JOIV : Int. J. Inform. Visualization, 5(3) - September 2021 313-319

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Computational Thinking Evaluation Tool Development for Early Childhood Software Education

Kyunghee Lee[a], Jungwon Cho[b,*]

[a]Department of Innovation and Convergence, Hoseo University, Chungnam, 31499, South Korea
[b]Department of Computer Education, Jeju University, Jeju, 63243, South Korea
Corresponding author: [*]jwcho@jejunu.ac.kr

***Abstract*— The early childhood software education is being actively conducted, but research on evaluation of computational thinking is in its infancy. The purpose of early childhood software education is to cultivate the computational thinking through activities centered on solving problems in everyday life. Evaluation in software education is very important in that it not only measures computational thinking simply but also improves computational thinking through evaluation. As such, guidelines for evaluating computational thinking that can be used in early childhood software education are needed, but they are very lacking. Therefore, in this study, the researcher developed an evaluation tool that can meet the ultimate purpose of software education, cultivating computational thinking. The developed evaluation tools are a software education effectiveness test tool and a computational thinking test tool. They were developed to the level of development and interaction of the early childhood. The developed evaluation tool has been validated by software experts, early childhood education experts, and early childhood teachers. As a result of the second step validity verification, all content validity was confirmed. Through this, it was confirmed that the evaluation tool developed in this study can be used as a tool for evaluating computational thinking. This study provides implications for evaluation of computational thinking for early childhood software education. In addition, it is meaningful that it has been suggested to be effectively used for proper evaluation in early childhood software education.**

***Keywords*— Early childhood software education; software education; computational thinking; evaluation; education effectiveness.**

Manuscript received 8 Feb. 2021; revised 29 Mar. 2021; accepted 20 Apr. 2021. Date of publication 30 Sep. 2021.

I. INTRODUCTION

Education in the era of the 4th industrial revolution and artificial intelligence is rapidly changing in various directions as it faces a new phase. While the future society and the technological revolution of jobs are leading the change in human resources, Through education reform, lifelong learning, and re-education, individual competitiveness should be maintained and economic opportunities should be had [1]. Software education is very important in providing this opportunity for all.

Accordingly, software education has also come to search for a new medium and a new method of education. It is not an education that unilaterally transfers knowledge, but is becoming an education that nurtures the ability to think so that students can solve practical and meaningful problems. If it can use the contents that can cultivate computational thinking through the experience of solving problems in problem situations related to real life, it will be a more desirable software education. In order to cultivate computational thinking through software education, the 5th and 6th grades of elementary school, and middle school are conducting mandatory education. In recent years, it has even covered artificial intelligence education.

As such, software education is gradually expanding, and studies on teaching methods, learning methods, and effectiveness verification are actively being conducted, but studies on evaluation are relatively insufficient [2] – [5]. Since cultivation of computational thinking is the ultimate purpose of software education, although how to evaluate the improvement of computational thinking is a very important issue, the research on this is relatively incomplete [6] – [10]. In order for such education to be carried out effectively, not only the educational purpose, the content system, and the educational program, but also the subjects of education must be appropriately evaluated.

The evaluation method of early childhood software education introduced in previous studies conducted so far is focused on the evaluation of creativity, so its purpose is different. Creativity is the ability to produce new thoughts and is a characteristic of human intellectual ability [11]. It is said that creative thinking takes the form of tying distant thoughts together [12]. In other words, it is the result of a new combination of two or more thoughts. Creativity can be said to be the ability to derive useful results, including new thoughts and opinions, by modifying or changing existing thoughts or ideas. On the other hand, computational thinking is an approach method to solving problems in a way that can be implemented using a computer [13] – [16].

Computational thinking components can be broadly divided into Abstraction and Automation. Abstraction is a thought process to express real life problems in a form that can be solved. Collects and analyzes data necessary to solve various problems in everyday life, and presents them in an easy-to-read manner using necessary expression methods such as diagrams and graphs. After that, it is the process of decomposing complex elements into small units, extracting variables necessary for solution, and designing an appropriate solution model [17] – [19]. In other words, it can be defined as the ability to understand the computer's problem solving method and apply it to the problem solving process in real life.

As such, creativity and computational thinking are different areas, so it is desirable to evaluate them separately. In addition, when the more emphasizing the computational thinking, the more creativity is bound to improve naturally.

