Mind Mapping Tools for Creative Programming Design[1]

Tzone-I Wang
Professor
Department of Engineering Science,
National Cheng Kung University
Tainan, Taiwan
E-mail: wti535@mail.ncku.edu.tw

Yu-Lin Kuo
Graduate Student
Department of Engineering Science,
National Cheng Kung University
Tainan, Taiwan

Conference Key Areas: Engineering Skills, Sustainability and Engineering Education, Engineering Education Research.

Keywords: Programming Design, Mind Mapping, Creative Thinking.

INTRODUCTION

Creative thinking is highly regarded in recent years as an important ability for college students, and the same is to engineering colleges'. A variety of courses are established which harness old or new tools for enhancing the creative thinking ability of students. Among the tools, mind maps are a structured divergent thinking tool based on images, which can be used for stimulating a user's creative thinking. For engineering colleges' students, learning computer programming is to solve engineering problems and to advance their problem-solving ability. Creative thinking is now thought important for them to perceive problems and find the ways to solve the problems. Yet, few courses or methodologies have been established to cultivate and enhance students' creative thinking ability in engineering colleges' computer programming learning environments, lecture rooms, or e-learning platforms. Learning computer programming in the traditional way, students may not be so competitive in their programming career in such a fast changing era. It has being recognized that at the problem solving stage, creative thinking skills and tools can stimulate the creativity of students and help them solving problems. However to what extent creative thinking would help programmer doing better programming design? To give this question an answer, this research designs a methodology that forces students to perform divergent creative thinking before designing applications, builds a Computer Supported Learning (CSL) platform with several divergent thinking tools operating in compliance with the creative problem-

[1] This study is supported by the National Science Council (now Ministry Of Science and Technology) of Taiwan under the project contract No.: NSC 102-2511-S-006 -017 -MY3

solving mode, and conducts two experiments using this CSL with 51participants recruited from engineering college freshmen who enrol in a programming design course. Participants are evenly divided into an experimental group and a control group and both groups try to design and create applications from same topics. They, after examining some open data, enter a creative development stage to spawn as many ideas as possible, from which the participants choose and integrate applicable ideas for possible applications related to the open data. In the creative development stage, participants of the experimental group use a mind mapping tool of the platform to help stimulating and organizing their creative thinking, while those in the control group use only simple text lists to note their created ideas. Every participant selects one of the possible applications as the target for implementation. The results indicate that the experimental group, which uses a mind mapping tool, doing significantly better in the fluency, flexibility, and originality of creativity assessments than the control group, which uses just a text-noting tool. This concludes that the mind mapping tool can help learners brainstorming and generating more ideas in the creative development stage and therefore can help them performing significantly better creative programming designs.

1 BACKGROUND

Nowadays, creativity is broadly defined as a person's ability to create ideas or products that people judge as novel products [1] [2]. Most scholars believe creativity is a complex concept, which must be explained from multi-dimensional aspects. The recent confluence approach perspective dominates researches that see creativity being affected by the factors of intelligence, knowledge, thinking patterns, personality, motivation, and environment [3]. From cognitive psychology point of view, creativity is the key to solving a problem and the process of creation is the process of solving the problem. Apart from this, many recent researches propose the so-called technological creativity and engineering creativity from science and technology perspectives [7]. Technological creativity refers to the creative performance exhibited by individuals in scientific and technological activities. It has a unique characteristic that the creative ideas must be based on domain technology expertise and emphasis not just on putting forward a variety of ideas but also on creating results. In addition to the technological creativity, researches in engineering fields also find that engineering creativity is essential for engineering works in the process of idea generation, proposition formation, and idea materialization, which finally create novel products with promoted values in innovations [4].

The fast advance of Information Technology (IT) produces more and more products, software or hardware, like Facebook, Twitter and smart phones, and generates great impacts on human life. To success in such a rapidly changing world, IT companies have to develop innovative hardware and applications fast enough, which makes huge demands on software engineers. Creativity becomes one important ability for software engineering, especially for developing Innovative software applications in information technology industry, which makes companies more competitive. The infrastructure of software engineering is programming ability, which is not only the ability to construct application programs but also, via appropriate training enhancement, the ability of problem solving [5], [6], [7]. Programmers must be creative to producing innovative applications but the traditional programming courses in colleges rarely put efforts on.

