to 94%, five did from 80% to 89%. For VTP-II, 21 students among 45 achieved 100% correct rate. 20 students achieved 95% to 99% correct rate, two students did 90% to 94%, two students did 80% to 89%. The average correct rate for VTP-I is 96.55%and for VTP-II is 98.30%. The smallest rate for VTP-I is 81% and that for VTP-II is 83%. Thus, the generated VTP instances are at proper levels for students to start studying JavaScript programming.

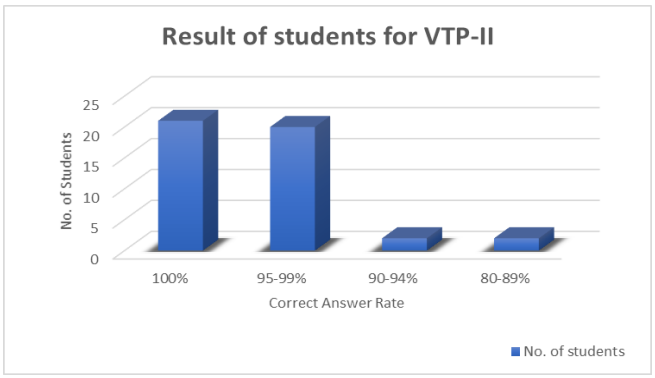

Fig. 2. Student results for VTP-I.

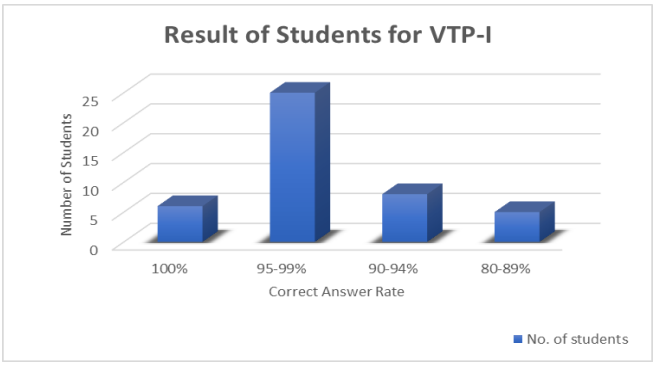

Fig. 3. Student results for VTP-II.

B. Individual Instance Results

Tables I and II include the average correct answer rate by the students for each VTP instance in VTP-I and VTP-II, respectively. Most of the instances are solved well by the students. However, in some instances, the correct answer rate does not reach 90%. We will discuss these hard VTP instances.

In VTP-I, the four instances with ID = 2, 9, 10 and 14 show the very low rates that are smaller than 80%, and the six instances with ID = 3, 7, 8, 18, 19, and 20 also give the low rates that are smaller than 90%.

TABLE I: VTP INSTANCES FOR VTP-I

ID	Basic Grammar Concepts	LOC	# of forms	Avg correct rate
1	Data Types	13	5	98%
2	Arithmetic Operators	32	11	48%
3	Assignment Operators	48	7	86%
4	Functions	27	6	95%
5	Objects	22	3	98%
6	Strings	15	5	98%
7	Backslash Characters	25	3	82%
8	String Specifications	28	4	89%
9	String Extraction	39	9	52%
10	String Replace and Trim	34	6	77%
11	String Conversion	41	9	98%
12	Number Methods	42	15	100%
13	Number Methods	29	8	100%
14	Number to String	40	10	57%
15	Number to String	28	8	98%
16	Arrays	55	11	84%
17	Arrays Association	29	4	91%
18	Basic Arrays Methods	36	8	89%
19	Basic Arrays Methods	29	6	89%
20	Arrays Changing	41	7	89%
21	Arrays Splicing	50	9	98%
22	Arrays Splicing	48	12	98%
23	Arrays Merging	39	8	91%
24	Arrays Sorting	46	8	93%
25	Conditions (if)	11	1	100%
26	Conditions (if, else)	13	2	100%
27	Conditions (if, elseif, else)	15	3	100%
28	Conditions (switch)	58	3	98%
29	Looping (for)	13	11	100%
30	Looping (do while)	9	5	100%
31	Looping (while)	9	5	100%
32	Looping (while)	9	3	100%
	Average	30.4	6.7	90.27%
	Total (SD)	973	215	13.87%

The instance with ID = 2 is on arithmetic operator. The questions ask the values of 11 variables after applying operators. It seems that many students cannot calculate the values correctly, although they understand the behaviors of the operators. The instance with ID = 3 is on assignment operator. The questions ask the values of seven variables after applying operators. Again, it seems that many students cannot calculate the values correctly. The instance with ID = 7 is on back slash character. The special characters "\", "‘", and "}" have specific roles as the commands in a web page. To avoid the role, "}" must be inserted before the character. The four instances with ID = 8, 9, 10, and 14 are on String. The source codes use several library functions in this class such as slice, substr, replace, trim, toExponential, and toFixed. The three instances with ID = 18, 19, and 20 are on Array. The source codes use several library functions in this class such as length, push, pop, and shift.

