J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

Examining the relationship between educational effectiveness and computational thinking in smart learning environment☆

Oakyoung Han[1] Jaehyoun Kim[2*]

ABSTRACT

The 4th industrial revolution has brought innovation in the educational environment. The purpose of this study is to verify the educational effectiveness of smart learning environment especially with the computational thinking. A big data analysis was performed to confirm that computational thinking is the one to prepare the 4th industrial revolution. To teach computational thinking at university, educational design should be careful. This study verified the relationship between improvement of computational thinking ability and major of students with coding education. There was difference in effectiveness of the coding education depending on the major of students, it means students must be guaranteed to be educated by the differentiated coding education for different major. This study extracted factors of computational thinking through literature review. Thirteen research hypotheses were applied for the statistical analysis in R language. It was proved that expectation of class and improvement of abstraction ability and algorithmic thinking ability had mediation effect to the relationship between knowledge acquisition and problem-solving abilities. Based on this study, effectiveness of education can be improved, and it will lead to produce a lot of distinguished students who are ready for the 4th industrial revolution.

☞ keyword : Smart Learning Environment, Computational Thinking, Educational Effectiveness, Factor Analysis

1. Introduction

Due to the significant development of information science and technology, smart learning environment has been established. There are continuous trial for paradigm shift in education to apply the smart learning environment[1]. Nevertheless, most students in Korea still get text-based lectures in a traditional way[2]. Unfortunately, whether we want it or not, the wave of the 4th industrial revolution is coming to us at a rapid pace. In order to prepare for the era of the 4th industrial revolution, education should be reformed[3].

Alvin Toffler said "Students in Korea are wasting time in schools and academies for almost 15 hours a day for unnecessary knowledge and for jobs that will not exist in the future"[4]. It is time to change method of education. Diversified and personalized education along with applying utilization of internet and communication services could be important approaches for the competitive future. This approach can be indicated as the smart learning environment. However, applying the smart learning environment for every education field could be an excessive approach. Since we need to verify whether the smart learning environment can improve the effectiveness of education since giving better opportunities to students and reflecting effectiveness to students are different matter.

It is obvious that we need the ability to understand machines and work with machines in the future. The language that can control machines is a programming language, and the way to communicate with the machine is the software. The requested qualification for the future

1 Sungkyun Software Education Institute, Sungkyunkwan University, Seoul, 03063, Korea
2 Dept. of Computer Education, Sungkyunkwan University, Seoul, 03063, Korea
* Corresponding author (jaekim@skku.edu)
[Received 25 October 2017, Reviewed 7 November 2017(R2 15 December 2017), Accepted 3 January 2018]
☆ "This research was supported by the MIST(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute for Information & communications Technology Promotion)"(2015-0-00914)
☆ A preliminary version of this paper was presented at APIC-IST 2017 and was selected as an outstanding paper.

http://dx.doi.org/10.7472/jksii.2018.19.2.57

workers is understanding software. In order to collaborate with software, we must have computational thinking. Learning computational thinking boom is happening nationwide in Korea. However, there is no accurate definition of the computational thinking.

In this paper, we study whether providing a smart learning environment actually improves the efficiency of education, and analyze the factors of computational thinking based on the survey results. The result of this study may be useful to improve the educational effectiveness with smart learning environment, and to prepare for the education of the 4th industrial revolution based on the effect of the factors analyzed for the computatioal thinking. However, you have to aware there are restrictions on applying it as a general case since the study of this paper is an analysis in a limited environment.

2. Related Work

2.1 Smart Learning Environment

Smart learning is self-directed, resource-enriched, motivated, adaptive, and technology-embedded learning method. These characteristics imply that smart learning extends educational time, methods, capacity, contents, and spaces. These extensions lead to a smart learning environment. A smart learning environment aims to provide personalized and customized learning services to learners, and its goal is to support easy, engaged and effective learning for learners in any possible smart ways[5].

The characteristics of smart learning are encompassed as following: intelligent personalized study, cooperative activity, bi-direction, participation activity, sharing activity, intelligent study information management, time/space limitation conquest, study information generation, application of social networking, application of convergent education media, and non-linear study[6].

To enhance theses characteristics, a success strategy is needed. As the strengths of the smart learning, educational environment based system along with propulsion capacity, a teacher's sense of duty, excellent support for information and communications infrastructure, and diversity of educational curriculum were reviewed[7]. It is important to have a strategy to success in smart learning; hence, it can bring the better educational environment for students who are the hopes of our future.

The purpose of providing smart learning environment is to implant problem-solving ability to students. Various researches are being conducted to maximize the improvement of problem-solving ability and to apply high technology to the educational field for better educational effectiveness[8].

2.2 Computational Thinking

Computational thinking is an approach to solving problems, building systems, and understanding human behavior that draws on the power and limits of computing. Computational thinking is the use of abstraction to tackle complexity and the use of automation to tackle scale. The combination of the automation of abstraction underlies the enormous capability and reach of computing[9].

