Journal of The Korean Association
of Information Education Vol. 24, No. 6,
December 2020, pp. 653-662

http://dx.doi.org/10.14352/jkaie.2020.24.6.653

초등 실과 교과서 내 소프트웨어교육 영역에 나타난 컴퓨팅 사고력 요소 분석

김정랑
광주교육대학교 컴퓨터교육과

요약

본 연구에서는 초등 실과 교과서의 소프트웨어교육 영역에 나타난 컴퓨팅 사고력의 요소의 내용, 수준 등을 분석하고 교과서 학습활동별 컴퓨팅 비율을 분석하였다. 컴퓨팅 사고력 요소는 교육부에서 제시한 컴퓨팅 사고력의 구성요소와 정의를 바탕으로 구성하였고, 6개 출판사에서 출판된 실과 교과서의 소프트웨어 교육 학습 내용을 분석하였다. 교과서별 컴퓨팅 사고력 요소와 컴퓨팅 수준을 분석한 결과 교과서별로 컴퓨팅 사고력의 하위 요소 포함 여부의 차이가 나타나며, 컴퓨팅과 연결된 추상화 활동을 제시하는 교과서의 비율이 비교적 낮게 나타났다. 추후 교육과정 개편 또는 교과서 개정시 컴퓨팅 사고력 요소가 균형있게 포함되도록 보완하고, 문제 발견과 이해부터 자동화까지 컴퓨팅 사고의 일련의 과정들이 이어질 수 있도록 차시를 편성할 필요가 있으며, 일상생활의 요소를 사용하되 컴퓨팅이 포함된 추상화 활동을 내포할 필요가 있다.

키워드 : 소프트웨어교육, 컴퓨팅 사고력, 교과서 분석, 추상화, 자동화

Analysis of Computational Thinking Skills in the Software Education field in Elementary Practical Educations Textbooks

Jeong-Rang Kim
Dept. of Computer Education, Gwangju National University of Education

ABSTRACT

In this study, the content and level of the elements of Computational Thinking in the Software Education area of elementary practical textbooks were analyzed, and also the computing ratio for each textbook learning activity was analyzed. The elements of Computational Thinking were defined based on the components and definitions of Computational Thinking skills suggested by the Ministry of Education. The contents of Software Education area in practical arts textbooks published by six publishers were analyzed. As a result of analyzing the elements of Computational Thinking for each textbook according to the achievement criteria, there was a difference in whether sub-elements of Computational Thinking were included for each textbook. Second, as a result of analyzing the level of Computing of learning content, the proportion of textbooks presenting Abstract activities connected to Computing was relatively low. When the curriculum is reorganized or the textbook is revised in the future, it is necessary to complement the elements of Computational Thinking in a balanced way, and to include general Abstraction activities and Abstraction activities that can lead to Automation.

Keywords : Software Education, Computational Thinking, Textbook Analysis, Abstraction, Automation

---

본 연구는 2020년 광주교육대학교 연구년 교수연구비 지원에 의해 연구를 수행하였음.
교신저자 : 김정랑(광주교육대학교 컴퓨터교육과)
논문투고 : 2020-11-26
논문심사 : 2020-12-04
심사완료 : 2020-12-14

1. 서론

소프트웨어가 중심이 되는 지능정보사회의 확산은 우리 사회의 패러다임을 급격하게 변화시키고 있다[11]. 인공지능, 데이터 과학 등의 발전은 이러한 변화를 더욱 가속시킬 것으로 보인다.

이러한 변화에 대응하기 위해 2015 개정 교육과정에 제시된 소프트웨어교육은 초등학교, 중학교 교육과정에서 필수 과정으로 지정되었다[17]. 이에 따라 2019년부터 소프트웨어교육이 초등 실과에 17차시, 중등 정보 교과에서 34차시가 배정되었다. 초등학교 소프트웨어교육의 목적은 정보윤리의식과 태도를 바탕으로 실생활의 문제를 컴퓨팅 사고(Computational Thinking)로 해결할 수 있는 역량을 함양하여 창의·융합형 인재를 양성하는 것을 목적으로 한다[15].

한편, Wing(2008)은 컴퓨팅 사고력을 '컴퓨터의 기본 개념을 바탕으로 시스템을 설계하고 문제를 해결하여 인간의 행동을 이해하도록 하는 것'으로 정의하고, 컴퓨팅 사고력의 구성 요소를 크게 추상화(Abstraction)와 자동화(Automation)로 분류하였다[22]. 추상화는 복잡한 문제를 강조해야 할 부분과 생략할 부분을 결정하는 과정이고, 자동화는 추상화를 통해 만들어진 추상 개념을 컴퓨터가 해석하여 실행할 수 있도록 하는 과정이다.

2015 개정 교육과정에서도 정보 교과의 핵심 역량으로 컴퓨팅 사고력을 제시하고 있으며 그 하위 요소로 추상화와 자동화를 강조하고 있다[15].