In VTP-II, the instance with ID=7 is on uppercase check and has the low rate of 70%. The source code determines where the first character of each string is uppercase or not using a regular expression. It seems that many students may not understand the regular expression.

TABLE II: VTP INSTANCES FOR VTP-II

ID	Basic Grammar Concepts	LOC	# of forms	Avg correct rate
1	Data Types (I)	33	4	100%
2	Data Types (II)	16	2	95%
3	Comparison Operators	51	12	95%
4	Logical Operators	38	3	95%
5	Conditional Operators	11	1	95%
6	Precedence Associativity	20	4	100%
7	Upper Case Check	24	2	70%
8	Vowel Checking	23	1	100%
9	Prime Number Check	34	3	98%
10	Data Types Check	30	3	91%
11	Perfect Number Check	28	2	91%
12	Global Functions (Eval())	15	2	100%
13	Global Functions (isFinite())	17	2	95%
14	Global Functions (isNaN())	37	12	93%
15	Global Functions(parseFloat())	20	7	93%
16	Global Functions (parseInt())	31	12	95%
17	Global Functions (Number())	24	9	91%
18	Numeric Array Sorting	46	3	95%
19	Dates	33	5	98%
20	Var VS Let – Block Scope	18	6	98%
21	Var VS Let – Redeclaration	17	2	98%
22	Var VS Let – Looping	16	2	98%
23	Variable Declaration (const)	25	5	93%
24	Var VS Const – Block Scope	11	3	98%
25	Arrow Functions	18	3	95%
	Average	25.4	4.36	94.91%
	Total (SD)	636	110	5.73%

C. Discussion

The average correct answer rate among the students is 96.55% for VTP-I and 98.30% for VTP-II. Thus, it is confirmed that the VTP instances in this paper are proper for JavaScript programming novice students to study. However, there exist several hard instances that resulted in low correct answer rates. To increase the correct rates to them, we will improve the references on these topics in JavaScript programming learning assistant system (JSPLAS), which will be in future works.

VI. RELATED WORKS IN LITERATURE

In this section, we discuss related works to this study in literature.

In [7], G`omez-Albarr`an reviewed several tools to support teaching and learning of programming, and categorized them into four groups, as tools including simple reduced development environment, example-based environments, tools based on visualization and animation, and simulation environments.

In [8], Brusilovsky *et al*. presented QuizPACK that asks the value of a particular variable or a string with some fragment of a program. They demonstrate that it significantly improved the knowledge of semantics. Their approach is similar to our works in this paper. We will compare this study.

In [9], Kordaki presented a computer-based problem-solving environment named LECGO (Learning Environment for programming using C using Geometrical Objects) for learning C programming by beginners. It emphasized on multiple external representations in student learning, motivation through performing problem-solving activities from the familiar and meaningful context, the active participation of students by using hands-on experience, appropriate feedback to aid self-corrections, and holistic, activity-based, multi-media, multi-representational and multi-layered content for learning basic concepts of C programming.

In [10], Lee *et al*. presented Gidget, a game where the eponymous robot protagonist is cast as a fallible character that blames itself for not being able to correctly write code to complete its missions. Players learn programming by debugging its problematic code.

In [11], Li *et al*. presented a game-based learning environment to support novice students learning programming. It exploits game construction tasks to make the elementary programming more intuitive to learn, and comprises concept visualization techniques, to allow students to learn key concepts in programming through game object manipulation.

In [12], Hwang *et al*. proposed the Web-based programming assisted system for cooperation called WPASC, designing learning activity for facilitating cooperative programming learning, and investigated cooperative programming learning behavior of students and its relationship with learning performance.