The factors of computational thinking from the related work are summarized in Table 1. The concepts of each factor are as follows: The reformulating factor is to do formulating problems in a way that enables us to use a computer and other tools to help solve the problems. The recursive factor is to solve problems repeatedly. The 'data as code' factor is to treat real-life data as code. The simplifying factor is to turn complex procedures into simple steps. The abstraction factor is to extract key elements. The decomposition factor is to break down into smaller problems for manageability. The pre-fetching factor is to prepare the necessary elements in advance for efficient problem solving. The 'resource sharing' factor is to use limited resources efficiently. The 'heuristic reasoning' factor is to discover a solution with informal, intuitive, and speculative procedures. The 'algorithmic and procedures' factor is to use a series of ordered steps to solve a problem. The 'data collection' factor is to collect input data for the solution of the problem. The 'data analysis' factor is to analyze input data with proper processing. The 'data representation' factor is to represent the result of data processing as an output. The simulation factor is to reproduce the behavior of a system. The automation factor is to compute the abstraction of the problem. The generalizing factor is to infer a general principle from particular facts, statistics, or results. The parallel factor is to relate with other components for a whole problem set. The

〈Table 1〉 Factors of Computational Thinking

FACTORS	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]
Reformulating	●			●		●	●		●		●	●		●
Recursive			●									●		
Data as code												●		
Simplifying												●		
Abstraction	●	●	●	●	●	●	●		●	●	●	●	●	●
Decomposition		●	●							●		●	●	
Pre-fetching												●		
Resource Sharing												●		
Heuristic Reasoning												●	●	●
Algorithmic & Procedures	●	●	●	●		●	●	●	●	●	●	●	●	●
Data Collection		●							●			●		
Data Analysis	●	●		●	●	●	●		●		●	●		
Data Representation	●	●	●	●		●	●	●		●		●		●
Simulation	●	●			●				●		●	●		
Automation	●	●		●	●	●	●		●		●	●	●	●
Generalizing	●					●					●			
Parallel		●								●				
Pattern Generalization			●	●			●		●					
Conditional Logic			●											
Error Detection			●											

'pattern generalization' is to generalize patterns by understanding patterns in problems on an abstract level. The 'conditional logic' factor is to apply logical thinking to specific conditions. The 'error detection' factor is to find potential errors for avoiding erroneous results.

3. Research Method

This section consists of five different studies. First, we review the improvement to the effectiveness of education in the smart learning environment. Second, a big data analysis for the 4th industrial revolution is studied to find the connect point with education. Third, we measure the difference in major for the improvement of computational thinking. Fourth, we perform factor analysis for computational thinking. Fifth, analysis of mediation effects between knowledge acquisition and problem-solving ability for 3 different factors is studied.

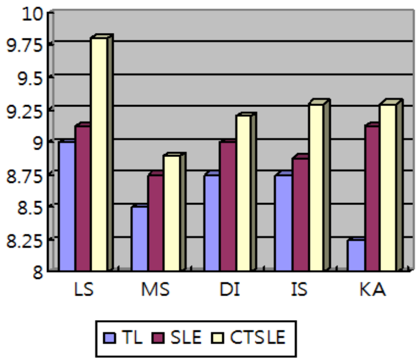
3.1 Effectiveness of Education

To measure the educational satisfaction, we traced the algorithm class for three years. In the first year, the algorithm class was held in a traditional learning environment. The class was instructor-centered, and the students learned the basic concepts of the algorithm along with only famous problems. C programming language was applied without any exception.

In the second year, a smart learning environment was selected for the algorithm class. In addition to the traditional learning environment, students are allowed to use any smart device for learning, and they implemented programs for smart devices using any language they wanted. SNS services were used to communicate with the instructor.

In the last year of the study, the enhanced smart learning environment was selected for the algorithm class. To enhance the smart learning environment, computational thinking approach was applied. The instructor taught only basic concepts of the theory, and the students had to create a proper problem for each strategy for algorithm. The students were required to simplify the problems, to draw the abstraction, to apply heuristic reasoning, to think recursively, to reformulate a new problem into a familiar problem, to program for automation, to parallelize their works, to implement a program in two different languages, to design a solution with procedural steps, and to cache some important concepts for the future.

Even though the last environment was the most difficult, the satisfaction level was the highest. Figure 1 shows the result of the satisfaction of students. In Figure 1, LS stands for Learning Satisfaction, MS stands for Material Satisfaction, DI stands for Degree of Interest, IS stands for Interaction Satisfaction, and KA stands for Knowledge Acquisition. TL was the method for the first year of the study, SLE is the smart learning environment which was the method for the second year of the study, and CTSLE is the computational thinking smart learning environment which was the method for the last year of the study.



(Figure 1) Satisfaction of Students

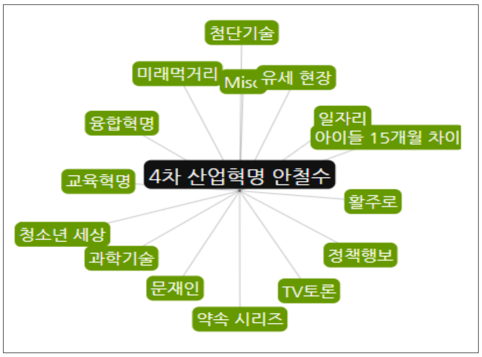
As the satisfaction level had been improved, the effectiveness of education was getting better. It meant that, even if the amount of learning increased, the learner 's experience was widened and the learning effect was improved when various learning environments were supported.

3.2 Big data Analysis

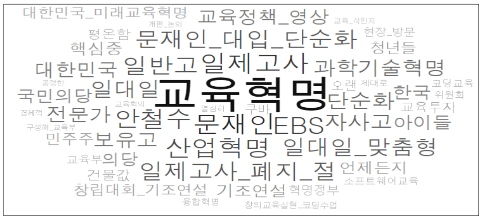
To check whether the 4th industrial revolution and education were related, a big data analysis was performed. The result of it is shown in Figure 2.

As a result of a big data analysis applying '4th industrial revolution' in Korean as a keyword, 'educational revolution' listed as a topic word. The duration for the big data analysis was from January 1, 2016 to September 30, 2017.

Since the word 'education revolution' was one of topic words, we went through another big data analysis for it, and the result is shown is Figure 3.



(Figure 2) Big data Analysis: Topic Words



(Figure 3) Big data Analysis : Word Cloud

When a big data analysis had been executed with 'educational revolution' in Korean as a keyword, it extracted 'coding training', 'coding class' and 'software education' as related words. The duration for the big data analysis was from January 1, 2016 to September 30, 2017.