Therefore, early childhood software education should also be evaluated according to computational thinking, which is the ultimate purpose of software education. This study aims to develop a tool for evaluating early childhood computational thinking that is suitable and valid for infant.

II. Material and Method

The concept of evaluation is classified differently depending on where the emphasis is on education. First, when the focus is on confirming the achievement of educational goals, it can be defined as determining how much educational goals have been achieved through the curriculum and class activities[20]. Second, when the focus is on making a decision, it can be defined as the process of making decisions about class improvement and individual students through the collection and use of information [21]. Third, as Integration-compromise theory, when it is used for the purpose of making certain decisions about the process or outcome of education, it systematically investigates and utilizes the value and merits of the process and outcome of education [22]. Therefore, evaluation can be said to be a series of processes in which the degree of achievement of educational objectives is judged, the outcome of educational activities is measured, and the value of the results is judged.

A. Purpose of Evaluation

The purpose of the evaluation is to check the degree of reaching the educational goals and to improve the quality of learning and class. The content of the evaluation is the achievement standard, and not only the outcome of learning but also the process should be evaluated [23]. That is, education evaluation can confirm whether educational goals are set correctly, and whether plans and processes for education to realize the goals are appropriate. And ultimately, it can be said to be a series of processes to confirm and judge whether the educational goals have been properly achieved [24]. Early childhood evaluation aims to maximize individual development by evaluating the degree of development of infant, the effect of interactions with specific early childhood education programs and infant, and individual differences in each infant ability, characteristics and potential [25].

B. Method of Evaluation

In early childhood developmental research, it is applied in various ways depending on the purpose and method of evaluation, but the commonly used data collection methods include observation method, questionnaire method, test method, and interview method.

Observation is a method of collecting objective data by observing and recording the behavior of a study subject that appears naturally in everyday situations. Observations are made to find out the development of infant, however, all objects, scenes, and situations in infant and class, such as the interaction between teachers and infant, and educational environment, can be object of observation [26]. It is an evaluation method that can be usefully applied to research subjects whose expressions are inexperienced. Compared to other data collection methods, this method requires relatively less cooperation of the object of observation, while the observer requires a high degree of professional skills.

Questionnaire is a method of analyzing the contents by first writing a question related to the content to be researched, and having the subject answer it. However, it is difficult to use it for infants, who it is difficult to read the text, and there is a limitation that the collection ratio is low when it is delivered to the home.

The test method is a method of collecting data on the characteristics of a study subject using standardized tests. Since it is possible to predict, measure, and diagnose individual differences, not only can the developmental characteristics be understood through the results, but also relative comparisons with the same age group are possible, which has the advantage of obtaining objective and specific information. However, in order to apply to younger research subjects, it is necessary to carefully select and conduct tests with sufficient reliability. Interview is a method of collecting data by asking a question directly to a research subject and recording the responses. It is possible to observe the behavior at the interview and ask additional questions depending on the response. Since it has the advantage of collecting more detailed data, it is a data collection method frequently used in early childhood development studies. However, the results may vary depending on the interviewee's subjective interpretation or the interviewer's conditions, and because the process is one-to-one, there is a disadvantage that it takes a lot of time and effort.

In summary, this researcher proposes a Test method and a Questionnaire+Interview method as an evaluation method for early childhood software education. Fig 1 shows the complementary relationship between the questionnaire method and the interview method. By using the questionnaire method used in the empirical research method and the interview method used in the interpretive research method together, the strengths and weaknesses of each test method can be supplemented.

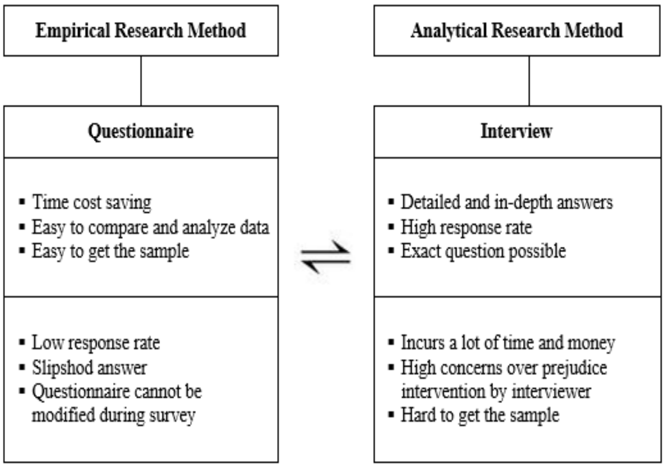

Fig. 1 The complementary relationship between the questionnaire method and the interview method

