Web Application for Automatic Code Generator Using a Structured Flowchart

Chanchai Supaartagorn
Department of Mathematics Statistics and Computer, Faculty of Science, Ubon Ratchathani University
Ubonratchathani, Thailand
chanchai.s@ubu.ac.th

***Abstract*—A flowchart can be a graphic diagram representation of a programming logic. There are shapes and connectors that represent the different types of actions or steps in a process. A flowchart is a very important tool in the planning phase in a program development cycle. Programmers can use it to design and develop an algorithm of a program. Moreover, a flowchart is also very effective for visual learners to write and comprehend algorithms in computer programming courses. This paper aims to provide a tool which serves as an automatic code generator using a structured flowchart. The tool is composed of basic flowchart shapes to be combined into a structured flowchart that can be converted into source codes. In addition, the system's performance has been evaluated by two groups: 5 experts and 93 general users. The results showed the average values of the satisfaction levels were 4.48 and 4.27 with standard deviations at 0.59 and 0.64 for the experts and the general users respectively. It was found that the system performance of the tool reached an agree level. It was revealed that the developed system can be used precisely as intended in an effective manner.**

Keywords—Automatic code generator; Structured flowchart; Programming logic; Algorithm

I. INTRODUCTION

A computer programming course is one of the core subjects of the curriculum in computer-related fields, including computer science, computer engineering, Information Technology and business computer. The most important problem for a student studying a computer programming course is understanding the syntax and logic of programming languages. The syntax of program refers to the rules of the programming language. Every programming language has rules governing its word and punctuation usage. However, a program without syntax errors might still produce an incorrect result. For a program to work properly, a programmer must develop correct logic [1]. Several research articles have shown a decline in the number of students in computer programming courses. For example, a study on an active learning approach at the University of Mary Washington confirmed the notion that learning how to program is considered to be a difficult task to the majority of students and this has been a prime reason for students' dropping out from computer courses [2]. A study estimated that between 25 to 80 percent of students dropped out from their first year computer science classes due to the difficulty they faced in learning how to program [3]. In addition, some instructors changed the teaching methods, such as changing the programming language, using a different textbook and taking additional measurements. Nevertheless, the problem of difficulty in learning a computer program persisted [4]. To alleviate the problem, one of the solutions is to let students present the programming logic in form of a flowchart. A flowchart plays an important role in system requirement analysis, preliminary design and a detailed design aspect [5]. A study on teaching computer programming to adult students at Tairawhit Polytechnic, New Zealand [6] showed that 40% of the beginners preferred to use a flowchart to understand programming, 40% wanted to use pseudo code, and 20% wanted to use a real language. A nifty tool for studying program and system behaviors have shown that flowcharting can be very effective for visual learners to write and comprehend algorithms [7]. Recently, there was a research study on using flowchart for system programming course for the 3rd year student in Computer Science and Engineering at Walchand Institute of Technology, India [8]. There were five steps of the experiment. 1) Blackboard teaching 2) Conducting a pre-test 3) Using animated flowchart with example activities 4) Conducting a post-test and 5) Checking the long term effect of the activity. The results showed the students' performance improved in the post-test as compared to that in the pre-test. In addition, the research evaluated the feedback to understand student's perception. The results showed that 100% of students were satisfied with this activity.

This research proposes a tool serving as an automatic code generator using a structured flowchart via a web browser. The tool comprises various symbols required to create a flowchart. Users can convert the flowchart into correct source codes written in Java and PHP programming languages. The key feature of the tool is its debugging support, which is the process of fixing errors in a computer program. Source codes written in Java and PHP can be copied or saved for developments in the other integrated development environments (IDE). In addition, the system can also save the flowchart files for later use. The tool was developed by using JavaScript language, GoJS library and CodeMirror library. Moreover, ExtJS framework was used for building an interactive user interface.

II. RELATED STUDIES

The program development life cycle is a set of phases and steps that are followed by developers to define, design, develop and maintain a computer program. Typical phase of the program development cycle [1] is shown in Fig. 1.