Figure 4 showed the big data analysis of software education for topic word. As a result of a big data analysis applying 'software education' in Korean as a keyword, 'computational thinking' listed as a topic word. The duration for the big data analysis was from January 1, 2016 to September 30, 2017.

(Figure 4) Big data Analysis with SW Education

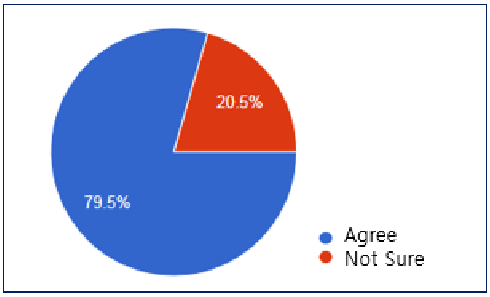
3.3 Difference in Major

We analyzed computational thinking through questionnaire. Since it is proved that the computational thinking is helping students to improve educational effectiveness, it is worthy to analyze factors in computational thinking. The research hypothesis for the survey is as follows.

- RH01: When computational thinking is improved, problem-solving ability is improved.
- RH02: The level of improvement on computational thinking is identical regardless of major.
- RH03: Reformulating thinking is an independent factor of computational thinking.
- RH04: Recursive thinking is an independent factor of computational thinking.
- RH05: Concept for 'data as code and code as data' is an independent factor of computational thinking.
- RH06: Simplicity is an independent factor of computational thinking.
- RH07: Abstraction is an independent factor of computational thinking.
- RH08: Decomposition of a problem is an independent factor of computational thinking.
- RH09: Prefetching and caching is an independent factor of computational thinking.
- RH10: Resource sharing is an independent factor of computational thinking.
- RH11: Heuristic Reasoning is an independent factor of computational thinking.
- RH12: Algorithmic thinking is an independent factor of computational thinking.
- RH13: Data collection is an independent factor of computational thinking.
- RH14: Data analysis is an independent factor of computational thinking.
- RH15: Data representation is an independent factor of computational thinking.

There was total of 303 answers for the survey. The students who took the questionnaires were the students of the 'problem solving and algorithm' class. To verify the questionnaire and to calculate statistics, R programming language had been used.

According to numerical statistics, almost 80% of the respondents agreed the need for education of software. Figure 5 is showing the result.



(Figure 5) Need for Education of Software

We used var.test function in R to compare the variances of two samples from normal populations. Two samples are improved problem-solving ability and improved computational thinking in coding education.

As the result of the var.test, the p-value is 0.7771, which is greater than significance level .05. Hence, the dispersion of the two groups is the same, and Figure 6 is showing it. Since has been proved that variances are same for improved problem-solving ability and improved computational thinking, research hypothesis RH1 is accepted.

```
> var.test(ImprovePS ~ ImproveCT, data=MyData)

        F test to compare two variances

data:  ImprovePS by ImproveCT
F = 0.90965, num df = 32, denom df = 269, p-value = 0.7771
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.5677549 1.6253008
sample estimates:
ratio of variances
         0.9096486
```

(Figure 6) Uniform Distribution

For deeper study of RH1, we checked the mean of the level for computational thinking improvement for respondents. Figure 7 is showing that the mean values are not same for students who answered the improvement of computational thinking differently. Those of who answered that their computational thinking was not improved showed that they only improved 1.92 for the improvement of problem-solving ability while students who answered that their computational thinking was improved showed 3.47 as the mean value for the problem-solving ability.

```
> t.test(ImprovePS ~ ImproveCT, data=MyData)

	Welch Two Sample t-test

data:  ImprovePS by ImproveCT
t = -9.797, df = 41.089, p-value = 2.609e-12
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.864824 -1.227431
sample estimates:
 mean in group NO mean in group YES
         1.924242          3.470370
```

(Figure 7) t.test for mean values

To measure the difference in major, it had been analyzed the relationship between improvement of computational thinking ability and major of students with coding education. In other words, regardless of major, it has been assumed that the coding education showed improvement of the same computational thinking capability to all students. The importance of this research is to verify the effectiveness of the coding education regardless of major. If there is difference in effectiveness of the coding education depending on the major of students, it means students must be guaranteed to be educated by the differentiated coding education for different major.

The probabilities of the null hypothesis that the major and the improvement of computational thinking are independent are less than .001 as in Figure 8. Hence, the null hypothesis is rejected. There is a relationship between major of students and the improvement computational thinking. It concludes the research hypothesis RH2 is rejected.

```
> M = xtabs(~ major + ImproveCT, data=MyData)
> M
                                ImproveCT
major                             NO YES
  Economics/Business Administration 12 134
  Education                       16 110
  Social Science                   5  26
>
> ht.out1 = chisq.test(M)
> ht.out1

	Pearson's Chi-squared test

data:  M
X-squared = 306.38, df = 6, p-value < 2.2e-16
```

(Figure 8) Test of Independence

To verify the difference in the improvement of computational thinking, the exact figures were confirmed in Figure 9. The students majored in economics and business administration reported 92% of the improvement for computational thinking while the students majored in education reported 87% of the improvement, and the students majored in social science reported 84% of the improvement of computational thinking. It means that they have to be taught software education differently depends on their major.

```
> prop.table(ht.out1$observed,1)
                                   ImproveCT
major                                       NO        YES
  Economics/Business Administration 0.08219178 0.91780822
  Education                         0.12698413 0.87301587
  Social Science                    0.16129032 0.83870968
```

(Figure 9) Exact Figures of Difference in Major

3.4 Factor Analysis

Based on Table 1 in Section 2, we extracted the basic 13 sub-elements for computational thinking, and Table 2 is the list of the sub-elements.

