

Development of a Software Education Curriculum for Secondary Schools

Seong-Won Kim*, Youngjun Lee**

Abstract

With more emphasis on importance of software, many countries try to provide software education. Of course Korea includes informatics courses in 2015 revised curriculum, so that software education will be administered briskly in soon. However there are practical challenges including a lack of teaching hour in classes and the monotony of educational contents which occurs with that. To solve these problems, this research develop software education curriculum model that could be practically used for both middle and high school. First this study compare the curriculum of Korea to that of United States and United Kingdom. After analyzing the result, the curriculum model for middle and high school is developed. The curriculum model can be classified into three types, middle, high and advanced-high levels and include key concepts like collaboration and convergence, computational thinking, computing practice and programming, computers and communications devices, community, global, and ethical impacts. To assess the feasibility of our software education curriculum model, examination was made by expert group and a hearing was held by related researchers. Then the model was modified in a way that adjustable to Korea education system. This study provides some important guidances on designing a curriculum for software education at middle and high school. However, there still are difficulty adjusting to the elementary school and university course. To be able to further research, same kind of studies on elementary school and university course need to be done. Also, continuous modifications are required to reflect reality including technological advance, curriculum, and changes of education system.

▶ Keyword : Software education, Curriculum development, Secondary schools, Software education standards.

I. Introduction

2016년에 열린 다포스 포럼에서는 4차 산업혁명 (The Fourth Industrial Revolution)에 접어들면서 영역의 경계가 없이 기술의 융합이 발생할 것이고, 경험하지 못한 속도로 기술이 발전할 것이라고 말하였다. 이에 따라 인공지능, 로봇, 사물인터넷과 같은 기술을 중심으로 사회의 변화가 진행될 것으로 예측하였다 [1]. 따라서 현대 사회에서는 이러한 기술들의 핵심으로 활용되고 있는 소프트웨어의 중요성이 증가할 것이다. 이와 같이 소프트웨어의 중요성이 커짐에 따라 소프트웨어를 활

용할 수 있는 능력뿐만 아니라 학생들이 직접 설계하고 제작할 수 있는 능력을 기를 수 있는 교육의 필요성이 증가하였다 [2, 3, 4, 5].

소프트웨어를 중심으로 한 사회의 변화에 따라서 전 세계적으로 실시되고 있는 컴퓨터과학 교육은 큰 변화가 일어나고 있다. 미국은 Computer Science Teachers Association (CSTA)가 수준에 따라 학생들이 배워야하는 컴퓨터과학 교육의 표준을 제안하기 위하여 'CSTA K-12 Computer Science

• First Author: Seong-Won Kim, Corresponding Author: Youngjun Lee

*Seong-Won Kim (sos284809@gmail.com), Dept. of Computer Education, Korea National University of Education

**Youngjun Lee (yjlee@knu.ac.kr), Dept. of Computer Education, Korea National University of Education

• Received: 2016. 06. 02, Revised: 2016. 06. 28, Accepted: 2016. 07. 26.

• This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2013R1A2A2A03068459).

• This paper is a revised and expanded version of a paper entitled 'Development of the Software Education Curriculum Model for Secondary Education in Korea' presented at the EDULEARN 2016 conference.

Standards'를 개발하였다 [6]. 또한, CSTA, Association for Computing Machinery(ACM), CSTA, National Math and Science Initiative(NMSI), Cyber Innovation Center(CIC), Code.org의 주도로 미국의 주별로 컴퓨터과학을 도입할 수 있도록 'A Framework for K-12 Computer Science Education'을 개발하고 있다 [7]. 이러한 Framework는 미국의 주에 따라 교육과정 모델과 표준을 개발할 수 있도록 Practice와 Concept을 제시하고 있다. 영국은 2014년 교육과정 개정을 통하여 교과목 명을 'Information Communication and Technology (ICT)'에서 'Computing'으로 바꾸고, 교과 내용도 컴퓨팅 사고력과 컴퓨터과학을 중심으로 변경하였다. 또한, 다양한 기업과 비영리단체의 지원을 바탕으로 학생들에게 소프트웨어 교육을 도입하기 위하여 노력하고 있다 [8]. 그뿐만 아니라 일본은 2020년부터 프로그래밍 교육을 도입하겠다고 발표를 하였으며 [9], 핀란드는 2016년부터 프로그래밍 교육을 도입하기 위하여 교육과정 개정과 교사 양성을 실시하고 있다 [10]. 인도도 컴퓨터 과학 교육을 위한 표준을 제시하고, ICT와 프로그래밍 내용이 통합된 교육과정을 개발하였으며, 학교 현장에서 소프트웨어 교육을 도입하기 위해 노력하고 있다 [11].