컴퓨팅에서의 추상화는 일반적인 추상화와 포괄적인 개념은 서로 일맥상통하지만, 좁은 의미에서는 그 의미와 요소에 차이가 있다[4]. 컴퓨팅 사고력의 추상화는 후속 요소인 자동화에 연결되는 개념으로, 추상화 결과물이 컴퓨팅을 통해 처리하는 자동화된 프로그램의 형태로 나타난다. 그러므로 초등학교 소프트웨어교육 영역에서 추상화의 개념은 일반적인 추상화와 다르게 제시되어야 하고[10], 추상화 활동은 자동화 활동으로 이어지도록 구성되어야 한다. 소프트웨어교육 운영 지침에서는 컴퓨팅 사고력을 기르기 위한 교육 방법으로 실생활의 문제를 인식하고 분석하는 단계에서부터 문제해결 절차에 따라 알고리즘을 설계하고, 이를 프로그램으로 구현하는 일련의 종합 활동을 권장하고 있다[16].

그러나 초등 실과 교과서에 포함된 소프트웨어교육 영역의 시수는 성취기준별로 차이가 커 체계적인 내용 구성이 필요하며[9], 현재 중등 정보 교과서는 대부분 일반적 추상화와 컴퓨팅 사고력의 추상화를 혼용하고 있고, 일부 교과서에는 일반적 추상화의 개념만 제시하고 있는 것으로 나타났다[10].

중등 교과서의 사례처럼 일반적인 추상화 개념만 다루는 경우, 추상화 활동을 자동화로 이어지도록 구성하지 못하여 학생들에게 컴퓨팅 사고력에 대한 오개념을 형성할 소지가 있다. 따라서, 초등 실과 6종 교과서에 제시된 추상화의 개념 및 관련 활동에 대한 면밀한 분석을 통해 초등학교 소프트웨어교육에 적합한 추상화 교육 방안을 제시할 필요가 있다.

본 연구에서는 초등 실과 교과서의 소프트웨어교육 영역에 나타난 컴퓨팅 사고력의 요소의 내용, 컴퓨팅 수준 등을 분석하여 그 시사점을 도출하였다.

2. 이론적 배경

2.1. 컴퓨팅 사고력

컴퓨팅 사고력이란 컴퓨터 과학자처럼 사고하는 방식으로, 컴퓨터 과학의 기본 개념과 원리에 따른 문제 해결, 시스템 설계, 인간 행동의 이해를 위한 사고로 정의되며 재귀적 사고, 추상적 사고, 선행적 사고, 절차적 사고, 논리적 사고, 동시적 사고, 분석적 사고, 전략적 사고 등을 포함하는 개념이다[5][21].

학자들마다 견해가 다르지만, 컴퓨팅 사고력은 분석적 사고력부터 창의적 사고력까지 수렴적인 사고과정과 확산적인 사고과정을 포함한다[7]. Wing은 컴퓨팅 사고력의 구성 요소로 추상화(Abstraction)와 자동화(Automation)로 분류하였고[21], 이후 미국의 CSTA(Computer Science Teacher Association)와 ISTE(International Society for Technology in Education), Google에서는 컴퓨팅 사고력에 대한 구성요소를 아홉 가지(자료 수집, 자료 분석, 자료 표현, 문제 분해, 추상화, 알고리즘과 절차, 자동화, 시뮬레이션, 병렬화)로 세분화하여 제시하였다[2][3]. 국내에서는 교육부와 한국교육학술정보원에서 컴퓨팅 사고력의 구성요소를 <Table 1>과 같이 제시하였다[16].

<Table 1> Components of Computational Thinking

<table>
<tr><th>Wing (2008)</th><th>CSTA & ISTE(2011)</th><th>Google for Education(2015)</th><th>MoE (2015)</th></tr>
<tr><td rowspan="10">Abstr action</td><td colspan="3">Data Collection</td></tr>
<tr><td rowspan="2">Data Analysis</td><td>Data Analysis</td><td>Data Analysis</td></tr>
<tr><td>Pattern Recognition</td><td rowspan="2">Data Represen -tation</td></tr>
<tr><td colspan="2">Data Representation</td></tr>
<tr><td colspan="2">Decomposition</td><td>Decompo -sition</td></tr>
<tr><td rowspan="2">Abstraction</td><td>Abstraction</td><td rowspan="2">modelling</td></tr>
<tr><td>Pattern Generalization</td></tr>
<tr><td>Algorithm & Procedures</td><td>Algorithm Design</td><td>Algorithm</td></tr>
<tr><td rowspan="3">Auto- mation</td><td colspan="2">Automation</td><td rowspan="2">Coding</td></tr>
<tr><td colspan="2">Parallelization</td></tr>
<tr><td colspan="3">Simulation</td></tr>
<tr><td></td><td colspan="2"></td><td>Generali- zation</td></tr>
</table>

2.3. 컴퓨팅 사고의 추상화

추상화는 일반적으로 구체적인 것을 단순화하고 세부적인 것을 소거하는 과정, 특정 요소의 공통적인 핵심과 본질에 접근하기 위해 일반화하는 과정을 의미한다[3].