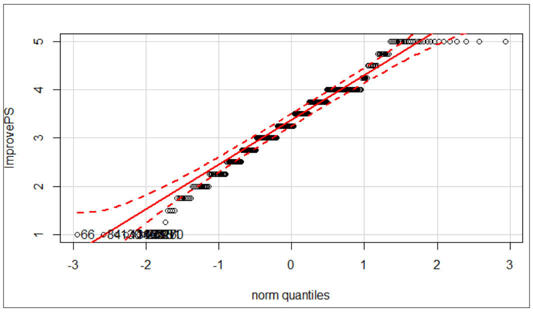
The research was carried out through the questionnaire analysis reflecting the elements of Table 2, and the research hypothesis RH03 thru RH15 has been made based on it. For the analysis, the answer that interfered with the regularity of the statistical data was examined first.

(Table 2) Sub-elements of Computation Thinking

Code	Stand for
REF	Reformulating Thinking
REC	Recursive Thinking
DAC	Data as Code, Code as Data
SMP	Simplicity
ABT	Abstraction
DCP	Decomposition
PAC	Prefetching and Caching
RSS	Resource Sharing
HRR	Heuristic Reasoning
AGM	Algorithmic Thinking
DCL	Data Collection
DAL	Data Analysis
DPR	Data Representation

As shown in Figures 10, eleven respondents appeared to observe regularity. In this study, we did not remove the 11 responses to normality and reflected the all answers to the results.

```
> qqPlot(ImprovePS, labels=row.names(MyData), id.n=11)
 66  84 130 134 160 195 209 218 226 231 270
  1   2   3   4   5   6   7   8   9  10  11
```



(Figure 10) Shapiro-Wilk normality test

The questionnaire was designed for 13 factors but still it is necessary to check how many factors existed in the questionnaire. Since the size of sample was 303, the minimum factor loading value must be higher than 0.325. We removed some of the questions by applying the minimum factor loading value, and reduced the number of factors with factor correlation values greater than or equal to 7. The final factor analysis is as shown in Figure 11.

```
Loadings:
    Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8 Factor9 Factor10
Q37  0.527                                   0.204                                  
Q38  0.533                                   0.315                                  
Q39  0.814                                                                   0.275  
Q40  0.973                          -0.209                                          
Q41  1.027                                                                          
Q42  0.965                                                                          
Q43  0.717                                                                          
Q44  0.756                                                                          
Q45  0.616   0.281                                                                  
Q13          0.631   0.257                                                          
Q14          0.924                                                                  
Q15          0.671                                                                  
Q16          0.718                                                                  
Q17          0.636                                                                  
Q18          0.697                                                                  
Q19          0.687                                                                  
Q8                   0.895                                                          
Q9                   0.987                                                          
Q10                  0.682                                                          
Q11                  0.679                                           0.207          
Q26                          0.615                                                  
Q27                          0.555                                                  
Q28                          0.736                                                  
Q24                                  0.719                                          
Q35  0.230                                   0.707                                  
Q20                                                  0.708                          
Q7                   0.435           0.208                                          
Q12                  0.440                                           0.414          
Q21  0.278                                           0.323                          
Q22          0.442           0.223                                                  
Q29                          0.366                                                  
Q32  0.350                                                   0.400                  
Q33  0.466                           0.215                   0.255                  
Q34  0.450                                                   0.363                  
Q36  0.322                                   0.201                                  
```

```
               Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8 Factor9
SS loadings      6.606   4.191   3.392   1.618   0.951   0.814   0.698   0.507   0.342
Proportion Var   0.189   0.120   0.097   0.046   0.027   0.023   0.020   0.014   0.010
Cumulative Var   0.189   0.308   0.405   0.452   0.479   0.502   0.522   0.536   0.546
               Factor10
SS loadings       0.317
Proportion Var    0.009
Cumulative Var    0.555
```

```
Factor Correlations:
         Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8 Factor9 Factor10
Factor1    1.000   0.638   0.767   0.763   0.767  -0.697   0.761  -0.391  0.3878   0.1884
Factor2    0.638   1.000   0.637   0.600   0.608  -0.592   0.602  -0.384  0.3149   0.2928
Factor3    0.767   0.637   1.000   0.674   0.786  -0.814   0.782  -0.470  0.3088   0.3224
Factor4    0.763   0.600   0.674   1.000   0.730  -0.641   0.692  -0.412  0.4442   0.1821
Factor5    0.767   0.608   0.786   0.730   1.000  -0.745   0.774  -0.472  0.3079   0.2595
Factor6   -0.697  -0.592  -0.814  -0.641  -0.745   1.000  -0.773   0.462 -0.3238  -0.2546
Factor7    0.761   0.602   0.782   0.692   0.774  -0.773   1.000  -0.464  0.3100   0.1809
Factor8   -0.391  -0.384  -0.470  -0.412  -0.472   0.462  -0.464   1.000 -0.0220  -0.2187
Factor9    0.388   0.315   0.309   0.444   0.308  -0.324   0.310  -0.022  1.0000   0.0454
Factor10   0.188   0.293   0.322   0.182   0.260  -0.255   0.181  -0.219  0.0454   1.0000

Test of the hypothesis that 10 factors are sufficient.
The chi square statistic is 571.96 on 290 degrees of freedom.
The p-value is 1.17e-20
```

(Figure 11) Factor Analysis

As the result of the factor analysis, the research hypothesis needs to be changed. Among 13 research hypothesis related to factors of computational thinking, only RH07 and RH12 were accepted. Nevertheless, other 11 research hypothesis did not need to be rejected since factors were combined with other factors and they came up as new factors for computational thinking. There were 8 newly combined factors for the computational thinking.