Although courses are established for engineering college students to enhance their creative thinking ability, few of them are specifically for the programming environments, lecture rooms, or e-learning platforms. This study, therefore, sets to investigate how creative thinking can affect programmers in doing their programming design. For this purpose, the research needs a Computer Supported Learning (CSL) platform with divergent thinking tools that operate in compliance with the creative problem-solving mode for the experiments. A methodology is also needed to guide the experiments in the study. However, it will focuses on promoting programmers' creative programming design ability for solving problems, not on boosting their coding skills.

2 THE METHODOLOGY AND THE TOOL

A Computer Supported Learning tool and a methodology are constructed and designed in this study in an attempt to answer research questions.

2.1 The Methodology

The methodology is specifically designed to use divergent thinking tool in an attempt to make learners do better creative programming design. Students use mind mapping tool to brainstorm on a specified topic in the first stage. They generate ideas, anything associated to the topic, as many as possible and, record these ideas by nodes and links on a graphic mind map. After the brainstorming, students come to the application generating stage, where they converge their ideas into categories by considering if they can make applications out of them. Students can produce as many applications as possible, but choose only the top rated candidate for coding. The methodology uses the processes are depicted as in *Fig. 1*.

Fig. 1. The Methodology for creative programming design

2.2 The tool

This study constructs an online Computer-Supported Learning (CSL) e-platform that equips several divergent thinking tools and operates in compliance with the creative problem-solving mode. As shown in *Fig. 2*, major components in the CSL e-platform include a Brainstorming Module, an Idea Converging Module, and a Coding Module.

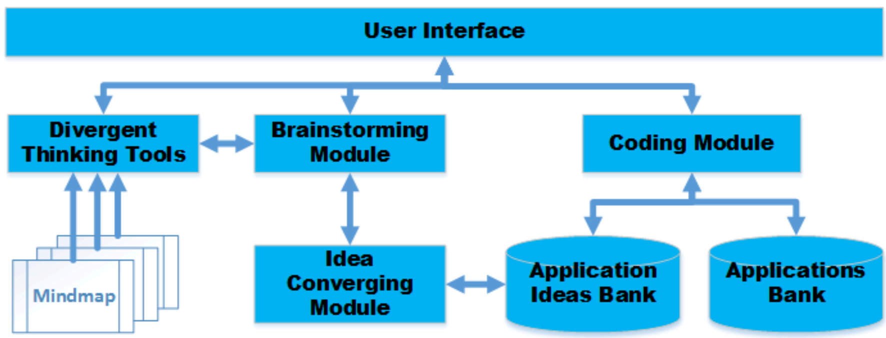

Fig. 2. The Computer-Supported Learning (CSL) platform

Learners can use a variety of divergent thinking tools in the Brainstorming Module to generate their creative ideas according to a given topic. One example of using the Mindmap tool is depicted in *Fig. 3*, in which the initial given topic is "a little apple", from which ideas are spawned and recorded by nodes and curved links. After finishes the divergent thinking, a user can go to the Idea Converging Module where a nested list is created and displayed, as in *Fig. 4* to reflect the ideas generated in the Brainstorming Module. This module operates in a convergent mode, in which users can create their application titles, choose for the applications the related ideas from the nested list, and save them to the Application Bank for later usages. These visualized operations make user easy in organizing their ideas into potential applications and in creating their applications as many as possible.