Wing(2006)은 추상화와 분해를 통해 복잡한 시스템을 설계하거나 어려운 문제를 해결하는 것이라고 하며 큰 문제를 덩어리로 분리해내는 능력으로 정의하였다[21]. 안상진 외(2012)는 실제 일어날 수 있는 문제를 계산 가능한 추상화된 문제로 단순화시키는 것으로 정의하였으며 추상화의 특징을 현실 세계에서 일어날 수 있는 문제를 컴퓨터가 계산 가능한 형태로 표현하는 것, 문제 해결 방법의 복잡도를 낮추기 위해 여러 단계를 거칠 수 있는 점 등을 제시하였다[1].

2015 개정 교육과정에서는 학교급에 따라 추상화에 관한 내용을 기술하고 있다[15]. 중학교 수준은 실생활 문제 상황에서 문제의 현재상태, 목표상태를 이해하고 목표 상태에 도달하기 위해 수행해야 할 작업을 분석하며, 문제해결에 필요한 요소와 불필요한 요소를 분류하는 것으로 설명하고 있다. 고등학교 수준에서는 복잡한 문제 상황에서 문제의 현재상태, 목표상태를 이해하고 목표 상태에 도달하기 위해 수행해야 할 작업을 분석하는 과정, 복잡한 문제 상황에서 문제 해결에 불필요한 요소를 제거하거나 필요한 요소를 추출하는 과정, 복잡하고 어려운 문제를 해결 가능한 작은 단위의 문제로 분해하고 모델링하는 과정으로 설명하고 있다.

3. 연구 방법

3.1. 연구 절차

본 연구에서 분석한 컴퓨팅 사고력 요소는 교육부에서 제시한 컴퓨팅 사고력의 구성요소와 정의를 바탕으로 6개 출판사에서 출판된 실과 교과서의 소프트웨어 교육 학습 내용을 분석하였다[6][13][14][18][19][20]. 그리고 두 차례의 델파이 분석을 거쳐 분석한 컴퓨팅 사고력 요소 및 컴퓨팅 수준에 대해 타당성을 검증하고 수정하여 확정하였다.

3.2. 연구대상

연구대상 교과서는 2015 개정 교육과정에 따른 실과 교과서 6종이며, 각 실과 교과서의 소프트웨어 교육 영역에 대해 연구를 실시하였다. 교과서별 기초 정보는 <Table 2>와 같다.

<Table 2> Textbook basic information for research

Textbook (sign)	Unit	Pages	Hours
금성(A)	3. Software and life	23	18
동아(B)	4. Communicating Programming	30	17
미래엔(C)	3. Software and life	24	18
교학사(D)	4. Software in life	20	17
비상교육(E)	4. Communicating Software	26	17
천재(F)	5. Easy to learn software and programming	24	17

3.3. 분석 준거 및 방법

교과서별 컴퓨팅 사고력 요소와 학습내용의 컴퓨팅 수준을 분석하기 위해 본 논문에서 컴퓨팅 사고력 요소는 <Table 3>과 같이 CSTA&ISTE(2011)의 CT요소를 바탕으로 제작된 교육부 SW교육 운영 지침을 근거로 하였다[16]. 분석시 교과서별 학습 활동이 분석 준거의

활동이나 사고 과정을 학생이 직접적으로 하게끔 제시할 경우 CT요소가 포함된 것으로 판단하였으며, 생략하거나 그 결과를 미리 제시할 경우 CT요소가 포함되지 않는 것으로 판단하였다.

<Table 3> Analysis Criteria of CT Component

Components		Definition
Data Collection		Gathering Data for problem solving
Data Analysis		Data Understanding, finding patterns, Conclusions
Data Representation		Visualize problems such as graphs, charts, and pictures
Abstraction	Decomposition	Dividing the problem into manageable levels
	Modelling	Extraction of problem solving key elements, model creation
	Algorithm	express problem solving steps to Algorithm
Automation	Coding	Automation using programming language
	Simulation	Running the program
Generalization		Applying the problem-solving process to other problems