978-1-5386-0497-7/17/$31.00 ©2017 IEEE

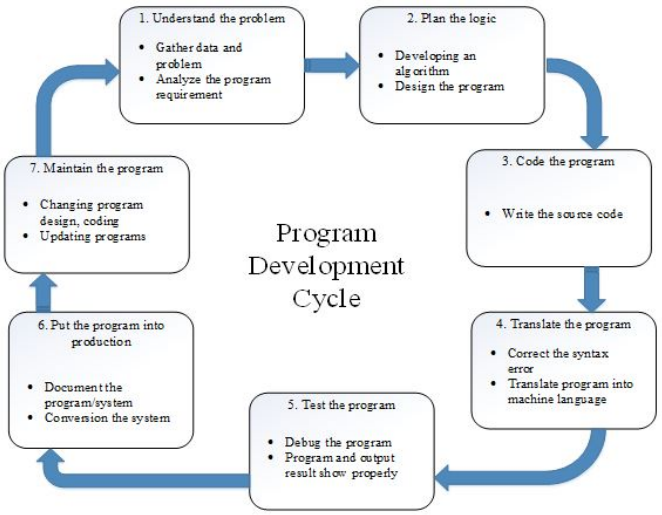

Fig. 1. Program development cycle.

The heart of a programming process lies in planning the program's logic (Phase 2). During this phase of the process, the programmer plans the steps of the program, deciding what steps to include. The most common planning tool is a flowchart. According to a conceptual model for learning program research, the visual planning tools (such as flowcharts and hierarchy charts) have been introduced to simplify learning program structure. It is easier to monitor the code and understand the flow [4]. We will focus on the tool development for creating a flowchart. The flowchart can be automatically generated into source codes. This tool will help reduce syntax errors and save time for coding a program in the next phase.

There is a great deal of research about the automatic generation of code from flowcharts available today. For example, using a flowchart for teaching computer programming [9]. Users draw a flowchart by using the drawing tool. Then, the translation module will translate the flowchart into C source codes. Moreover, the system can proceed with the compilation and linking of C source code. Problem Analysis Diagram (PAD) for the generation of source code [10]. The system automatically converts a structured flowchart to PAD that enhances the readability of the algorithm, reduces the difficulty of the system design and improves the reliability and robustness of the software. However, the tools created by these researches studies have all run on windows-based applications, while the tool in this research could be developed in a web-based application. Users can use and access the tool via a web browser. Moreover, the users also have the option of converting the flowchart into Java source code or PHP source code.

Next, we reviewed the structure of the algorithms to be used in determining the scope of the system. There are three basic structures: sequence, selection and iteration. Sequence structure performs actions or tasks in order. Selection structure or decision structure performs check of condition, taking one or two actions. Iteration structure or loop structure repeats actions while a condition remains true. Fig. 2 shows three basic structures of a flowchart.

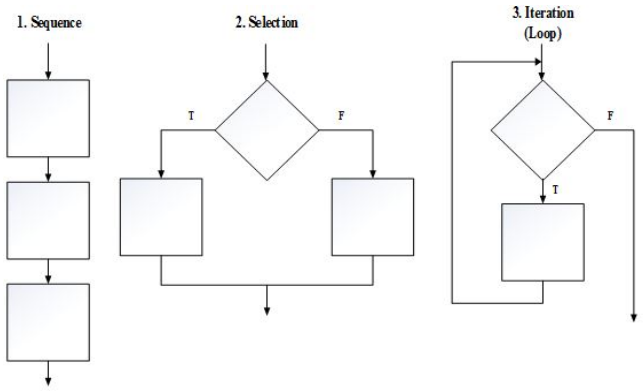

Fig. 2. Three basic structures of a flowchart.

These basic structures can be coordinates, they can include each other: stacking, nested selection, nested loop, nesting (selection-loop) and nesting (loop-selection). Fig. 3 shows the stacking structure that combines three basic structures end to end.

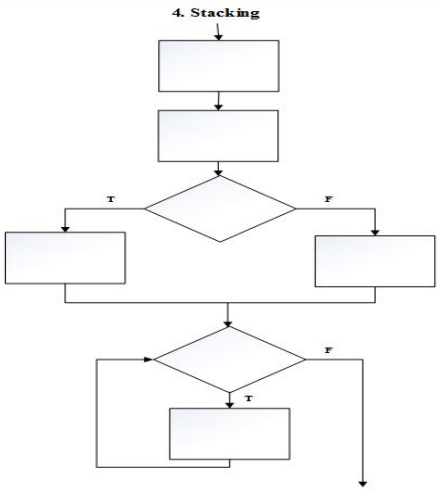

Fig. 3. Stacking structure of a flowchart.