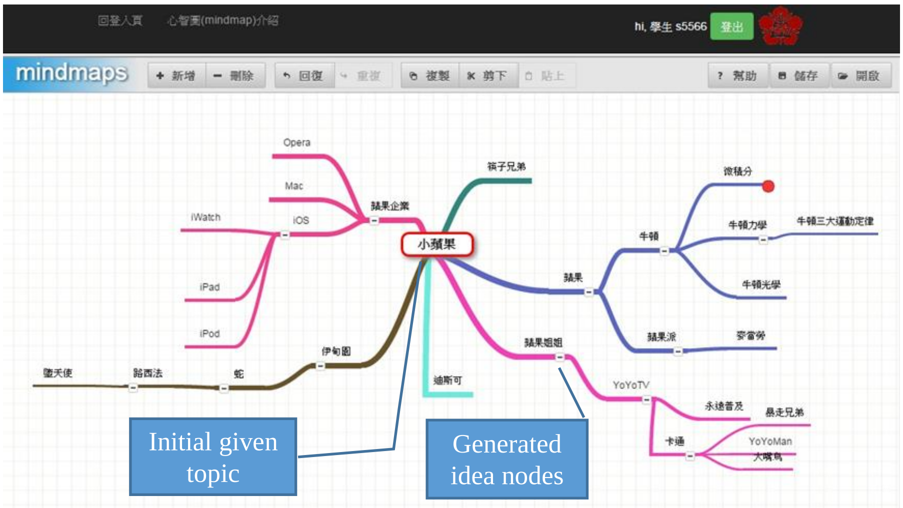

Fig. 3. Using the divergent thinking tool Mindmap

Fig. 4. Converge ideas into potential applications

2.3 Creativity Assessment

The original Torrance Tests of Creative Thinking (TTCT) [8] evaluates creativity on four scales: fluency, flexibility, originality, and elaboration, using verbal and figural tests. This study is to assess the effectiveness of mind mapping tools on students' creative programming design and to use the number of ideas generated in the brainstorming stage as the basis for creativity assessment. The original TTCT verbal and figural tests seem inappropriate, thus, in this study, is adapted by defining the first three scales based on the number of ideas generated as:

- **Fluency**: the number of categories of ideas integrated, i.e. the number of potential applications,
- **Flexibility**: how the applications divert from each other,
- **Originality**: how unique an application is among all the applications generated by all the students.

The study ignores the elaboration scale because the theme focus on whether students can, via the brainstorming tool, generate more creative potential applications and every of them counts, irrelevant to its detail.

3 EXPERIMENTS AND RESULTS

Two experiments are conducted in this study to answer two research questions:

Q1: whether students' minds will be constrained in some way when they know the ideas they brainstorm are for programming purpose?

Q2: whether students who use a divergent thinking tool for brainstorming will perform better than those who do not use any tool in programming design?

3.1 The experiments

The design of the experiments procedures is as shown in *Fig. 5*, which complies with the methodology of this study. Recruited for the experiments are 51 college freshmen who enrolled in a programming design course. Conventionally, they are evenly divided into an experimental group and a control group using s-distribution based on the results of a prior knowledge pre-test. The pre-test include two activities, the test of their prior knowledge on programming and a creative self-efficacy measurement with 22 items [9] [10] [11]. The ANOVA test on the prior knowledge between the two groups is depicted in *Fig. 6*, which confirms the two groups are evenly separated. The ANOVA test on creative self-efficacy measurement on each member of the two groups, as illustrated in *Fig. 7,* also confirms that the two groups have no significant difference on producing ideas from a topic before the experiments.

During the course of the experiments, the experimental group uses the Mindmap tool for brainstorming, while the control group uses only a simple text-noting tool.