학습활동의 컴퓨팅 수준을 분석하는 준거는 김수환(2018)의 광의의 추상화와 협의의 추상화 분석 방법을 사용하였다[10]. 자동화의 경우 프로그래밍언어를 이용하여 자동화하고 실행하는 과정이므로 컴퓨팅 수준 분석에서 제외하였으며, 자료수집, 자료분석, 구조화, 추상화 단계에서 컴퓨팅 및 CS와 직접적인 관련이 있는 추상화 활동을 제시하는지, 일상 생활이나 타 교과에서도 활용가능한 일반적 추상화 활동을 제시하는지 분석하였다. 컴퓨팅 활동이나 소프트웨어 설계 과정에서 이루어지는 추상화는 데이터 추상화(Data abstraction), 기능 추상화(Functional abstraction)이 있으며, 학자에 따라 절차적 추상화(Procedural abstraction)과 컨트롤 추상화(Control abstraction)를 포함하기도 한다. 이들 추상화는 일반적인 의미의 추상화인 '중요한 부분을 단순한 것으로 표현하는 것'에 대해서는 공통적이나, SW 개발을 위한 알고리즘 설계, 객체 설계, 함수 및 변수 설계 등 컴퓨팅 활동을 위해 사용되는 협의의 추상화라는 점에서 일반적인 추상화와 차이가 있다. 김수환(2018)은 컴퓨터 과학과의 연계성을 고려하여 세 가지 추상화의 내용을 제시하였다[10]. 기능 추상화란 추상화된 객체(스프라이트, 오브젝트) 등을 생각해 보고, 정의하는 활동 등을, 컨트롤 추상화는 문제해결을 위한 작업순서를 생각해 보고, 단순화 하는 활동, 알고리즘을 만들어 보는 활동 등을, 데이터 추상화란 문제의 기반이 되는 자료를 추상화하여 컴퓨팅이 용이한 자료구조를 만들거나 자료의 속성, 변수를 정의하는 활동 등을 의미한다.

본 논문에서는 교과서 분석을 위해 내용분석법을 활용하였으며 양적 내용 분석과 질적 내용 분석을 활용하였다. 양적으로는 전체 학습 활동 중 컴퓨팅과 관련된 활동이 어느 정도 포함되었는지 분석하여 제시하였고, 질적으로는 단위 학습 활동에 컴퓨팅 사고력 요소가 포함되었는지 교과서 맥락을 통찰하여 분석하였다. 질적 내용 분석의 객관성을 확보하기 위해 두 차례의 델파이 분석을 실시하여 내용 타당도를 검증하고 수정하였다.

3.3. 델파이 분석

각 교과서별 컴퓨팅 사고력 구성요소와 학습내용의 컴퓨팅 수준 분석에 대해 컴퓨터 교육을 전공한 현장교사를 10명을 대상으로 두 차례 타당도 검증을 실시하였다. 분석 내용은 각 교과서의 소프트웨어 교육 영역 세부 활동에 대한 컴퓨팅 사고력 요소(자료 이해, 자료 표현, 알고리즘 등) 포함 여부 및 비율, 세부 활동의 컴퓨팅(자료/기능/컨트롤 추상화 등) 포함 여부 및 컴퓨팅 수준이며 그 결과는 <Table 4>와 같다. 타당도는 Lawshe(1975)의 내용 타당도 비율(Content Validity Ratio, CVR)로 검증하였으며 내용 타당도는 0.62 이상일 경우 타당한 것으로 간주하였다[12].

<Table 4> Validity results of Delphi analysis

Text-book	Element/Level	1st		2nd	
		M	CVR	M	CVR
A	CT Element General/Computing level	4.30	0.53	4.25	0.75
B	CT Element General/Computing level	4.27	0.50	4.42	0.77
C	CT Element General/Computing level	4.54	0.70	4.36	0.80
D	CT Element General/Computing level	4.66	0.75	4.30	0.90
E	CT Element General/Computing level	4.63	0.82	4.47	0.83
F	CT Element General/Computing level	4.51	0.60	4.36	0.87

첫 번째 차례에서는 각 교과서의 컴퓨팅 사고력 구성요소와 주요 학습내용에 대한 컴퓨팅 수준을 분석하여 타당도 검사를 실시하였으며, 두 번째 차례에서는 각 교과서의 학습내용을 세부적으로 분류하여 컴퓨팅 사고력 구성요소와 학습내용의 컴퓨팅 수준을 세세하게 분석하고 첫 번째 타당도 검사에서 타당도 점수가 낮은 항목을 수정·보완하여 타당도 검사를 재실시하였다.

4. 연구 결과

4.1. 교과서별 주요 학습활동

초등 실과 교과서의 컴퓨팅 사고력 요소와 컴퓨팅 수준을 분석하기 위해 각 출판사의 소프트웨어교육 내용에 대해 성취기준별로 주요 학습활동을 분석하였다. 성취기준은 소프트웨어교육에 해당하는 [6실04-07]부터 [6실04-11]을 기준으로 하였다. 주요 학습활동은 김정랑(2019)의 성취 기준에 따른 교과서 활동 분류를 바탕으로 주요 내용을 요약·선정하였다[8].