The purpose of early childhood software education is to cultivate computational thinking through activities centered on solving problems in everyday life. Evaluation in software education is very important in that it not only measures computational thinking, but also improves computational thinking through evaluation. Therefore, it is suggested that the evaluation of early childhood software education should be conducted in two forms: a software education effectiveness test and a computational thinking test.

III. Result and Discussion

A. Computational Thinking Effectiveness Test Tool Development

The evaluation targeting early childhood implies that a value judgment should be made on the overall process and outcome of early childhood education activities, rather than focusing only on learning achievement in early childhood. The evaluation method should be appropriate for the development and experience of early childhood, and the purpose and nature of the evaluation should be beneficial to early childhood [27].

Programming is not necessarily included in the evaluation of computational thinking in early childhood. Even without programming, if the problem is effectively solved using the concepts and principles of computer science, it can be included in computational thinking [28]. In a study on the development of a diagnostic tool for elementary and middle school software education capabilities by Jae-myung Yang, et al.(2017) it was considered that there was a need to expand it by constructing sub-items according to the problem situation and consisting of contents that can be experienced in real life [29]. In addition, it is necessary to consider the composition of the scenario as the subject of real life, the composition so that the problem can be solved with only thinking ability, not including the professional concept, the thinking ability of the problem solving process, the question considering the flow of thought, and the level of vocabulary suitable for school age.

The details and methods of the evaluation tools derived from this basis are as follows.

It is a test of the effectiveness of software education. Through this, it is possible to improve the overall educational activities and reset the direction by revealing whether the goal of software education has been achieved. In particular, it is possible to find basis in the evaluation of how much educational activity has reached the set goal, what is the factor if it is not achieved properly, and how it needs to be modified [30]. For this evaluation the questionnaire+interview method is the most suitable. Since it is one-to-one, it takes a lot of time and effort, but even the thoughts and actions of early childhood can be observed. The early childhood software education effectiveness test was based on an example of a question presented in the study of 'Development of evaluation factors for SW education in elementary and secondary schools' by Joo-Yeon Park et al. to examine the effects of education on early childhood [31]. It was developed by reorganizing the questionnaire+interview method by modifying and supplementing it according to the level of infant who is 5 years old and the purpose of the study. It was developed as a questionnaire method, but in consideration of early childhood inability to read texts, a reading interview method was also used.

The test sheet consists of 4 areas and 13 sub areas. It is divided into values, attitudes, computational thinking efficacy, and interest for the overall contents of education.

Value area consists of the following evaluation elements : The importance of SW, the usefulness of SW education, the importance of SW education, the convenience of SW, the need for SW class, the need for SW, the preference for the SW class, the satisfaction with the contents of the SW class, the expectation for the SW class, the perception of the SW related profession, and the awareness for SW related major, and the will for SW education. Attitude area consists of the following evaluation elements : Programming immersion, communication through class, interaction with friends, interaction with teachers, cooperation in class, creativity through cooperation with others, sense of community, and responsibility for programming ethics.

Computational thinking efficacy consists of the following evaluation elements : The field of computational thinking efficacy consists of the following evaluation elements: effort to solve a task, ability to express one's own thoughts, inquisitiveness for programming execution, active class participation, confidence in programming, confidence in information processing thinking process, and achievement through SW class. Interest area consists of the following evaluation elements : Interest in class, interest in learning activities, persistence in class, and interest in SW. The question was designed with 5 points that early childhood thinks it really is, 4 points that it is a little bit, 3 points that it is normal, 2 points that it is not, and 1 point that it is not at all.

The computational thinking effectiveness tool sample is shown in Fig 2.

Computational Thinking Effectiveness Test

You have learned about SW over the past two weeks with the teacher and friends. You learned commands that dictate the order and the ways of solving certain tasks, and found out what are needed for the processes. I'm going to ask you a few questions because I'm curious about your ideas on SW.
If you think it's really true, mark V on 5 and if it's not really true, mark V on 1.