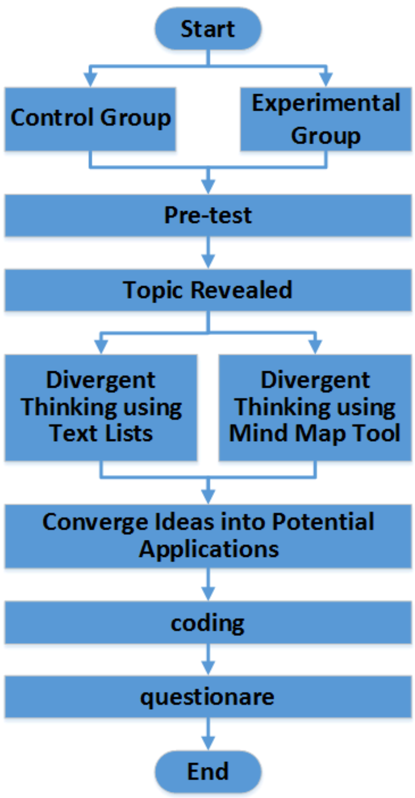

Fig. 5. The experiments procedures

Group	Number of participants	Average	STD
Experimental Group	26	57.615	21.929
Control Group	25	59.440	20.906

ANOVA

	Sum of square	Degree of freedom	F value	p-value
Between groups	11.009	1	.022	.882*
Within group	25051.783	51		

**p>0.05*

Fig. 6. One way ANOVA on the prior knowledge between groups.

Self-Efficacy Measurement

Items	Sum of square	F values	p-values
1~22 Items	0.000~ 1.923	0.000~ 2.299	1.000~0.136**

**All p-values >0.05

Fig. 7. One way ANOVA on the Self-efficacy measurement between two groups.

In the two experiments, participants in both groups use the same open data on two topics, i.e. Taiwanese snacks and pondage of reservoir related to the accumulated rainfall in Taiwan respectively, from which, they brainstorm ideas and converge their applications. The only difference between the two experiments is in the first experiment students in both groups are not informed in the beginning that their brainstormed ideas are for programming applications. The purpose of is setting is to find the answer to research question Q1 of this study. All the students, after converging their ideas to potential applications and choosing one as the final application, must put pseudocodes for the application before translating it into C++ codes. Students in the experimental group answer a questionnaire on their usages of the mind mapping tool and the CSL system before the end of each experiment. The answers to the questionnaire are rated with 5-points Likert Scale.

3.2 The results

This study uses the number of ideas, created by each participant in the brainstorming stage of the experiments, as the basis for assessing on creativity in programming design. Three criteria: fluency, flexibility, and originality as defined in the methodology are used for the analysis of experimental data. The assessments of fluency is directly done by counting the numbers of categories of ideas integrated by each student on the CSL platform, while the assessments of flexibility and originality, between 1 to 10 points, are carried out by three programming experts recruited from the same engineering college as students'.

Fig. 8 exhibits the ANOVA test on the average number of ideas created by each participant in both experiments. As can be seen, participants in the experimental group, using the Mindmap divergent thinking tool, produce significantly more ideas than those in the control group do. What can also be noted is participants in both groups, knowing that the ideas created are for programming purpose in the second experiments, produce far less ideas than they do in the first experiment. This makes this study believe that the answer to the research question Q1 is positive.

Experiment	group	No. of Participants	Average number of ideas	STD	F	p-value
1	Experimental	26	130.269	38.691	17.418	**.000****
	Control	25	72.520	58.505		
2	Experimental	25	94.40	43.156	60.873	**.000****
	Control	26	25.27	13.162		
***p<0.05, **p<0.01**						

Fig.8. ANOVA on the average number of ideas created by each participant

The results of the ANOVA tests on the collected data from the first experiment for the three creativity scales are shown as in *Fig. 9*. As indicated, the experimental group performs significantly better than the control group in all the three scales.

The results of the ANOVA tests on the collected data from the second experiment for the three criteria are shown as in *Fig. 10*. Although the experimental group performs significantly better than the control group in all the three criteria, the constraint in Q1 seems to ferment in the second experiment.

From these experiment results, this study can confirm that students who use a divergent thinking tool for brainstorming will perform better than those who do not use any tool in programming design, i.e. a positive to the research question Q2.

The average score of the answers to the questionnaire is 3.984 out of 5-points Likert Scale, indicating most of the students approve that using mind mapping tool can boost their creativity in this study.