교과서별 소프트웨어교육 성취기준에 따라 교과서 학습 활동을 분류한 결과, 주요 학습활동은 유사한 것으로 나타났으며, 구체적인 학습 활동은 교과서별로 상이한 것으로 나타났다. 성취기준별 시수는 다소 차이가 있는 것으로 나타났다. [6실04-07]은 1-3차시 사이로 상대적으로 차이가 적게 나타났고, [6실04-08]과 [6실04-11]은 각각 2-6차시, 4-8차시로 상대적으로 차이가 크게 나타났다. 교과서별 학습 활동을 분류한 결과는 <Table 5>와 같다.

4.2. 컴퓨팅 사고력 요소 분석

성취기준에 따라 교과서별 컴퓨팅 사고력 요소를 분석한 결과는 <Table 6>과 같다.

성취기준 [6실04-07]은 소프트웨어의 의미를 알고, 생활 속 사용되는 소프트웨어를 탐색하는 활동으로 컴퓨팅 사고력 요소가 포함되지 않아 분석에서 제외하였다. 성취기준 [6실04-08]은 절차적 사고(알고리즘)에 따른 문제 해결의 순서를 생각하고 적용하는 것으로 교과서에 따라 프로그래밍이 포함되지 않고 탐구·조사·놀이 활동 등으로 구성되어 있으며, 일부 교과서의 경우 프로그래밍을 하기 위한 순서도를 완성하는 활동이 포함되었다.

<Table 5> Main activities by textbook

Achieve-ment standards	Text-book	Main activities	Hours
6실04-07	A	• Understanding of computer & SW • Software and life	2
	B	• Understanding meaning, type and function of Software • The effect of Software on our life	2
	C	• Understanding meaning of Software • The effect of Software on our life	2
	D	• Understanding meaning, type and function of Software • The effect of Software on our life	3
	E	• Understanding meaning of Software • The effect of Software on our life	1
	F	• Understanding meaning of Software • Software and life • The case of software making life easier	3
6실04-08	A	• Understanding meaning of Algorithm • Solving Problems with Algorithm	3
	B	• Understanding meaning of Algorithm • Understanding meaning of Algorithm process • Solving Problems with Algorithm	3
	C	• Understanding meaning of Algorithm • Solving everyday life problems • Exploring the route of the amusement park • Playing a picture sorting game	6
	D	• Dividing procedures for problems in life • Expressing problems in everyday life in words and pictures	3
	E	• Understanding meaning of Algorithm • Creating a Drone Moving Path • Planning to use amusement parks	3
	F	• Problems, problem solving meanings, life-in-life examples • Stacking paper cups, finding directions	2
6실04-09	A	• Understanding of Programming • Explore programming languages • Experience Programming	3
	B	• Experience Programming	3
	C	• Understanding of Programming • Experience Programming • Creating an animation program	4
	D	• Understanding of Programming • Understanding of How to Use Entry • decorating a story with an entry	3
	E	• Understanding of Programming • Creating a program to introduces me	2
	F	• Understanding of Programming • Programming Components • Creating a simple program • Debugging the program	3

Achieve-ment standards	Text-book	Main activities	Hours
6실04-10	A	• Creating a program with input, processing, and output	2
	B	• Understanding the program execution process	1
	C	• Understanding input, processing, and output of software • Creating an Age Calculation Program	2
	D	• Making a calculator with input and output • Creating a two-word link program	2
	E	• Understanding the problem, how to solve the problem • Expressing problem solving methods • Creating a program with input and output	2
	F	• Understanding input, processing, and output of software • Design and Manufacture of Numerical Input-processing Program • Design and Manufacturing Character Input-processing Program	4
6실04-11	A	• Understanding of Sequence, Selection, and Loop structure • Creating Sequence Structure Program • Creating Selection Structure Program • Creating Loop Structure Program	8
	B	• Making program with Sequence,Selection,Loop	8
	C	• Understanding of Sequence, Selection, and Loop structure • Creating an electronic locking device program	4
	D	• Sequence,Loop,Selection Structure Design • Creating and sharing my own program	6
	E	• Understanding of Sequence, Selection, and Loop structure • Cushion Quiz game Programming • Problem solving method expression	8
	F	• Understanding of Sequence, Selection, and Loop structure • Creating a simple program • Debugging the program	4

<Table 6> Components of CT by textbook

Achievement standards	Text-book	Components of CT
6실04-08	A	• Data Analysis, Algorithm
	B	• Data Collection, Decomposition, Abstraction, Algorithm
	C	• Data Collection, Data Analysis, Decomposition, Abstraction
	D	• Data Collection, Decomposition, Algorithm
	E	• Data Analysis, Representation, Decomposition, Abstraction, Algorithm
	F	• Data Collection, Decomposition, Algorithm
6실04-09	A	
	B	• Algorithm, Automation, Simulation
	C	• Representation, Abstraction, Algorithm, Automation, Parallelization
	D	• Automation, Simulation
	E	• Algorithm, Automation, Simulation
	F	• Automation, Simulation
6실04-10	A	• Automation, Simulation
	B	• Algorithm, Automation, Simulation
	C	• Decomposition, Abstraction, Algorithm ,Automation, Simulation
	D	• Data Analysis, Decomposition, Algorithm, Automation, Simulation
	E	• Data Collection, Data Analysis, Representation, Decomposition, Abstraction, Algorithm, Automation
	F	• Automation, Simulation
6실04-11	A	• Automation, Simulation
	B	• Algorithm, Automation, Simulation
	C	• Algorithm, Automation, Simulation
	D	• Data Analysis, Decomposition, Algorithm, Automation, Simulation
	E	• Data Collection, Data Analysis, Representation, Decomposition, Abstraction, Algorithm, Automation, Simulation
	F	• Algorithm, Automation, Simulation

