

소프트웨어 교육 활성화를 위한 초등 소프트웨어 융합 교육 지도의 실제

구재훈[†] · 김태영[†]

[†] 한국교원대학교 컴퓨터교육과

A Study on the Practical Teaching of Elementary Software Convergence Education for Activating Software Education

Jaehoon Ku[†] · Taeyoung Kim[†]

[†] Dept. of Computer Education, Korea National University of Education

요 약

오늘날 미국과 영국 등을 포함한 세계 주요국가에서는 미래사회에 경쟁력을 기르기 위한 방법으로 소프트웨어 교육을 내세우며 컴퓨팅 사고력 기반 문제 해결을 위한 소프트웨어 교육을 정규 교육과정에 포함 시키고 있다. 이에 우리나라도 교육부에서 '2015 개정 교육과정'에 소프트웨어 교육을 포함시키고, 초등 소프트웨어 교육의 목표를 '컴퓨팅 사고력과 건전한 정보윤리의식을 바탕으로 알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 이해하여 컴퓨팅 사고력을 가진 창의·융합 인재를 기르는 데에 목표가 있다' 라고 발표하였다. 또한 소프트웨어 교육의 방향과 운영에 필요한 내용들 중에 교육 현장에서 소프트웨어 교육을 하기에 부족한 초등 소프트웨어 교육 시수의 적절한 확보와 교수·학습 방법의 연구가 필요한 것으로 발표하였다.

이에 소프트웨어 교육 목표에 부합하고, 초등학교 현장에서 부족한 소프트웨어 교육 시수 해소와 소프트웨어 교육 활성화를 위한 초등 소프트웨어 융합 교육 지도의 실재를 연구해 보기로 하였다.

1. 서 론

오늘날 전세계는 정보교육 관련 패러다임의 변화에 따라 소프트웨어 교육이 크게 확산되고 있다. 미국과 영국 등을 포함한 세계 주요국가에서는 미래사회에 경쟁력을 기르기 위한 방법으로 소프트웨어 교육을 내세우며 컴퓨팅 사고력 기반 문제 해결을 위한 소프트웨어 교육을 정규 교육과정에 포함 시키고 있다[1][2].

전세계적인 정보교육의 패러다임 변화에 따라 우리나라도 교육부에서 '2015 개정 교육과정'에 소프트웨어 교육을 포함시키고, 초등 소프트웨어 교육의 목표를 '컴퓨팅 사고력과 건전한 정보윤리의식을 바탕으로 알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 이해하여 컴퓨팅 사고력(Computational Thinking, 이하 CT)을 가진 창의·융합 인재를 기르는 데에 목표가 있다' 라고 발표하였다. 또한 소프트웨어 교육의 방향과 운영에 필요한 내용들 중에 교육 현장에서 소프트웨어 교육을 하기에 부족한 초등 소프트웨어 교육 시수의 적절한 확보와 교수·학습 방법의 연구가 필요한 것으로 발표하였다[3][4].

이에 소프트웨어 교육 목표에 부합하고, 초등학교 현장에서 부족한 소프트웨어 교육 시수의 해소와 소프트

웨어 교육 활성화를 위한 초등 소프트웨어 융합 교육 지도의 실제 부분을 교수·학습 과정안을 통해 제시하고자 한다.

2. 이론적 배경

2.1 컴퓨팅 사고력(Computational Thinking)

컴퓨팅 사고력이란 용어는 2006년 카네기 멜론 대학의 Jennette Wing(2006)에 의해 정립 되었는데 Wing은 '컴퓨터 과학의 기초 개념을 토대로 문제해결 시스템 설계, 인간 행동의 이해를 내포 한다' 고 하였으며 아이들의 분석 능력을 키우기 위해 컴퓨팅 사고력을 읽고, 쓰고, 선택하기에 추가해야 한다' 고 하였다[5][6].

Wing의 주장 이후 컴퓨팅 사고력에 대한 다양한 논의가 이뤄졌으며, 2009년에 National Research Council의 주도로 이뤄진 워크샵에서 컴퓨팅 사고력에 대해 여러 학자들이 모여 다양한 정의를 내렸다[5][6][7][8][9].

이 워크샵에서 이뤄진 다양한 컴퓨팅 사고력에 대한

정의를 살펴보면 <표 1>과 같다.