With the result of the factor analysis, we checked the interaction of the improvement of computational thinking with each of 10 final factors. However, only 6 factors showed the interactive relationship with computational thinking based on the improvement of problem-solving ability. In other words, if a student showed the improvement of problem-solving ability, that student also improved computational thinking along with the interaction with 6 different factors. When we analyze the interaction, the significance level must be smaller than 0.05. The results of the interaction analysis are following.

```
> rpt1 = lm(ImprovePS ~ factor(ImproveCT) * factor(RFT), data=MyData)
> anova(rpt1)
Analysis of Variance Table

Response: ImprovePS
                              Df  Sum Sq Mean Sq  F value    Pr(>F)    
factor(ImproveCT)              1  70.295  70.295 177.6968 < 2.2e-16 ***
factor(RFT)                   12 116.072   9.673  24.4511 < 2.2e-16 ***
factor(ImproveCT):factor(RFT)  6   9.425   1.571   3.9707  0.000785 ***
Residuals                    283 111.952   0.396                       
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> rpt2 = lm(ImprovePS ~ factor(ImproveCT) * factor(RCT), data=MyData)
> anova(rpt2)
Analysis of Variance Table

Response: ImprovePS
                              Df  Sum Sq Mean Sq  F value    Pr(>F)    
factor(ImproveCT)              1  70.295  70.295 155.8062 < 2.2e-16 ***
factor(RCT)                   12  96.681   8.057  17.8574 < 2.2e-16 ***
factor(ImproveCT):factor(RCT)  7  13.537   1.934   4.2864 0.0001598 ***
Residuals                    282 127.230   0.451                       
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> rpt3 = lm(ImprovePS ~ factor(ImproveCT) * factor(ABT), data= MyData)
> anova(rpt3)
Analysis of Variance Table

Response: ImprovePS
                              Df  Sum Sq Mean Sq  F value    Pr(>F)    
factor(ImproveCT)              1  70.295  70.295 137.7448 < 2.2e-16 ***
factor(ABT)                    4  79.151  19.788  38.7746 < 2.2e-16 ***
factor(ImproveCT):factor(ABT)  4   8.771   2.193   4.2966  0.002154 ** 
Residuals                    293 149.527   0.510                       
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> rpt4 = lm(ImprovePS ~ factor(ImproveCT) * factor(SMC), data=MyData)
> anova(rpt4)
Analysis of Variance Table

Response: ImprovePS
                              Df  Sum Sq Mean Sq  F value  Pr(>F)    
factor(ImproveCT)              1  70.295  70.295 161.4827 < 2e-16 ***
factor(SMC)                   31 115.318   3.720   8.5454 < 2e-16 ***
factor(ImproveCT):factor(SMC)  9   8.515   0.946   2.1733 0.02426 *  
Residuals                    261 113.616   0.435                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> rpt5 = lm(ImprovePS ~factor(ImproveCT) * factor(RRF), data=MyData)
> anova(rpt5)
Analysis of Variance Table

Response: ImprovePS
                              Df  Sum Sq Mean Sq  F value    Pr(>F)    
factor(ImproveCT)              1  70.295  70.295 181.5726 < 2.2e-16 ***
factor(RRF)                   23 118.704   5.161  13.3310 < 2.2e-16 ***
factor(ImproveCT):factor(RRF) 10  14.989   1.499   3.8718 6.388e-05 ***
Residuals                    268 103.755   0.387                       
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> rpt6 = lm(ImprovePS ~ factor(ImproveCT) * factor(AGM), data= MyData)
> anova(rpt6)
Analysis of Variance Table

Response: ImprovePS
                              Df  Sum Sq Mean Sq  F value    Pr(>F)    
factor(ImproveCT)              1  70.295  70.295 142.8492 < 2.2e-16 ***
factor(AGM)                    4  82.810  20.703  42.0705 < 2.2e-16 ***
factor(ImproveCT):factor(AGM)  4  10.455   2.614   5.3113 0.0003836 ***
Residuals                    293 144.184   0.492                       
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(Figure 12) Analysis of Interaction

Figure 12 showed that 6 factors such as RFT, RCT, ABT, SMC, RRF, and AGM were interacted with the improvement of computational thinking. Table 3 summarized the revised factors of the computational thinking.

(Table 3) Revised Factors of Computational Thinking

Original Factor	Applied as
Reformulating Thinking	RFT, RRF
Recursive Thinking	RCT
Data as Code, Code as Data	SMC
Simplicity	SMC
Abstraction	As it is (ABT)
Decomposition	RCT
Prefetching and Caching	RRF
Resource Sharing	RFT
Heuristic Reasoning	RRF
Algorithmic Thinking	As it is (AGM)
Data Collection	SMC
Data Analysis	SMC
Data Representation	SMC

3.5 Analysis of Mediation Effect

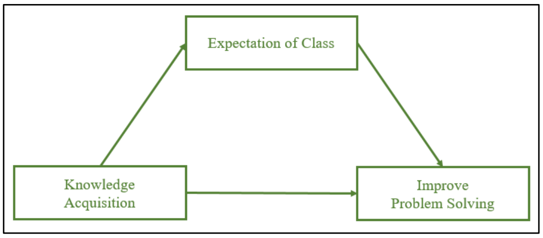
Three mediation effect analysis were conducted to test the relationship between the knowledge acquisition and the improvement of problem solving ability with following factors: expectation of class, improvement of abstraction ability, and improvement of algorithmic thinking ability. The assumption of the analysis is that the higher the knowledge acquisition, the better the computational thinking. To prove the assumption, a function of linear model has been applied

```
> YN=factor(MyData$ImproveCT)
> KN=factor(MyData$KnowledgeAcq)
> RA=data.frame(YN,KN,MyData)
> rst1=lm(ImprovePS ~ YN + KN, data=MyData)
> summary(rst1)