Area	Detailed area	Questionnaire	Evaluation Element	Really True	True	Neutral	Not True	Not Really True
Value	Usefulness	1. I think SW is helpful on solving problems.	Importance and Usefulness of SW	⑤	④	③	②	①
	Importance	2. I think learning SW is important.	Importance of SW education	⑤	④	③	②	①
	Convenience	3. I think SW makes our lives easy.	Convenience of SW	⑤	④	③	②	①
	Necessity	4. I think SW is necessary for our lives.	Necessity of SW education	⑤	④	③	②	①
	Class satisfaction	5. I like learning SW.	Preference, Satisfaction and Anticipation SW	⑤	④	③	②	①
	Career search	6. I like to learn SW in elementary school.	Awareness of jobs related to SW, Awareness of majors related to SW, Willingness to learning SW	⑤	④	③	②	①
Attitude	Immersion	7. I can focus on learning SW.	Immersion in programming	⑤	④	③	②	①
	Communication	8. I can talk about SW more with my friends while learning.	Communication through classes, interaction with friends, interaction with teachers	⑤	④	③	②	①
	Cooperation (Sharing, Collaboration)	9. I can work with my friends more while learning SW.	Cooperation in classes, Expression of creativity through cooperation with others, Sense of community	⑤	④	③	②	①
	Accountability	10. I know I shouldn't say what others think as if it were mine.	Responsibility for programming ethics	⑤	④	③	②	①
Computational Thinking Efficacy	Self-directedness	11. I try hard to solve the problem while learning SW.	Effort to solve the task, Ability to express one's thoughts, Desire to explore the execution of programming	⑤	④	③	②	①
	Confidence	12. I can think of how to solve the problems in order while learning SW.	Active participation in classes, Confidence in programming & information processing, Sense of accomplishment	⑤	④	③	②	①
Interest	Learning interest	13. I'm looking forward to learning SW and it's fun.	Interest in SW, classes & learning activities, Willingness to continue learning SW	⑤	④	③	②	①

Fig. 2 Computational Thinking Effectiveness Tool Sample

B. Development of Computational Thinking Evaluation Tools

After introducing the computational curriculum in the UK in 2014, the evaluation of computational thinking began to be discussed in earnest. The evaluation of computational thinking is conducted in three types, using a specific test tool, a survey, and an observation method [32]. Brennan & Resnick presented three types of evaluation methods in the framework for computational thinking evaluation, an automated evaluation tool using the Scrape tool, an evaluation through interviews, and a method using an evaluation tool [33]. Seiter & Foreman proposed a method to evaluate computational thinking through scratch output in three aspects: Evidence Variables, Design Pattern Variable, and Computational Thinking Concepts [34]. In order to confirm the achievement of the ultimate purpose of software education, computational thinking must be evaluated. However, it is not easy to develop standardized items to measure this because of the abstractness of the thinking ability itself [35]. In addition, there is a need for evaluation by developing evaluation questions and methods according to the purpose and content of software education and education methods.

To find out the effect of improving computational thinking through software education, the computational thinking test was based on the questionnaire in 'Bebras Challenge – Castors : Years 4 & 5', in addition, taking into consideration the characteristics of early childhood in Korea and early childhood who are not familiar with letters, it was developed by reorganizing it into a picture-oriented test paper [36].

Each question is structured in a balanced way, including all areas of computational thinking, problem analysis, problem decomposition, abstraction, algorithm, automation, simulation, and computer science, so that the reliability of the test can be secured.

The computational thinking evaluation tool sample is shown in Fig 3.

Computational Thinking Evaluation Tools		
CT Elements		Computer Science Area
☐ Problem analysis ☐ Abstraction ☑ Automation	☐ Problem resolution ☑ Algorithm ☑ Simulation	☑ Algorithm and Programming ☐ Data analysis and data presentation ☑ Computing system composition and operation principle

6. Friends are playing robot games. The robot's name is SinB, but SinB only understands three commands: Go straight, Right, and Left.

If you order SinB to go straight, SinB will walk forward until hits a building, fence, or bush.
If you order SinB to the 'Right,' it changes direction to the right but does not move.
If you order SinB to the 'Left,' it changes direction to the left but does not move.