<표 1> 컴퓨팅 사고력에 대한 다양한 정의[10],[11]

이름	정의
David Moursund	Saymour Papert가 그의 저서 'Mindstorms'에서 언급하였던 절차적 사고와 연관되어 있는 개념이다. 절차적 사고는 개발(developing), 표현(representing), 시험(testing), 디버깅(debugging)의 순서로 구성되어 있고, 효과적인 절차를 만들기 위해서는 컴퓨터 등의 특별한 장치로 수행 가능한 형태인 구체적인 단계별 명령이 필요하다.
Peter Lee	CT는 인간 지능을 확대하여 실제적인 적용을 할 수 있는 인간 지능의 메커니즘에 관한 연구이다. 즉, 인간의 정신적인 능력의 복잡도를 관리하거나 일을 자동적으로 처리하도록 하는 추상화 도구를 통해 확장하는 것이라 할 수 있다.
Bill Wulf	과학은 물리적인 대상(physical object)에 관련된 것이고, CT는 어떠한 문제를 해결하는 과정과 그 과정의 진행을 가능하게 하는 추상적인 현상들에 초점을 맞추고 있다.
Don brahamson	CT는 형식지를 설명하고, 암묵지를 객관화하고, 이러한 지식을 컴퓨터적인 형태로 퍼뜨리고, 이러한 활동에서 발생한 결과물을 관리하는 컴퓨터 관련 기호 체계의 사용이다.
Gerald Sussman	CT는 일을 처리하는 정확한 방법을 공식화하는 방법이다. 다시 말해 특정한 문제를 효율적으로 처리하기 위해 그 문제를 철저하게 분석하고 해결하기 위한 엄밀한 절차를 만드는 과정이 CT인 것이다.
Edward Fox	문제해결을 목적으로 눈에 보이지 않는 추상화된 개념을 다루거나 조작하는 것이 CT의 핵심이다. CT는 인간이 세상에 접근하고, 과정들을 생각하고, 디지털로 표현된 것들을 다룰 때 하는 일이라고 표현할 수 있다.
Robert Constable	CT는 특정한 기술이나 사고 과정들의 집합이 아니라 개방적이고 점점 발전하는 기술의 역동적인 본질에 관련된 개념이다. 현재까지 나타난 CT를 대략적으로 기술한다면 지적인 과정을 자동화하는 것과 정보 처리 과정을 연구하는 것 정도가 될 것이다.

우리나라에서는 'Computational Thinking'의 한국어 화 표기로 '계산적 사고력'나 '컴퓨터적 사고력' 등 다양한 표기로 사용하였으나, 한국과학창의재단 및 전문가들의 협의에 의해 '컴퓨팅 사고력'으로 제시 하였고, 컴퓨팅 사고력을 '컴퓨팅 시스템의 역량을 활용하여 해결하고자 하는 문제를 효율적으로 해결할 수 있는 절차적 사고 능력' 이라고 정의 하였다. 이에 교육부가 '2015 개정 교육과정'에서도 '컴퓨팅 사고력'으로 사용 하였다[5][11].

2.2 컴퓨팅 사고력(CT) 구성 요소

<표 2> 컴퓨팅 사고력(CT) 구성 요소[12],[13]

CT 구성 요소	개념 설명
자료 수집 Data Collection	문제의 이해와 분석을 토대로 문제를 해결하기 위한 자료를 모으는 단계
자료 분석 Data Analysis	수집된 자료와 문제에 주어진 자료를 세심히 분류하고 분석하는 단계
자료 표현 Data Representation	문제의 자료 내용을 그래프, 차트, 단어, 이미지 등으로 표현하는 단계
문제 분해 Problem Decomposition	문제를 해결해나가기 위해 문제를 나누어 분석하는 단계
추상화 Abstraction	문제의 복잡도를 줄이기 위해 해 기본 주요 개념의 정의를 설정하는 단계
알고리즘과 절차 Algorithms & procedures	지금까지 문제를 해결하기 위한 과정을 순서적 단계로 표현하는 단계
자동화 Automation	순서적으로 나열하고 표현한 내용을 컴퓨팅 기기를 이용하여 해결과정의 최선책을 선택하는 단계
시뮬레이션 Simulation	복잡하고 어려운 해결책이나 현실적으로 실행이 불가능한 해결책을 선택하기 위해 모의 실험하는 단계
병렬화 Parallelization	문제를 해결하기 위한 공동의 목표를 달성하기 위한 작업을 수행하는 단계

<표 2>는 CSTA & ISTE(2011)에 따른 컴퓨팅 사고력(CT)의 주요 구성 요소에 대한 개념 설명을 간추린 것이다[13].

이 9가지의 요소들은 문제를 해결하는 데 있어 특정 위계나 절차를 갖고 있는 것은 아니며, 모든 요소들이 반드시 포함되어야 하는 것도 아니라고 한다[7].