Call:
lm(formula = ImprovePS ~ YN + KN, data = MyData)

Residuals:
     Min       1Q   Median       3Q      Max
-2.36500 -0.25502  0.09041  0.34041  1.49498

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.25502    0.13256   9.467  < 2e-16 ***
YNYES        0.05491    0.13678   0.401    0.688
KN2          0.98806    0.16743   5.902 9.81e-09 ***
KN3          1.63797    0.17870   9.166  < 2e-16 ***
KN4          2.34966    0.17896  13.130  < 2e-16 ***
KN5          3.30507    0.18830  17.552  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5374 on 297 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.7212,	Adjusted R-squared:  0.7166
F-statistic: 153.7 on 5 and 297 DF,  p-value: < 2.2e-16
```

(Figure 13) Improve CT and Knowledge Acquisition

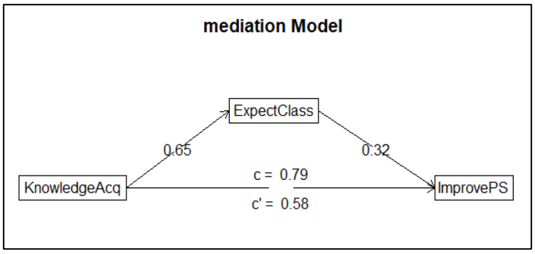
as Figure 13. Since we already proved that the problem solving ability is improved when computational thinking is improved by the research hypothesis RH01, anything that can improve the problem solving ability can also improve the computational thinking.

Figure 14 is a model to analyze the mediation effect. Statistical analysis in R showed that there is mediation effect in the given model as Figure 15. It means that when the knowledge acquisition can improve the problem solving ability, expectation of the class has a mediation effect.



(Figure 14) A Model for Mediation Effect

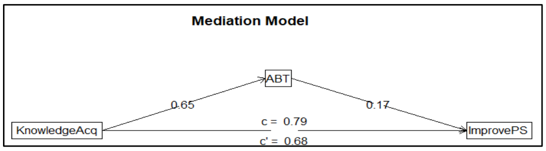
```
> library(psych)
> MArpt=mediate(y="ImprovePS", x="KnowledgeAcq", m="ExpectClass", data=MyData)
> print(MArpt, short=FALSE)
Call: mediate(y = "ImprovePS", x = "KnowledgeAcq", m = "ExpectClass",
    data = MyData)

The DV (Y) was  ImprovePS . The IV (X) was  KnowledgeAcq . The mediating variab
le(s) =  ExpectClass .

Total Direct effect(c) of  KnowledgeAcq  on  ImprovePS  =  0.79   S.E. =  0.03
 t direct =  27.5   with probability =  0
Direct effect (c') of  KnowledgeAcq  on  ImprovePS  removing  ExpectClass  =  0
.58   S.E. =  0.03  t direct =  18.73   with probability =  0
Indirect effect (ab) of  KnowledgeAcq  on  ImprovePS  through  ExpectClass   =
 0.21
Mean bootstrapped indirect effect =  0.21  with standard error =  0.03  Lower C
I =  0.15    Upper CI =  0.28
R2 of model =  0.8

 Full output

 Total effect estimates (c)
             ImprovePS   se    t Prob
KnowledgeAcq      0.79 0.03 27.5    0

Direct effect estimates     (c')
             ImprovePS   se     t Prob
KnowledgeAcq      0.58 0.03 18.73    0
ExpectClass       0.32 0.03 10.89    0

 'a'  effect estimates
            KnowledgeAcq   se     t Prob
ExpectClass         0.65 0.05 13.68    0

 'b'  effect estimates
            ImprovePS   se     t Prob
ExpectClass      0.32 0.03 10.89    0

 'ab'  effect estimates
             ImprovePS boot   sd lower upper
KnowledgeAcq      0.21 0.21 0.03  0.15  0.28
```



(Figure 15) The mediation effect on ExpectClass

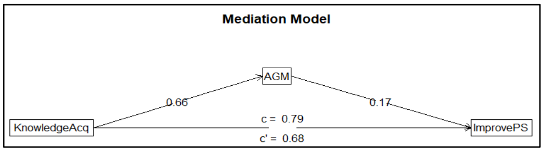
Since abstraction and algorithmic thinking are the key aspects of the computational thinking, the mediation effects on two aspects had been reviewed. Figure 16 is the mediation effect of abstraction while Figure 17 is the mediation effect of algorithmic thinking.

```
> MArpt2=mediate(y="ImprovePS", x="KnowledgeAcq", m="ABT", data=MyData)
> print(MArpt2, short=FALSE)
Call: mediate(y = "ImprovePS", x = "KnowledgeAcq", m = "ABT", data = MyData)

The DV (Y) was  ImprovePS . The IV (X) was  KnowledgeAcq . The mediating variable(s) =  ABT .

Total Direct effect(c) of  KnowledgeAcq  on  ImprovePS  =  0.79   S.E. =  0.03  t direct =  27
.5   with probability =  0
Direct effect (c') of  KnowledgeAcq  on  ImprovePS  removing  ABT  =  0.68   S.E. =  0.04  t d
irect =  18.2   with probability =  0
Indirect effect (ab) of  KnowledgeAcq  on  ImprovePS  through  ABT   =  0.11
Mean bootstrapped indirect effect =  0.11  with standard error =  0.03  Lower CI =  0.05    Up
per CI =  0.17
R2 of model =  0.73

 Full output

 Total effect estimates (c)
             ImprovePS   se    t Prob
KnowledgeAcq      0.79 0.03 27.5    0

Direct effect estimates     (c')
             ImprovePS   se     t     Prob
KnowledgeAcq      0.68 0.04 18.20 0.00e+00
ABT               0.17 0.04  4.34 1.94e-05

 'a'  effect estimates
    KnowledgeAcq   se     t Prob
ABT         0.65 0.04 15.52    0

 'b'  effect estimates
    ImprovePS   se    t     Prob
ABT      0.17 0.04 4.34 1.94e-05

 'ab'  effect estimates
             ImprovePS boot   sd lower upper
KnowledgeAcq      0.11 0.11 0.03  0.05  0.17
```



(Figure 16) The mediation effect of abstraction

```
> MArpt3=mediate(y="ImprovePS", x="KnowledgeAcq", m="AGM", data=MyData)
> print(MArpt3, short=FALSE)
Call: mediate(y = "ImprovePS", x = "KnowledgeAcq", m = "AGM", data = MyData)

The DV (Y) was  ImprovePS . The IV (X) was  KnowledgeAcq . The mediating variable(s) =  AGM .