Creativity Scales	group	No. of Participants	Average	STD	F	p-value
fluency	Experimental	26	10.807	7.299	11.620	**0.001****
	Control	25	5.120	4.116		
flexibility	Experimental	25	5.705	0.667	29.876	**0.000****
	Control	26	2.533	1.162		
Originality	Experimental	25	6.859	0.957	63.279	**0.000****
	Control	26	4.466	1.182		
***p<0.05, **p<0.01*						

Fig.9. ANOVA tests on the three creativity scales for the first experiment

Creativity Scales	group	No. of Participants	Average	STD	F	p-value
fluency	Experimental	26	6.68	3.579	6.437	**0.014***
	Control	25	4.31	3.069		
flexibility	Experimental	25	3.743	1.744	5.137	**0.028***
	Control	26	2.666	1.668		
Originality	Experimental	25	7.421	0.809	7.562	**0.008****
	Control	26	6.893	0.540		
***p<0.05, **p<0.01*						

Fig.10. ANOVA tests on the three creativity scales for the second experiment

4 CONCLUSION

This study designs a methodology and builds a Computer Supported Learning online system with several creative thinking tools and a C++ programming environment. The methodology forces students to perform divergent creative thinking before designing their applications in the experiments. The experiments use the mind mapping tool in the system for divergent creative thinking. This study also modifies the Torrance Tests of Creative Thinking for evaluating the effectiveness of mind mapping tool in creative programming design. The creativity assessments are not only on the final applications generated, but also on the processes of creating the ideas and applications, making it more comprehensive and suitable for programming design. The results confirm that the usage of mind mapping tool in programming design has positive influence on stimulating learners' creative thinking and therefore helps them performing significantly better in creative programming designs.

On the other hand, this study focuses on boosting students' creativity on programming design, not on providing the assistance in writing pseudo codes and coding statements. In the experiments, some students may perplexed by the implementation of programs, that is he or she perhaps has excellent creativity in designing applications, but cannot translate them into actual codes owing to the limited programming ability. This situation is frustrating to students and, in the future, experiments should recruit students more experienced in actual programming.

REFERENCES

[1] Amabile, T. M. (1996a). Creativity in context: Update to" the social psychology of creativity.": Westview press.

[2] Sternberg, R. J., & Lubart, T. I. (1995). Defying the crowd: Cultivating creativity in a culture of conformity: Free Press, New York, 1995.

[3] Sternberg, R. J, & Lubart, T. I. (1999). The concept of creativity: Prospects and paradigms. In Handbook of creativity, Cambridge University Press, 1999, pp. 3-15.

[4] Cropley, D. H. (2015), Creativity in Engineering: Novel Solutions to Complex Problems, Academic Press, Cambridge, Massachusetts, 2015.

[5] Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2013). Computer Aided Instruction as a Vehicle for Problem Solving: Scratch Programming Environment in the Middle Years Classroom. Retrieved September Vol. 22.

[6] Mayer, R. E., & Wittrock, M. C. (2006). Problem solving. In Alexander, P., Winne, P., & Phye, G. (Eds.), Handbook of educational psychology, pp. 287-303, Mahwah. NJ: Erlbaum.

[7] Mayer, R. E, Dyck, J. L., & Vilberg, W., (1986). Learning to program and learning to think: what's the connection? Communications of the ACM, Vol. 29, No. 7, pp. 605-610.

[8] Torrance, E. (1972). Predictive Validity of the Torrance Tests of Creative Thinking*. The Journal of Creative Behavior, Vol.6, No. 4, pp. 236-262.

[9] Robbins, T. L., Kegley, K. (2010). Playing with Thinker toys to build creative abilities through online instruction. Thinking Skills and Creativity, Vol. 5, No. 1, pp. 40-48.

[10] Scott, S. G., Bruce, R. A. (1994). Determinants of innovative behavior: A path model of individual innovation in the workplace. Academy of management journal, Vol. 37, No. 3, pp. 580-607.

[11] Tierney, P., Farmer, S. M. (2002). Creative self-efficacy: Its potential antecedents and relationship to creative performance. Academy of Management Journal, Vol. 45, No. 6, pp. 1137-1148.