2.3 소프트웨어(SW) 교육의 인재상과 교육 목표

SW 교육 목표 중에 초등 SW 교육 목표는 컴퓨팅 사고력과 건전한 정보윤리의식을 바탕으로 알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 이해하여 컴퓨팅 사고력을 가진 창의·융합 인재를 기르는데에 목표가 있다.

SW 교육은 지식 위주의 교육보다는 수행 위주의 교육을 통하여 디지털 사회의 필수적 요소인 컴퓨팅 사고력의 의미와 중요성을 학습자 스스로 인식하고 그 가치를 확인할 수 있도록 교육 방법을 설계한다[4][5].

2.4 시수 확보 및 연계 운영 방안

교육부의 ‘2015 개정교육과정’ 중에 실과 교과와 소프트웨어 교육과 관련하여 교수 학습 방법 및 유의 사항에 ‘컴퓨팅 사고는 소프트웨어 교육에 국한 되는 것이 아니므로 국어, 사회, 수학, 과학 등 다양한 교과에서도 반영하여 지도 한다’라고 제시 되었다.

또한 실과 교과와 창의적 체험 활동 등을 통해 최소 연간 17시간 이상을 확보하여 소프트웨어 교육을 실시하도록 되어 있다[3][4].

교육부에서 실과 교과와 창의적 체험 활동에서 ‘자율 활동’ 영역의 재구성을 예시로 제시하였으나, 중요한 것은 교사가 교과 간의 융합에 대한 많은 관심을 통해 소프트웨어 교육과 여러 교과 간에 융합이 잘 될 수 있도록 교육과정을 재구성하는 연구와 노력이 필요하다.

3. 지도의 실제

3.1 관련 교과 교육과정 분석 및 학습 내용 추출

초등 SW 융합 교육 학습 내용 구성을 위해 2009 개정 초등 교육과정에서 관련 수학 교과와 단원을 분석하였다.

<표 3>과 같이 6학년 1학기 수학과 4단원에서 관련 학습 내용을 추출 하였다.

<표 3> 초등 SW 융합 교육 관련 교육과정 학습 내용 추출

교과	학년-학기-단원	차시	학습 목표 및 학습 내용
수학	6-1-4. 비와 비율	6	<ul style="list-style-type: none"> • 학습목표 <ul style="list-style-type: none"> - 백분율의 뜻을 알고, 분수와 소수를 백분율로 나타낼 수 있다. • 학습 내용 <ul style="list-style-type: none"> - 백분율의 뜻을 알아보게 한다. - 비율을 백분율로 나타내는 방법을 알아보게 한다.

본 연구에서는 <표 3>의 4단원 전체 내용 중 6차시 학습 내용과 소프트웨어 교육 학습 내용을 융합하여 재구성 하였다. <표 4>는 초등 SW 융합 교육 학습의 학습 내용으로 재구성한 표이다.

또한 초등 소프트웨어 교육에서 컴퓨팅 사고력(CT) 교육에 효과적으로 알려진 미국 MIT대학에서 개발한 무료 소프트웨어 프로그램인 스크래치(Scratch)를 사용하기로 하였다[14].

<표 4> 관련 교육과정 분석 및 학습 내용 재구성

교과	학년-학기-단원	차시	학습 목표 및 학습 내용 재구성
수학	6-1-4. 비와 비율	6	<ul style="list-style-type: none"> • 학습목표 <ul style="list-style-type: none"> - 백분율의 뜻을 알고, 분수와 소수를 백분율로 나타낼 수 있다. • 학습 내용 <ul style="list-style-type: none"> - 과제로 제시된 백분율 관련 수업 동영상 내용을 확인해 본다. - 학습지를 통해 보충 및 심화 학습을 한다. - 스크래치 프로그램으로 백분율 구하기 프로그램 만들고 발표해 본다.

3.2 지도의 실제

<표 5>는 초등 소프트웨어 융합 교육 지도의 실제로 지금까지 재구성한 내용을 바탕으로 교수·학습 과정을 작성한 내용이다.