Total Direct effect(c) of  KnowledgeAcq  on  ImprovePS  =  0.79   S.E. =  0.03  t direct =  27
.5   with probability =  0
Direct effect (c') of  KnowledgeAcq  on  ImprovePS  removing  AGM  =  0.68   S.E. =  0.04  t d
irect =  17.19   with probability =  0
Indirect effect (ab) of  KnowledgeAcq  on  ImprovePS  through  AGM   =  0.11
Mean bootstrapped indirect effect =  0.12  with standard error =  0.04  Lower CI =  0.05    Up
per CI =  0.19
R2 of model =  0.73

 Full output

 Total effect estimates (c)
             ImprovePS   se    t Prob
KnowledgeAcq      0.79 0.03 27.5    0

Direct effect estimates     (c')
             ImprovePS   se     t     Prob
KnowledgeAcq      0.68 0.04 17.19 0.00e+00
AGM               0.17 0.04  4.09 5.47e-05

 'a'  effect estimates
    KnowledgeAcq   se     t Prob
AGM         0.66 0.04 17.15    0

 'b'  effect estimates
    ImprovePS   se    t     Prob
AGM      0.17 0.04 4.09 5.47e-05

 'ab'  effect estimates
             ImprovePS boot   sd lower upper
KnowledgeAcq      0.11 0.12 0.04  0.05  0.19
```



(Figure 17) The mediation effect of AGM

4. Conclusions & Further Research

The results of the verification of the educational effectiveness and the analysis for the factors of computational thinking are summarized as follows.

In this paper, it is carefully reviewed for the educational effectiveness with students who had experienced computational thinking in the smart learning environment. As the result, the effect of education is remarkably improved compared with traditional learning environment.

We also conducted the big data analysis to extract keywords in the education field of the 4th industrial revolution era. The 4th industrial revolution was related with the educational revolution, and the big data analysis of the educational revolution gave the keyword as software education. When we put 'software education' for the big data analysis, the term of computational thinking was appeared. It implies that people think the concept of computational thinking is a requirement for the 4th industrial revolution.

The survey was conducted to find out the singularity of computational thinking. We analyzed whether there is a difference in the level of computational thinking by majors. The results showed a different level in learning. This means that it needs to teach students differently by their majors to maximize the effectiveness of the education.

Since it is proved that the concept of computation thinking is important for the future education, the factor analysis for computational thinking was conducted. Through literature review, we come up with thirteen factors for the computational thinking. As the result of various analysis with 13 factors, only 2 elements: abstraction and algorithmic thinking were influenced singly while other 11 factors were combined with other factors to affect new factors.

This study also analyzed the factors that have mediating effect on the improvement of problem solving ability. As a result of the mediation effect analysis, it clearly showed that the higher the expectations for learning, the better the problem solving ability. In addition, the improvement in abstraction and algorithmic thinking can lead the problem solving ability.

Various studies are needed to enhance the effectiveness of education in the smart learning environment. Computational thinking is a hot issue in education, and it has been proved that there is the benefits of computational thinking in this study. Consequently, educators should study how to apply computational thinking approach into their classes for better educational effect.

If computational thinking is engaged with the real education field, students may acquire better quality education. To become a protagonist in the era of the 4th industrial revolution, every possible effort should be made. A study for a better future is a worthwhile effort. Through further study, students should be provided with the perfect educational environment and educational methods.

Reference

[1] O. Han and J. Kim, "A study on an Improvement to the Effectiveness of Education in Smart Learning Environment", in Proceedings of the 12th APIC-IST, June 25-28, 2017.

[2] H. Han and J. Lee, "Comprehension of the Competency-based Curriculum by Comparison with the Discipline-based Curriculum and the Understanding by Design", Journal of Curriculum Studies, Vol 35, No.3, pp.203-221, 2017. http://210.101.116.16/kiss6

[3] M. Peters, "Technological unemployment: Educating for the fourth industrial revolution", Journal of Educational Philosophy and Theory, Vol.49, No.1, pp.1-6, 2017. http://dx.doi.org/10.1080/00131857.2016.1177412

[4] A. Toffler, "Think outside the box: Message for Korean", Yonhap News, 2016/06/30. http://www.yonhapnews.co.kr/bulletin/2016/06/30/0200000000AKR20160630169100797.HTML