The nested selection structure has a condition within a condition. There is more than one decision to be made before appropriate action can be taken. The nested loop is a loop within a loop, an inner loop within the body of an outer one. It is characterized by two or more iteration statements that are placed in a nested form. Fig. 4 shows a nested selection structure and nested loop structures of the flowchart.

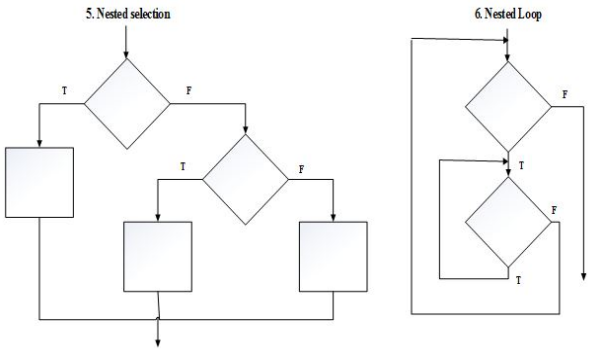

Fig. 4. Nested selection structure and nested loop structure of a flowchart.

The nesting structure performs the placing of one structure within another. The nesting (selection-loop) structure placing a loop structure is within a selection structure. The nesting (loop-selection) structure placing a selection structure is within a loop structure. Fig. 5 shows the nesting (selection-loop) structure and the nesting (loop-selection) structure of a flowchart.

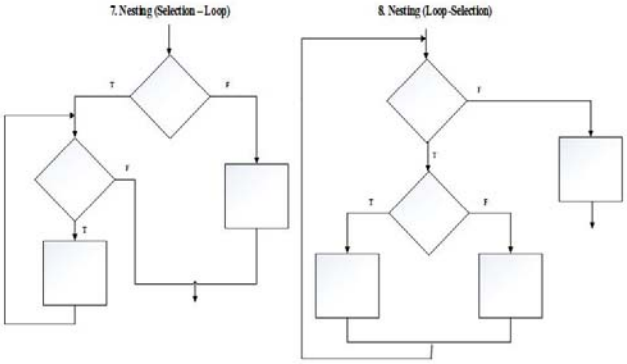

Fig. 5. Nesting (selection-loop) structure and nesting (loop-selection) structure of flowchart.

From the literature reviews, we designed and developed a tool for automatic code generator using a structured flowchart. There are eight structures of the flowchart, which can be used to convert a flowchart into source code.

III. System Design

The automatic code generator tool was developed by using JavaScript language, GoJS library and CodeMirror library. Fig. 6 illustrates the architecture of the tool.

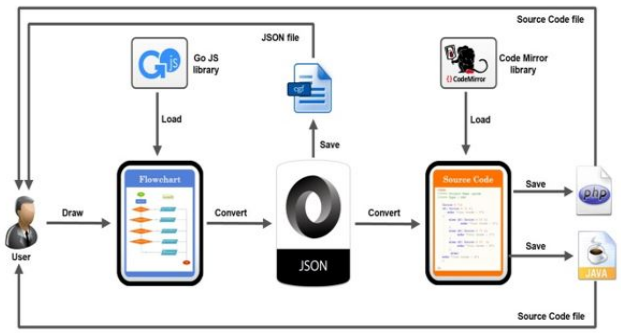

Fig. 6. Architecture of an automatic code generator.

The process flow of the tool can be described as follows.

1. A user draws a flowchart by using the drawing tool loaded from GoJS library.
2. The user converts the flowchart into JSON format (JavaScript Object Notation), e.g. the flowchart of the sum of the numbers 1-10. The example of flowchart and JSON format is as shown in Fig. 7.
3. The JSON format will be converted into Java source code or PHP source code. For this step, CodeMirror library is loaded into the system to help add color to the texts written in a given language.
4. The flowchart and source code can be saved for later use.