SinB is looking at the bush and is in the lower left corner. Please order and help SinB can go home.

1.
2.
3.
4.

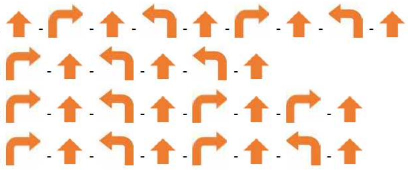

Fig. 3 Computational Thinking Evaluation Tool Sample

C. Test Tool Design

The early childhood software education evaluation tool developed in this study proceeded with Delphi in two steps to verify its validity. In the first stage, to verify the validity of the evaluation tool development, 10 university professors were composed of 5 software education experts and 5 early childhood education experts. In the second stage, 5 in-service early childhood teachers were composed to verify the validity of the field application. The study sequence is shown in Fig 4.

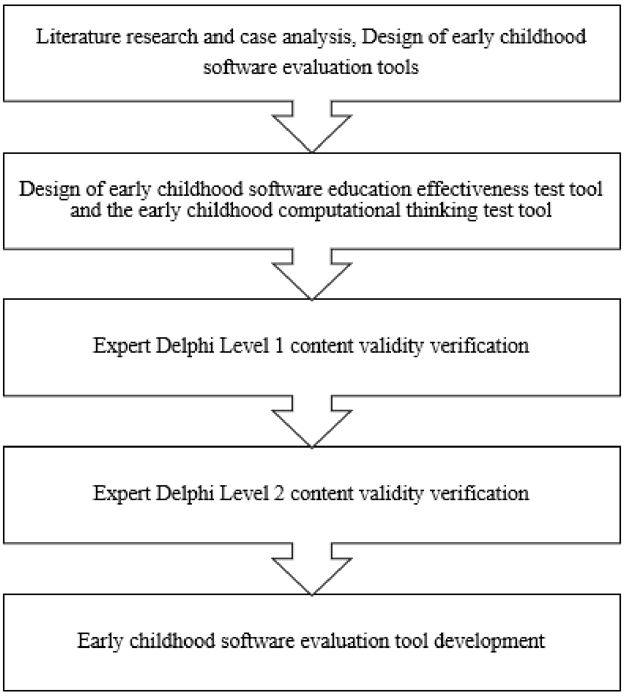

Fig. 4 Study Sequence

1) Level 1 Expert Delphi

To verify the validity of the developed evaluation tool, a validity test was conducted with 10 experts consisting of early childhood education professors and software education professors.

The content validity formula is estimated by (1).

$$CVR = \frac{n_e - \frac{N}{2}}{\frac{N}{2}} \qquad (1)$$

n_e is the number of cases answered as appropriate, and N is the total number of panels. The content validity ratio is the minimum value according to the number of panels, and it is judged that there is a content validity for the item when it reaches over the minimum value. In this study, 10 respondents answered the questionnaire, and the content validity ratio was based on CVR of 0.62 or higher.

Accordingly, the validity of the evaluation items was based on a CVR of 0.62 or higher. The validity of the contents was confirmed as the CVR values of all sub-areas were shown to be more than the standard 0.62. Table 1 shows the results of Delphi content validity for the software education effectiveness test tool.

TABLE I
SOFTWARE EDUCATION EFFECTIVENESS TEST TOOL DELPHI CONTENT VALIDITY

	Area	Detailed area	Average	SD	CVR
Q.1	Value	Usefulness	5.00	0.00	1.00
Q.2		Importance	4.90	0.30	1.00
Q.3		Convenience	4.80	0.40	1.00
Q.4		Necessity	4.40	0.66	0.80
Q.5		Class satisfaction	5.00	0.00	1.00
Q.6		Career search	5.00	0.00	1.00
Q.7	Attitude	Immersion	4.60	0.49	1.00
Q.8		Communication	4.30	0.46	1.00
Q.9		Cooperation (Sharing, Collaboration)	4.40	0.66	0.80
Q.10		Accountability	4.90	0.30	1.00
Q.11	Computational Thinking Efficacy	Self-directedness	4.90	0.30	1.00
Q.12		Confidence	4.70	0.64	0.80
Q.13	Interest	Learning interest	4.80	0.40	1.00

Table 2 shows the results of Delphi content validity for the software computational thinking test tool. The validity of the contents was confirmed as the CVR values of all the items appeared above the standard 0.62.