[6실04-09]는 프로그래밍의 의미를 알고 체험해보는 활동으로 대부분의 교과서가 블록형 프로그래밍 언어 도구의 화면 구성을 알아보고 프로그램을 실행시켜 간단한 동작을 통해 프로그래밍을 체험해보는 활동으로 구성되어 있다. 컴퓨팅 사고력 요소의 관점에서 자료수

집, 자료분해, 추상화 등은 나타나지 않았고 자동화와 시뮬레이션으로 대부분 구성되어 있다. 일부 교과서의 경우 추가 시수를 확보하여 문제를 제시하고 해결하기 위한 과정을 설계하고(추상화·알고리즘), 간단한 동작을 통해 프로그램을 실행하는 활동이 포함되었다. [6실04-10]은 입력, 처리, 출력과정이 포함된 단순한 프로그램을 설계하고 실행하는 활동으로 교과서별 활동의 차이가 나타났다. A, B, F교과서의 경우 숫자·문자의 입력, 처리, 출력 과정과 추상화된 알고리즘을 직접적으로 제시하고 프로그래밍 활동에 집중하는 것으로 보이며, C, D, E교과서의 경우 문제 해결 방법을 생각하여 순서를 설계한 후 프로그래밍하는 활동으로 구성하였다. [6실04-11]은 순차, 선택, 반복 구조를 이해하고 문제를 해결하는 프로그램을 만드는 활동이며 교과서별 활동의 차이가 나타났다. A, D교과서의 경우 순차, 반복, 선택 구조의 프로그램을 별도로 제작하는 활동으로 구성하였으며, B, C, F교과서의 경우 순차, 반복, 선택 구조에 대해 학습한 후, 세 구조가 혼합된 프로그램을 설계하여 제작하는 활동으로 구성하였다. E교과서의 경우 순차구조로 프로그램을 제작한 후 반복과 선택 구조로 변경하도록 활동을 구성하였고, 프로그램 설계와 제작 과정이 모두 포함된 활동을 별도로 구성하였다.

4.3. 학습 내용의 컴퓨팅 수준 분석

교과서의 학습활동에 따라 컴퓨팅 활동이 어느 정도 포함되어있는지 분석한 결과는 <Table 7>과 같다. 종이접기, 달걀 삶기 방법 표현 등 일상 생활의 절차적인 표현은 '일반'으로, CS의 정렬, 알고리즘 등이 포함된 활동은 '컨트롤'로, 오브젝트 등 추상화된 객체를 생각해보거나 정의하는 활동이 포함되었을 경우 '기능'으로, 객체의 자료 속성, 변수 등을 정의할 경우 '데이터'로, 추상화 없이 프로그램 제작 활동이 포함된 경우 '자동화'로 표기하였고 컴퓨팅 포함 비율은 기능,컨트롤,데이터 추상화 포함 여부를 전체 활동 수로 나누어 그 비율을 계산하였다. 대부분의 교과서는 일상 생활의 절차적 사고 표현 활동과 교과서에 제시된 프로그램을 실습·제작하는 활동으로 구성되어 있다. B교과서의 경우 로봇의 구출 활동에서 추상화된 기호를 사용하거나 '앞으로 가기' 등 단순화한 알고리즘이 포함되어 있다. C교과서는 정렬망 활동, 다익스트라 알고리즘 등이 포함되어 있으며, 프로그램 제작 과정에서 오브젝트 정의, 변수 정의, 알고리즘 제작 등이 포함되어 있다. E교과서의 경우 문제 해결 방법 생각·표현(알고리즘 설계), 오브젝트 정의, 자료 속성 정의 등이 모두 제시된 활동이 포함되어 있다.