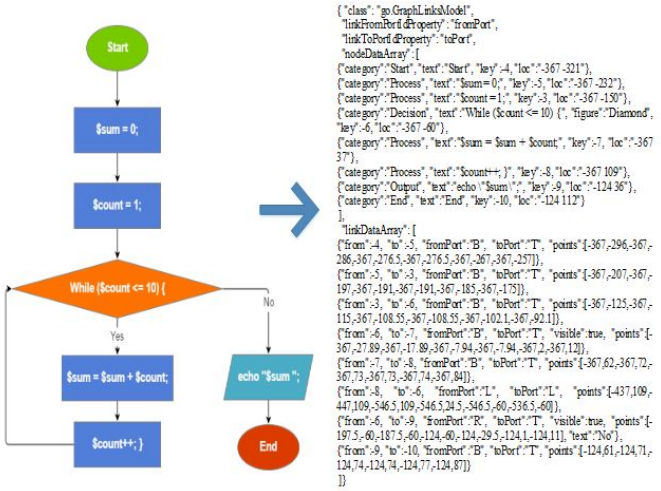

Fig. 7. An example of converting flowchart into JSON format.

IV. System Implementation

In this section, the interface of the tool and an example of the process of automatic code generating are presented. When a user runs the tool through a web browser, the layout of the user interface will appear as shown in Fig. 8.

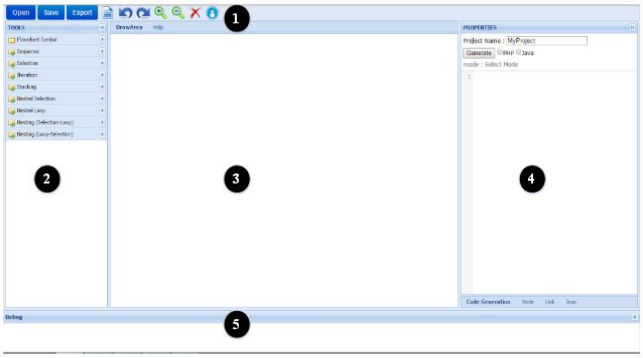

Fig. 8. The layout of user interface.

The interface is divided into five areas:

Number 1 (Toolbar). It is a set of buttons and icons namely open, save, export, new, undo, redo zoom in, zoom out, delete and about me.

Number 2 (Drawing tool). It prepares the symbols required to draw a flowchart. There are two parts: basic flowchart symbols and models of flowcharts. The basic flowchart symbols are composed of start/end, input, output, process and decision. The models of flowcharts are examples of each structure that help a user to draw flowcharts more easily.

Number 3 (Drawing view). This section is an area to draw a flowchart. A user drags and drops symbols from the drawing tool. In addition, a user can use a model example that helps a user to draw flowcharts more easily.

Number 4 (Code view). This section is an area to show the source code. The user simply selects a language (PHP or Java) and clicks the Generate button.

Number 5 (Debugging). This section is an area to display an error message that informs of the problems in source code.

Next, we showed an example of a nesting (loop-selection) structure, including the examination scores of five students. Each score was checked to see whether they passed the test or not. The flowchart is illustrated in Fig. 9.

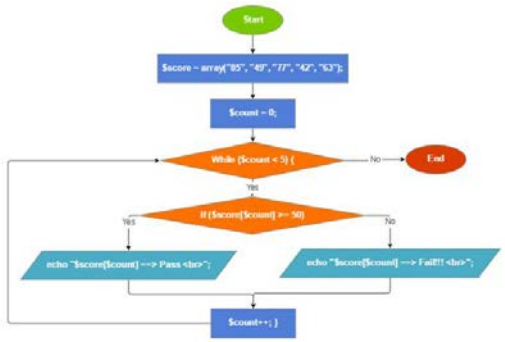

Fig. 9. The example of the flowchart.

After that, we converted the flowchart into PHP source code. The result is shown in Fig. 10.

Fig. 10. PHP source code after converting.