TABLE II
COMPUTATIONAL THINKING TEST TOOL DELPHI CONTENT VALIDITY

	CT Elements	Computer Science Area	Average	SD	CVR
Q.1	Automation, Algorithm	Algorithm and Programming	5.00	0.00	1.00
Q.2	Algorithm, Simulation	Algorithm and Programming	4.90	0.30	1.00
Q.3	Problem analysis, Problem resolution	Algorithm and Programming	4.70	0.46	1.00
Q.4	Abstraction, Automation, Simulation	Computing system composition and operation principle	4.70	0.46	1.00
Q.5	Problem analysis	Data analysis and data presentation	4.90	0.30	1.00
Q.6	Algorithm, Automation, Simulation	Algorithm and Programming Computing system composition and operation principle	4.90	0.30	1.00
Q.7	Problem analysis, Problem resolution	Data analysis and data presentation	4.70	0.64	0.80
Q.8	Algorithm	Algorithm and Programming	4.40	0.66	0.80
Q.9	Algorithm, Simulation	Algorithm and Programming Computing system composition and operation principle	4.90	0.30	0.80
Q.10	Abstraction	Data analysis and data presentation	4.70	0.46	1.00

2) Level 2 Field Application Aspect Delphi

In order to verify the validity of the evaluation tool in terms of field application, a validity test was conducted with five early childhood teachers.

Accordingly, the validity of the evaluation items was based on a CVR of 0.99 or higher. All CVR values were 0.99 or higher, confirming content validity. Table 3 shows the results of Delphi content validity for the software education effectiveness test tool.

TABLE III
SOFTWARE EDUCATION EFFECTIVENESS TEST TOOL DELPHI CONTENT VALIDITY (FIELD APPLICATION ASPECT)

	Area	Detailed area	Average	SD	CVR
Q.1	Value	Usefulness	5.0	0.00	1.00
Q.2		Importance	4.8	0.40	1.00
Q.3		Convenience	4.8	0.40	1.00
Q.4		Necessity	4.6	0.49	1.00
Q.5		Class satisfaction	5.0	0.00	1.00
Q.6		Career search	5.0	0.00	1.00
Q.7	Attitude	Immersion	4.6	0.49	1.00
Q.8		Communication	4.0	0.00	1.00
Q.9		Cooperation (Sharing, Collaboration)	4.2	0.40	1.00
Q.10		Accountability	4.8	0.40	1.00
Q.11	Computational Thinking Efficacy	Self-directedness	4.8	0.40	1.00
Q.12		Confidence	5.0	0.00	1.00
Q.13	Interest	Learning interest	4.8	0.40	1.00

Table 4 shows the results of Delphi content validity for the software computational thinking test tool. The validity of the contents was confirmed as the CVR values of all items were 0.99 or higher.

TABLE IV
COMPUTATIONAL THINKING TEST TOOL DELPHI CONTENT VALIDITY (FIELD APPLICATION ASPECT)

	CT Elements	Computer Science Area	Average	SD	CVR
Q.1	Automation, Algorithm	Algorithm and Programming	5.0	0.00	1.00
Q.2	Algorithm, Simulation	Algorithm and Programming	4.8	0.40	1.00
Q.3	Problem analysis, Problem resolution	Algorithm and Programming	4.8	0.40	1.00
Q.4	Abstraction, Automation, Simulation	Computing system composition and operation principle	4.8	0.40	1.00
Q.5	Problem analysis	Data analysis and data presentation	4.8	0.40	1.00
Q.6	Algorithm, Automation, Simulation	Algorithm and Programming, Computing system composition and operation principle	4.8	0.40	1.00
Q.7	Problem analysis, Problem Resolution	Data analysis and data presentation	4.4	0.80	1.00
Q.8	Algorithm	Algorithm and Programming	4.4	0.80	1.00
Q.9	Algorithm, Simulation	Algorithm and Programming, Computing system composition and operation principle	4.2	0.75	1.00
Q.10	Abstraction	Data analysis and data presentation	4.8	0.40	1.00

IV. CONCLUSIONS

Society is rapidly changing, and with the change, the structure of life is becoming diverse and complex. Infants will face a wider variety of problems to be solved in the future society. In this era, software education is bound to be a basic literacy that everyone must learn for their future life. It is not just a method of memorizing and acquiring knowledge, it should be an education that fosters computational thinking that can discover and approach various problems in daily life through algorithms and programming, think, and find and solve optimal methods. There is a need to be an education that can be approached naturally so that early childhoods do not perceive software education as education, and think of it as a play or experience while hanging out with their peers. Also, evaluation is absolutely necessary to check whether these classes are well-formed for the purpose.