[5] Z. Zhu, M. Yu, and P. Riezebos, "A research framework of smart education", Smart Learning Environments, Vol 3, No. 4, pp.1-17, 2016. https://doi.org/10.1186/s40561-016-0026-2

[6] W. Jun and S, Hong, "A Study on Development of Quality Standards of Educational Smart Contents", KSII Transactions on Internet and Information Systems, Vol 8, No.6, pp.2152-2170, 2014. http://dx.doi.org/10.3837/tiis.2014.06.020

[7] T. Kim, J. Y. Cho, and B. G. Lee, "A Study on the Development Strategy of Smart Learning for Public Education", Journal of Internet Computing and Services(JICS), Vol 16, No.6, pp.123-131, 2015. http://dx.doi.org/10.7472/jksii.2015.16.6.123

[8] M. Kim, "A Study on Educational Application of Smart Devices for Enhancing the Effectiveness of Problem Solving Learning", Journal of Internet Computing and Services(JICS),Vol 15, No.1, pp. 143-156, 2014. http://dx.doi.org//10.7472/jksii.2014.15.1.143

[9] J. Wing, "Computational Thinking", Journal of Computing Sciences in Colleges, Vol 24, No.6, pp. 6-7, 2009. https://dl.acm.org/citation.cfm?id=1529997

[10] D. Barr, J. Harrison, and L. Conery, "Computational Thinking: A digital Age Skill for Everyone", International Society for Technology in Education,Vol 38,No. 6, pp.20-23, 2011. https://eric.ed.gov/?id=EJ918910

[11] V. Barr and C. Stephenson, "Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community", ACM Inroads,Vol 2, No. 1, pp.48-54, 2011. http://dx.doi.org/10.1145/1929887.1929905

[12] S. Grover and R. Pea, "Computational Thinking in K-12: A Review of the State of the Field", Educational Researcher,Vol 42, Issue 1, pp.38-43, 2013. http://dx.doi.org/10.3102/0013189X12463051

[13] A. Ioannidou, V. Bennett, A. Repenning, K. Koh, and A. Basawapatna, "Computational Thinking Patterns", the Annual Meeting of the American Educational Research Association, pp.1-15, 2011. https://eric.ed.gov/?id=ED520742

[14] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith and L. Werner, "Computational Thinking for youth in practice", ACM Inroads, Vol 2, Issue 1, pp.32-37, 2011. http://dx.doi.org/10.1145/1929887.1929902

[15] I. Lee, F. Martin and K. Apone, "Integrating Computational Thinking across the K－8 Curriculum", ACM Inroads, Vol 5, Issue 4, pp. 4-71, 2014. http://dx.doi.org/10.1145/2684721.2684736

[16] L. Mannila, V. Dagiene, B. Demo, N. Grgurina, C. Mirolo, L. Rolandsson and A. Settle, "Computational Thinking in K-9 Education", Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference, pp.1-29, 2014. http://dx.doi.org/10.1145/2713609.2713610

[17] L. Miller, L, Soh, V. Chiriacescu, E. Ingraham, D. Shell and M. Hazley, "Integrating Computational and Creative

Thinking to Improve Learning and Performance in CS1", Proceedings of the 45th ACM technical symposium on Computer Science Education, pp.475-480, 2014. http://dx.doi.org/10.1145/2538862.2538940

[18] J. Rode, A. Weibert, A. Marshall, K. Aal, T. Rekowski, H. Mimouni and J. Booker, "From Computational Thinking to Computational making", Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp.239-250, 2015. http://dx.doi.org/10.1145/2750858.2804261

[19] L. Seiter and B. Foreman, "Modeling the learning progressions of computational thinking of primary grade students", Proceedings of the ninth annual international ACM conference on International Computing Education Research, pp.59-66, 2013. http://dx.doi.org/10.1145/2483384.2493403

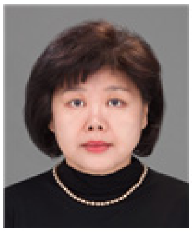
[20] M. Voskoglou and S. Buckley, "Problem Solving and Computational Thinking in a Learning Environment", Egyptian Computer Science Journal, Vol 36, No. 4, pp.28-46, 2012. https://arxiv.org/abs/1212.0750

[21] J. Wing, "Viewpoint: Computational Thinking", Communications of the ACM, Vol 49, Issue 3, pp. 33-35,2006.http://dx.doi.org/10.1145/1118178.1118215

[22] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch and J. Korb, "Introducing computational thinking in education courses", Proceedings of the 42nd ACM technical symposium on Computer science education, pp.465-470, 2011. http://dx.doi.org/10.1145/1118178.1118215

[23] A. Yadav, C. Mayfield, N. Zhou, S. Hambrusch and J. Korb, "Computational Thinking in Elementary and Secondary Teacher Education", ACM Transactions on Computing Education, Vol 14, Issue 1, Article No. 5, 2014. http://dx.doi.org/10.1145/2576872

◐ 저 자 소 개 ◑



한 옥 영(Oakyoung Han)
1985년 The University of Kansas 대학교 Computer Science 학과 (B.S. 학사)
1989년 California 주립 대학교 대학원 Computer Science 학과 (M.S. 석사)
1999년 한국과학기술원(KAIST) 대학원 전산학과 (박사수료)
2012년 성균관대학교 대학원 컴퓨터교육학과 (박사)
1989년~1991년 삼성전자 컴퓨터부문 응용개발실
1991년~1999년 덕성여자대학교 전산과장, 연구교수
2000년~현재 성균관대학교 컴퓨터교육과 겸임교수
2015년~현재 성균관대학교 성균SW교육원
관심분야 : 소프트웨어교육, 컴퓨팅사고력, 자료구조, 알고리즘, 인공지능, SW+인문, 인터넷중독, etc.
E-mail : oakyoung@skku.edu

김 재 현(Jaehyoun Kim)
1988년 성균관대학교 수학과 졸업(학사)
1992년 웨스턴일리노이 주립대학교 대학원 전산학과 석사
2000년 일리노이공과대학교 대학원 전산학과 박사
2014년~현재 한국컴퓨터교육학회 부회장
2010년~현재 한국인터넷정보학회 부회장
2002년~현재 성균관대학교 컴퓨터교육과 교수
2016년~현재 성균관대학교 성균SW교육원 원장
관심분야 : 객체지향 소프트웨어공학, 컴퓨터교육, Computer Based LET etc.