<Table 7> Computing Ratio by Textbook's Learning Activities

Text-book	Main activities	ratio	Ratio (%, Computing / Total)
A	• Commanding the robot to get into the car	General	50 (3/6)
	• Paper Frog Folding	General	
	• Command activity	General	
	• Experience Programming	Auto-mation	
	• Programming with input, processing, output	Auto-mation	
	• Sequence, Selection, Loop structure	Auto-mation	
B	• Making Local travel guide material	General	25 (2/8)
	• Eggs boiling activities	General	
	• Decryption activities	General	
	• Rescue robot activities	Control	
	• Basic Programming activities	Function	
	• Making a program to calculate the purchase amount of preparations	Auto-mation	
	• Sequence,Selection,Loop Program	Auto-mation	
	• Making Happy piggy bank program	Auto-mation	
C	• Algorithm applied to everyday problems	General	57 (4/7)
	• How to get as many rides as possible in an amusement park	Control	
	• Sorting numbers (sorting network)	Control	
	• Basic programming experience	Auto-mation	
	• Creating an animation program	Auto-mation	
	• Programming age calculation	Control, Data	
	• Creating an electronic locking device program	Control, Data	
D	• Expressing motion in text/ picture	General	30 (3/10)
	• Museum tour plan	General	
	• Command to bring the robot home	Control	
	• Program that moves and speaks	Auto-mation	
	• My own story scene	Function, Control	
	• Program asking for a name and saying hello	Auto-mation	

Text-book	Main activities	ratio	Ratio (%, Computing / Total)
	• Programming calculator	Auto-mation	
	• Creating a two-word link program	Auto-mation	
	• Programming robot Cleaner	Auto-mation	
	• Creating and sharing my own program	Function, Control	
E	• Delivery of preparations using drones	Control	71 (5/7)
	• Exploring the route of the amusement park	General	
	• Creating a program to introduces me	Function	
	• Creating a program with input, and output	Control, Data	
	• Cushion Quiz game	General	
	• Cushion Quiz game Programming	Control, Data	
	• Problem solving programming	Function, Control, Data	
F	• Field trips planning	General	22 (2/9)
	• Solving classroom problems	General	
	• Stacking paper cups	Control	
	• Basic programming experience	Auto-mation	
	• Numeric input/output Program	Auto-mation	
	• Character input/output Program	Auto-mation	
	• Multiple character input/output program	Auto-mation	
	• Sequence·Selection·Loop Program	Auto-mation	
	• programming to solve problems in everyday life	Control	

5. 논의 및 결론

본 연구에서는 초등 실과 교과서의 소프트웨어교육 영역에 나타난 컴퓨팅 사고력 요소와 교과서 학습활동별 컴퓨팅 비율을 분석하였다.

첫째, 성취기준에 따라 교과서별 컴퓨팅 사고력 요소를 분석한 결과 교과서별로 컴퓨팅 사고력의 하위 요소 포함 여부의 차이가 나타났다. 이는 숫자, 문자의 입·출력 과정, 추상화된 알고리즘 등을 직접적으로 제시한 것으로 보인다. 그러나, 프로그래밍하는 과정에서 컴퓨팅 사고력이 증진되기 위해서는 프로그래밍 전에 이루어지는 자료수집, 자료분석, 자료표현, 자료분해, 추상화, 알고리즘 설계 등의 단계가 필수적이며, 다수의 연구자가 자동화 전 이루어지는 추상화의 중요성에 대해 언급하고 있는 만큼[4][5], 추후 교과서 수정 및 보완시 자동화로 이루어지기 위한 추상화 과정을 학습 활동에서 이끌어낼 수 있도록 교과서 내용을 구성할 필요가 있다.

둘째, 학습 내용의 컴퓨팅 수준 분석 결과, 컴퓨팅과 연결된 추상화 활동을 제시하는 교과서의 비율이 비교적 낮게 나타났다. 일부 교과서에서는 일상생활에서의 절차적 사고를 활용한 문제 해결 활동만 제시하였다. 또한, 일부 교과서에서는 문제탐색, 문제표현, 추상화가 유기적으로 연결되지 않고 단순한 CS적인 요소가 포함된 활동만 제시하는 사례도 나타났다. 이는 김수환(2018)의 논의와 같이 학생들이 컴퓨팅의 추상화로 연결되지 못하고 일반적인 추상화에서 그칠 가능성이 있으며, 이는 컴퓨팅 사고력에서 의미하는 추상화에 대해 오개념을 형성할 가능성이 있다. 컴퓨팅 사고력에서 이루어지는 추상화와 자동화가 유기적으로 이루어지기 위해서는 기능 추상화, 컨트롤 추상화, 데이터 추상화가 포함되어야 하며, 일상생활의 사례를 컴퓨팅의 사례로 그 범위와 내용을 심화되도록 나선형으로 이끌어낼 필요가 있다.

셋째, 소프트웨어의 의미와 우리 생활에 미치는 영향은 대체로 교과서와 학습 자료, 영상 자료 등을 활용하는 것으로 보인다. 절차적 사고를 이해하고 일상생활 속에서 절차적 사고로 과제를 해결하는 활동은 교과서별로 시수의 차이가 있으며, 이는 절차적 사고의 중요성을 바라보는 저자의 시각이 드러난 것으로 보인다. 일부 교과서에는 정렬망이나 다익스트라 알고리즘과 같이 컴퓨터 과학의 원리를 제시한 활동도 나타났다. 추후 교과서 수정 및 보완하는 경우 교육과정에서 제시하는 절차적 사고인 알고리즘의 특징, 알고리즘의 표현 방법으로 활동을 구성할 필요가 있다고 보인다.