Therefore, in this study, an early childhood software effectiveness test tool and a computational thinking ability test tool were developed suitable for the purpose and target. Content validity was verified through expert Delphi, and the conclusions were as follows.

First, the evaluation of early childhood software education should not be a creativity test, but a computational thinking evaluation that meets the ultimate purpose of software education.

Second, the evaluation method should be suitable for the developmental state and experience of early childhood, and the purpose and nature of the evaluation should be able to give benefits to early childhood.

Third, evaluation of early childhood software education should not only measure computational thinking, but also improve computational thinking through evaluation.

Finally, through the results of this study, the researcher hopes that interest and research will continue so that early childhood software education evaluation can be properly performed in accordance with the purpose of education in many countries interested in software education as well as Korea.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea.(NRF-2017S1A5A2A03067937). This article is extended and excerpted from the presented at Doctoral dissertation.

References

[1] G. Ashok, "Editorial: AI Education for the World," *AI magazine*, vol.38, pp.3, 2017.

[2] Latham, B. Poyade, M. Finlay, C. Edmond, A and McVey, M, "New Tools in Education: Development and Learning Effectiveness of a Computer Application for Use in a University Biology Curriculum," *Advances in experimental medicine and biology*, vol.1138, pp.29-46, 2019.

[3] Tsai, M. J, Wang C. Y and Hsu P. F, "Developing the Computer Programming Self-Efficacy Scale for Computer Literacy Education," *Journal Of Educational Computing Research*, vol.56, pp.1345-1360, 2019.

[4] Román-González, M., Moreno-León, J., and Robles, G. "Complementary tools for computational thinking assessment," *Proceedings of International Conference on Computational Thinking Education*, pp.154-159, 2017.

[5] David W, "Block-based Programming in Computer Science Education," *Communications of the ACM*, vol.62, pp.22-25, 2019.

[6] S. S. Kim, Y. J. Kim, A. R. Cho and M. W. Lee. "Development of a Tool for Computational Thinking Assessment in Problem-Solving Programming Education: Paper Type Inspection and Self-Report Questionnaire," *The Journal of Korean Association of Computer Education*, vol.22, pp. 89-99, 2019.

[7] Fonseca Ferreira N. M, and Freitas Elisabete D C. "Computer applications for education on industrial robotic systems," *Computer applications in engineering education*, vol.26 pp. 1186-1194, 2018.

[8] S. B. Sousa, Dischler, J. M, Adzhiev, V, Anderson, E. F, Ferko A, Fryazinov, O, Ilčík M, Ilčíková I, Slavik P and Sundstedt V, "Distinctive approaches to computer graphics education," *Computer Graphics Forum*, vol.37, pp. 403-412, 2018 .

[9] Zendler, A and Reile, S, "The effect of reciprocal teaching and programmed instruction on learning outcome in computer science education," *Studies In Educational Evaluation*, vol.58, pp. 132-144, 2018.

[10] Voas. J, Kuhn. R, Paulsen, C and Schaffer, K, "Computer Science Education in 2018," *IT professional*, : vol.20, pp.9-14, 2018.

[11] Guilford, J. P, "Three faces of intellect," *American Psychologist*, vol.14, pp.469-479. 1959.

[12] Mednick, S. "The associative basis of the creative process.," *Psychological Review*, vol.69, pp.220–232, 1962.

[13] CSTA. "K-12 Computer Science Standards." 2017.

[14] W. Xuefeng, Li. Lin, . Nanxi, T. Wei and K. S. Cheung, "The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy," *Computers & education*, vol.160, 2021.

[15] A. Charoula and G. Michail,"Computational thinking education: Issues and challenges," *Computers in human behavior*, vol.105, 2020.