<표 5> 초등 소프트웨어 융합 교육 교수·학습 과정안

수업 일시	2017. 00. 00. (0요일) 0교시	대상	6-0 (00명)	장소	6-0 교실	
단원	4. 비와 비율		차시	6/15	지도 교사	0 0 0
학습 목표	백분율의 뜻을 알고, 백분율로 나타낼 수 있다.			소수와 소수를	교과서 108 ~109쪽	
수업 모형	일반 학습 모형					
활용 자료	컴퓨터, 광고 동영상, 학습지, 윈도우패드, PPT					
학습 과정 (시간)	교수· 학습 활동					
도입 (5분)	<ul style="list-style-type: none"> ● 동기 유발 <ul style="list-style-type: none"> ○ 재미있는 광고 동영상을 보여주고, 동영상에서 공통으로 나오는 것을 발표하기 ● 학습 문제 확인 <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">백분율의 뜻을 알고, 백분율로 나타내 봅시다.</div> 					
전개 (32분)	<ul style="list-style-type: none"> ● 학습 활동 안내 <ul style="list-style-type: none"> <활동 1> 확인해 볼까요? <활동 2> 해결해 볼까요? <활동 3> 발표해 볼까요? ● <활동 1> 확인해 볼까요? <ul style="list-style-type: none"> ○ 과제로 제시한 백분율 관련 수업 동영상에 대해 어느 정도 이해를 하였는지 발표 및 기본 학습지로 확인하기 ● <활동 2> 해결해 볼까요? <ul style="list-style-type: none"> ○ 기본 학습지로 확인한 내용을 통해 보충 학습과 심화 학습 문제를 해결하기 ○ 학습지를 해결한 학생은 스크래치 프로그램으로 문제를 해결하기 ● <활동 3> 발표해 볼까요? <ul style="list-style-type: none"> ○ 스크래치 프로그램으로 해결한 내용을 발표한다. 					
정리 (3분)	<ul style="list-style-type: none"> ● 학습 내용 정리하기 <ul style="list-style-type: none"> ○ 배운 내용 정리하기 ● 차시 예고하기 <ul style="list-style-type: none"> ○ 사건이 일어날 가능성을 알아봅시다. 					

4. 결론 및 논의

지금까지 소프트웨어 교육 활성화를 위한 초등 소프트웨어 융합 교육 지도의 실제에 대해 연구하였다.

이미 언급 되어진 소프트웨어 교육 활성화를 위해 초등 소프트웨어 교육의 부족한 시수를 확보하는 방법 중에 하나는 교사가 교과 간의 융합에 대한 관심을 통해 소프트웨어 교육과 여러 교과 간에 융합이 잘 될 수 있도록 교육과정을 재구성하는 연구와 노력이 필요할 것이다.

이에 앞으로 다양한 교과와 연계한 효과적인 초등 소프트웨어 융합 교육이 이루어지도록 다양한 교수·학습 방법에 대해 더 연구해 볼 것이다.

마지막으로 본 연구를 계기로 일선 초등학교 현장에서 소프트웨어 교육의 활성화 방안과 초등 소프트웨어 융합 교육의 연구가 좀 더 활발히 이루어지는 촉매제가 되기를 기대해 본다.

참 고 문 헌

- [1] 오경선, 안성진(2016). **소프트웨어 교육을 위한 컴퓨팅사고 교육내용 설계 기본 연구**. 한국컴퓨터교육학회 논문지 제19권 제2호.
- [2] 박효민(2014). **글로벌 소프트웨어 교육 현황 및 교육 도구 동향**. KISA.
- [3] 교육부(2015). **2015 개정 교육과정**.
- [4] 교육부(2015). **SW 교육 운영 지침**.
- [5] 구재훈, 김태영(2015). **컴퓨팅 사고력 기반 창의·융합 인재 육성을 위한 초등 소프트웨어 융합 교육 지도의 실제 연구**. 한국교원대학교 정보교육연구소 논문집 제11권 제1호.
- [6] 최무영(2015). **초등학생 컴퓨팅 사고력(CT) 능력 신장을 위한 찾아가는 융합인재교육(STEAM) 프로그램 개발**. 경인교육대학교 석사학위논문.
- [7] 최정원(2015). **정보 영재의 컴퓨팅 사고력 향상을 위한 퍼즐 기반 알고리즘 학습 모형**. 한국교원대학교 박사학위논문.
- [8] Wing, J. M. (2006). **Computational Thinking**. Communications of the ACM, 49(3), 33-35.
- [9] 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.
- [10] National Research Council. (2009). **Report of a Workshop on The Scope and Nature of Computational Thinking**. National Academies Press.
- [11] 한국과학창의재단(2014). **초·중등 단계 Computational Thinking 도입을 위한 기초 연구**.
- [12] Computer Science Teachers Association (CSTA) and the International Society for Technology in Education(ISTE) (2011).
- [13] 김순화외 2명(2015). **컴퓨팅 사고력 기반 융합인재교육 프로그램의 효과성 분석 연구**. 한국컴퓨터교육학회 논문지 제18권 제3호.
- [14] Scratch(스크래치), <https://scratch.mit.edu/>.