추후 교육과정 개편 또는 교과서 개정시 다음과 같은 교육 내용 및 방법을 포함할 필요가 있다. 첫째, 문제 발견과 이해부터 자동화까지 컴퓨팅 사고의 일련의 과정들이 이어질 수 있도록 차시를 편성할 필요가 있다. 둘째, 일상생활의 요소를 사용하되, 자료 추상화, 기능 추상화 등 컴퓨팅이 포함된 추상화 활동을 내포할 필요가 있다. 셋째, 컴퓨터 과학의 원리를 포함할 경우, 컴퓨팅 사고력 과정 속에서 자연스럽게 제시할 필요가 있다. 향후 컴퓨팅 사고력 요소와 과정이 균형있게 제시되도록 보완하고, 일반적인 추상화 활동과 자동화로 이어질

수 있는 추상화 활동이 함께 포함되도록 보완한다면 2015 개정 교육과정 소프트웨어 교육 목표에 부합하는 '컴퓨팅 사고력을 지닌 창의·인재 양성'을 위한 효과적인 교과서로서 자리매김할 수 있을 것이다.

참고문헌

[1] An, S. J. & Lee, Y. J. (2016). Designing Programming Curriculum for Developing Programming Pedagogical Content Knowledge of Pre-service Informatics Teachers. *The Journal of Korean Association of Computer Education, 19*(2), 1-10.

[2] Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. *Acm Inroads, 2*(1), 48-54.

[3] Google for Education. (2015). Exploring Computational Thinking, retrieved from https://edu.google.com/resources/programs/exploring-computational-thinking/

[4] Jeong, I. K. (2016). Review of Concept of Abstraction of Computational Thinking. *Journal of The Korean Association of Information Education, 20*(6), 585-596.

[5] Ju. Y. J. & Ma, D. S. (2018). The Development of Abstractable Competency Assessment Standards for the Measurements of Computational Thinking. *Journal of The Korean Association of Information Education, 22*(3), 375-383.

[6] Jung, S. B. et al.(2019). Practical Arts Textbook for elementary School grades. Seoul: Kyohak

[7] Kim B. S. (2014). Programming education program based on PPS to improve computational thinking ability, doctoral dissertation. Jeju National University of Education.

[8] Kim, J. R. (2019). Analysis of textbook contents according to the 2015 revised Elementary Software Education Achievement Standards. *Journal of The Korean Association of Information Education, 23*(1), 9-18

[9] Kim, M. N. & Park, S. J. (2019). The Analysis of 'Software Education' Unit in the Practical Arts Textbooks According to 2015 Revised Curriculum. *Journal of The Korean Association of Information Education, 23*(3), 255-264.

[10] Kim, S. H. (2018). Analysis of Abstraction Contents in Informatics Textbooks of Middle School According to 2015 Revised Curriculum. *The Journal of Korean Association of Computer Education, 21*(5), 1-10.

[11] Kim, Y. M. (2019). Artificial Intelligence (AI) Manpower Training Policies and Implications in Major Countries. Health Industry Brief, 276, 1-20

[12] Lawshe, C.H. (1975). A quantitative approach to content validity. *Personnel psychology, 28*, 563-575.

[13] Lee, C. H. et al.(2019). Practical Arts Textbook for elementary School grades. Seoul: MiraeN.

[14] Lee, C. S. et al.(2019). Practical Arts Textbook for elementary School grades. Seoul: CHUNJAE.

[15] Ministry of Education (2015). Practical Arts (Technology and Home)/Informatics Education Curriculum, Ministry of Education Notice No. 2015-74, separate volume 10. Ministry of Education.

[16] Ministry of Education(2015). 2015 software education guidelines

[17] Ministry of Education. (2015). Elementary and secondary school curriculum, Ministry of Education Notice No. 2015-74, separate volume 1. Ministry of Education.

[18] Ryu, C. S. et al.(2019). Practical Arts Textbook for elementary School grades. Seoul: Kumsung.

[19] Seo, W. S. et al.(2019). Practical Arts Textbook for elementary School grades. Seoul: Donga.

[20] Song, H. S. et al.(2019). Practical Arts Textbook for elementary School grades. Seoul: Visang.

[21] Wing, J. M. (2006). *Computational thinking. Communications of the ACM, 49*(3), 33-35.

[22] Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: *Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725.

저 자 소 개

김 정 랑

1997 전남대학교 (이학박사)
1999 San Jose State University
객원교수
1985~현재 광주교육대학교
컴퓨터교육과 교수

관심분야 : 컴퓨터교육, 디지털교과서, 이러닝, 교육정보화, 스마트교육, 소프트웨어교육, 인공지능교육
e-mail : jrkim@gnue.ac.kr