VII. CONCLUSION

This paper studied the value trace problem (VTP) for JavaScript programming study by novice students in JSPLAS. 57 VTP instances were generated using various source codes for basic grammar concepts of JavaScript. The application results to 45 university students in Myanmar and Japan confirmed the effectiveness of them, where every student achieved the 80% or higher correct rate on average. In future works, we will improve the references on the hard topics, generate new VTP instances for studying advanced grammar concepts, common libraries, or data structure and algorithms, and apply them to students in JavaScript programming courses.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

K. T. M, K. H. W and H. H. S. K conducted the research. K. T. M and K. H. W analyzed the data and wrote the paper. N. F reviewed the paper and finalized the paper. All the authors had approved the final version.

ACKNOWLEDGMENT

We would like to thank to the students in Okayama University, Japan, and Thanlyin Technology University, Myanmar, to answer the VTP instances and give us valuable comments. They are inevitable to complete this paper.

REFERENCES

[1] JavaScript is the most popular programming language: Stack overflow survey. [Online]. Available: https://fossbytes.com/javascript-most-popular-programming-language

[2] S. L. Ao *et al*., *IAENG Transactions on Engineering Sciences - Special Issue for the International Association of Engineers Conferences 2016 (Volume II)*, pp. 517-530, World Sci. Pub., 2018.

[3] 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," Inform. Eng. Exp., vol. 1, no. 3, pp. 9-18, Sep. 2015.

[4] Javascript-features. [Online]. Available: https://www.studytonight.com/javascript/javascript-features

[5] features-of-javascript. [Online]. Available: https://data-flair.training/blogs/features-of-javascript/.

[6] X. Lu, N. Funabiki, H. H. S. Kyaw, S. L. Aung, and N. K. Dim, "A study of value trace problems for code reading study of C programming," in *Proc. WANC*, pp. 445-459, Nov. 2020.

[7] M. G`omez-Albarr`an, "The teaching and learning of programming: a survey of supporting software tools," *Comput. J.*, vol. 48, no. 25, pp. 130-144, 2005.

[8] P. Brusilovsky and S. Sosnovsky, "Individualized exercises for self-assessment of programming knowledge: An evaluation of QuizPACK," *ACM J. Edu. Res. Comput*., vol. 5, no. 3, pp. 1-22, Sep. 2005.

[9] M. Kordaki, "A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation," *Comput. Edu*., vol. 54, pp. 69{87, 2010.

[10] M. J. Lee and A. J. Ko, "Personifying programming tool feedback improves novice programmers learning," in *Proc. ICER*, pp. 109-116, Aug. 2011.

[11] F. W. B. Li and C. Watson, "Game-based concept visualization for learning programming," in *Proc. MTDL*, pp. 37-42, Dec. 2011.

[12] W.-Y. Hwang, R. Shadiev, C.-Y. Wang, and Z.-H. Huang, "A pilot study of cooperative programming learning behavior and its relationship with students' learning performance," *Comput. Edu*., vol. 58, pp. 1267-1281, 2012.

Khin Thet Mon received the B. E. degree in information technology from Yangon Technological University, Myanmar, in 2006. After working in several companies, she joined Ascender Japan in 2020. She is currently a Ph. D. candidate in Graduate School of Natural Science and Technology Computer Science, Okayama University, Japan. Her research interests include software development management and web application systems. She is a member of IEICE.

Khaing Hsu Wai received the B. E. and M. E. degrees in information science and technology from University of Technology (Yatanarpon Cyber City), Myanmar, in 2016 and 2020, respectively. She is currently a Ph. D. candidate in Graduate School of Natural Science and Technology Computer Science, Okayama University, Japan. Her research interests include educational technology and web application systems.

Nobuo Funabiki received the B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, and the M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991respectively. From 1984 to 1994, he was with Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became an associate professor in 1995. He stayed at University of Illinois, Urbana-Champaign, in 1998, and at University of California, Santa Barbara, in 2000-2001, as a visiting researcher. In 2001, he moved to the Department of Communication Network Engineering (currently, Department of Electrical and Communication Engineering) at Okayama University as a professor. His research interests include computer networks, optimization algorithms, educational technology, and Web technology. He is a member of IEEE, IEICE, and IPSJ.

Htoo Htoo Sandi Kyaw received the B. E. and M. E. degrees in information science and technology from University of Technology (Yatanarpon Cyber City), Myanmar, in 2015 and 2018, and Ph. D. in Information Communication Engineering from Okayama University in 2021, respectively. She is currently an assistant professor in Division of Advanced Information Technology and Computer Science, Tokyo University of Agriculture and Technology, Koganei, Japan. Her research interests include educational technology and web application systems.