[16] L, Joseph, A, and J, Magana, A, "Computational thinking in higher education: A review of the literature" *Computer applications in engineering education*, vol.28, pp.1174-1189, 2020.

[17] A. Charoula and V. Nicos, "Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy," *Computers in human behavior*, vol.105, 2020.

[18] C. E. Nadire, and B. Jeppe, "Computational Thinking in Compulsory Education: A Survey Study on Initiatives and Conceptions," *Educational technology research and development*, vol.68, pp. 551-573, 2020.

[19] Kong, S. C, Chiu M. M and L. Ming, "A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education," *Computers & education*, vol.127, pp.178-189, 2018.

[20] G. D. Dafna, K. Avi and F. Hanoch. "Theory-based assessment in environmental education: a tool for formative evaluation" *Environmental Education Research*, vol.23, pp.269-299, 2017.

[21] H. Sheng-Jen, "Development and Evaluation of Remote Virtual Teach Pendant for Industrial Robotics Education," *International Journal of Engineering Education*, vol.35, pp.1816-1826, 2019.

[22] A. Sue and D. Richard, "The informed study project: An innovative online self evaluation of fitness to practise within a professional higher education degree programme," *Innovations in Education and Teaching International*, vol.56, pp.529-541, 2019.

[23] F. Sarah, B. R. Katersky and K. Sue, "Equipping parents to support their children's higher education aspirations: a design and evaluation tool," *Educational review*, vol.71, pp. 198-217, 2019.

[24] A. Angelica, S. Donatella and P. L. Elvira, "The evaluation of behavioural problems in the first three years of life: Comparing parents and early childhood educators," *Infant behavior & development*, vol.50, pp.324-327, 2018.

[25] Fons. N. Elina and S. Ulla. "Sustainable Pedagogical Leadership in Finnish Early Childhood Education (ECE): An Evaluation by ECE Professionals," *Early childhood education journal*, vol.48, pp.213-222, 2020.

[26] M. D. Amy, H. Carmen, S. Shukla and D. Lena, "Effective Early Childhood STEM Education: Findings from the "Little Scientists" Evaluation," *Early childhood education journal*, vol.48, pp. 353-363, 2020.

[27] K. H. Lee, E. H. Koh, C. U. Hong, Y. S. Lee, E. K. Moon and J. W. Cho. "New Curriculum of Software Education for 5-Years-Old Children in Korea, *Jour of Adv Research in Dynamical & Control Systems*, Vol. 10, 14-Special Issue, 2018.

[28] N. Jiyae and J. M. Lee, "Effects of Robotics Programming on the Computational Thinking and Creativity of Elementary School Students," *Educational technology research and development*, vol.68, pp.463-484, 2020.

[29] J. M. Yang, J. M. Kim, W. K. Lee, I, K, Lee and J. H. Seo. *A Study on the Development of Software Training Competency Diagnostic Tool*, Daegu: KERIS, 2017.

[30] K. H. Lee, E. H. Koh, C. U. Hong, Y. S. Lee, E. K. Moon and J. W. Cho. "Suggestions for Evaluation Tool of Evaluating Educational Effectiveness of Early Childhood Software education," *Proceedings of the Korean Association of Computer Education Conference*, vol.23, pp.77-79, 2019.

[31] J. Y. Park, J. H. Kim, S. H. Kim, H. S. Lee and S. H. Kim. "Development of evaluation factors for SW education in elementary and secondary schools," *The Journal of Korean Association of Computer Education*, vol.20, pp.47-59, 2017.

[32] M. J. Kim, W. K. Lee and J. M. Kim. "Presenting the Development Direction Through the Analysis of Tool used to Measure Computational Thinking," *The Journal of Korean Association of Computer Education*, vol.20, pp.17-25, 2017.

[33] Brennan, K. and Resnick, M. *New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada, 1, pp.25, 2012.

[34] CACM Staff, "Computational Thinking Is Not Necessarily Computational," *Communications of the ACM*, vol.60, pp.8-9, 2017.

[35] Sung W, Ahn J and Black, J. B, "Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education," *Technology, Knowledge and Learning*, vol.22, pp. 443-463, 2017.

[36] Bebras Challenge 2018. The Bebras Challenge website. [Online]. http://bebras.or