V. System Evaluation

The automatic code generator tool was evaluated by two groups of individuals. The first group consisted of experts in the field of information technology, who taught Information Technology courses, at the faculty of Science at Ubon Ratchathani University (5 instructors). The second group comprised general users, who were students studying in the Data structure and algorithms courses during the second semester, of the 2015 academic year (93 students). The system evaluation consisted of five parts: a functional requirement test, a function test, a usability test, a performance test and a security test. Five-point Likert-type questions were used in the questionnaire to assess the experts' satisfaction and the general users' satisfaction. The highest possible score was 5. The results were interpreted based on the calculations of means and standard deviations as shown in Table I and Table II.

TABLE I. Result of Experts' Satisfaction

Evaluation List	$\bar{X}$	*S.D.*	*Satisfaction Level*
Functional Requirement Test	4.23	0.70	Agree
Function Test	4.30	0.80	Agree
Usability Test	4.49	0.55	Agree
Performance Test	4.65	0.45	Strongly agree
Security Test	4.75	0.45	Strongly agree
Result Summary	4.48	0.59	Agree

TABLE II. Result of Generals' Satisfaction

Evaluation List	$\bar{X}$	*S.D.*	*Satisfaction Level*
Functional Requirement Test	4.27	0.65	Agree
Function Test	4.30	0.60	Agree
Usability Test	4.20	0.69	Agree
Performance Test	4.35	0.63	Agree
Security Test	4.24	0.61	Agree
Result Summary	4.27	0.64	Agree

From Table I and Table II, the evaluation results of users' satisfaction indicated the system has met the need of all users.

VI. Conclusion and Future Research

To facilitate the studying in a programming course, we have designed and developed a web application serving as an automatic code generator using a structured flowchart. Students and novice users can use the tool to understand the logic of programming. The tool is composed of various basic flowchart shapes (i.e. start/end, input, process, output and decision) needed to create a structured flowchart. Then, users convert the flowchart into the JSON format and the source code respectively. In addition, the tool was evaluated in five parts: a functional requirement test, a function test, a usability test, a performance test and a security test. The results showed that the system performance of the tool reached an agree level. It indicates that the developed system can be used correctly and effectively. In future research, we will add more flowchart shapes, such as document, external data, databases, sub-process. Moreover, we will improve the design of the tool in order to make it more user-friendly.

References

[1] Joyce Farrell, "Programming Logic and Design", Eighth Edition, Cengage Learning, 2015.

[2] Karen Anewalt. "Making CS0 fun: an active learning approach using toys, games and Alice". Journal of Computing Sciences in Colleges, vol. 23, issue 3, pp. 98-105, January, 2008.

[3] Janet Carter and Tony Jenkins, "Gender differences in programming?", in Proc. of the 7th Annual Conference on Innovation and Technology in Computer Science Education, pp. 188-192, June 24-28, 2002.

[4] Azad Ali, "A Conceptual Model for Learning to Program in Introductory Programming Courses", Issues in Informing Science and Information Technology, vol. 6, pp. 517-529, 2009.

[5] S. Sendall and W. Kozaczynski, "Model Transformation: The Heart and Soul of Model-Driven Software Development", IEEE Software, vol. 20, issue 5, pp.42-45, September 2003.

[6] Min Hu, "A Case Study in Teaching Adult Students Computer Programming", in Proc. of the 16th Annual NACCQ, Palmerston, pp. 287-291, July, 2003.

[7] Mark S, Hall. "Raptor: nifty tools". Journal of Computing Sciences in Colleges, vol. 23, issue 1, pp. 110-111, October 2007.

[8] Sunita M. Dol, "Fe.g.: An Animated Flowchart with Example to Teach the Algorithm based Course in Engineering", in IEEE Seventh International Conference on Technology for Education", pp. 49-52, 10-12 December 2015.

[9] Kanis Charntaweekhun and Somkiat Wangsiripitak, "Visual Programming using Flowchart", in International Symposium on Communications and Information Technologies, pp.1062-1065, October 18-20, 2006.

[10] WU Xiang-Hu, QU Ming-Cheng, Liu Zhi-Qiang and Li Jian-Zhong, "A Code Automatic Generation Algorithm Based on Structured Flowchart", Applied Mathematics & Information Sciences An International Journal, vol. 6, no.6, pp. 1-8, 2012.