한국도 이러한 흐름에 발맞추어 2015년에 교육과정 개정이 이루어졌고, 2018년부터 새로운 교육과정이 도입될 예정이다. 개정된 교육과정에서는 소프트웨어 인재 양성을 위하여 기존의 선택 교과였던 '정보' 교과를 초·중학교에 필수 교과로 지정하고, 컴퓨팅 사고력과 정보 문화 소양, 협력적 문제 해결을 중심으로 교육과정 개편이 이루어졌다 [12, 13]. 교육과정의 개정이 이루어졌지만, 필수 교과로 배정된 시수가 초등학교에서는 17차시, 중학교에서는 34차시에 불과하여 컴퓨팅 사고력을 배양하기 위해서는 시수가 부족하다는 의견이 제기되었다 [14]. 또한, 시수 부족으로 교육과정에서 많은 내용이 빠지게 되었다. 이러한 시수와 교육과정의 문제 때문에 정보 교과 교육의 목적인 창의적으로 문제를 해결할 수 있는 소프트웨어 인재 양성이 어려움에 봉착할 수 있다. 이러한 어려움에도 학교 현장에서 원활하고 안정적인 소프트웨어 교육을 도입하기 위해서는 중·고등학교 소프트웨어 교육의 기준이 되는 소프트웨어 교육 과정의 필요성이 요구된다.

따라서 본 연구에서는 중·고등학교에서 시행되어야 하는 소프트웨어 교육의 표준을 제시하기 위한 소프트웨어 교육과정을 개발하고자 한다. 소프트웨어 교육과정을 개발하기 위하여 여러 나라의 교육과정을 분석하여 연구에 대한 시사점을 얻고, 시사점을 바탕으로 한국의 교육 상황에 맞는 소프트웨어 교육과정을 개발하였다. 본 논문은 국내·외 소프트웨어 관련 교육과정을 소개하고, 연구에 대한 방법에 대한 내용을 서술하였다. 그 다음으로 영국, 미국, 한국의 소프트웨어 교육과정을 비교·분석한 내용을 제시하고, 이를 기반으로 개발한 소프트웨어 교육과정을 제안하였다. 마지막으로 연구에 대한 결론을 도출하였다.

II. Korea and foreign country curriculum

1. 미국의 교육과정: CSTA K-12 Computer Science Standards

미국은 한국과 다르게 50개의 주마다 마련한 기준안에 따라서 교육을 실시하고 있다. CSTA는 컴퓨팅이 직업과 일자리를 늘려주고 과학과 통합되어 새로운 발견을 하는데 주도적인 역할을 하고 있음에도 불구하고, 컴퓨터과학 교육이 부실하게 이루어짐을 지적하였다. 학교 현장에서는 70% 이상의 중등 학교에 컴퓨터과학 교육을 위한 표준이 없는 상태이고, 컴퓨터과학 교육에서 중요한 시기인 유아와 초등학교에서 컴퓨터 과학 교육이 거의 이루어지지 않고 있다는 것을 지적하였다. 교사 교육도 컴퓨터과학 교육 표준이 없음에 따라 학생들에게 가르쳐야 할 내용과 능력에 대한 교육이 제대로 이루어지지 않는다고 말하였다. 이러한 문제를 해결하기 위하여 CSTA와 ACM은 컴퓨터과학에 대한 내용을 배우고, 컴퓨터과학적 소양과 능력을 강화할 수 있도록 'CSTA K-12 Computer Science Standards'이라는 컴퓨터과학 교육과정 모델을 개발하였다.

'CSTA K-12 Computer Science Standards'는 학생들이 컴퓨터과학의 목적과 필요성에 대해 설명하고 있으며, 학생들의 나이에 따라 level을 나누어 가르치고자 하는 컴퓨터 과학 내용을 제시하고 있다. 학생들의 수준에 따른 컴퓨터 과학 내용을 살펴보면, Level 1 (K-6, 초등학교 1-6학년)은 학생들이 컴퓨팅 사고력에 대한 호기심을 유발하고, 컴퓨팅 사고력에 대한 중요성을 인식하기 위한 내용을 포함하고 있다. Level 2 (6-9, 중학교 1-3학년)는 학생들이 컴퓨팅 사고력을 활용하여 문제를 해결하는 내용을 포함하고 있는데 단순히 컴퓨터 과학의 문제뿐만 아니라 타 교과와의 연계를 통한 문제 해결도 포함하고 있다. Level 3 (9-12, 고등학교 1-3학년)은 세부 level이 3개 포함되어 있다. Level 3A (9-10)는 컴퓨터과학과 관련된 진로, 가치, 사회적 윤리에 대한 이해를 이해하는 내용을 포함하고 있고, level 3B (10-11)는 컴퓨터과학이 타 영역에 미치는 영향에 대해 이해하고, 컴퓨터 과학 기술을 활용한 문제 해결에 대한 내용을 심도 있게 이해할 수 있는 내용으로 구성되어 있다. 마지막으로 level 3C는 컴퓨팅에서 특정 분야를 선택하여 집중적으로 학습하는 내용이다.

CSTA K-12 Computer Science Standards의 컴퓨터과학 내용은 5개의 핵심 영역 (Strand)을 중심으로 제시되어 있다. 첫 번째 영역으로 컴퓨팅 사고력 (Computational thinking)은 문제를 해결하기 위한 추상화와 데이터 표현, 알고리즘, 모델링과 시뮬레이션 내용과 다른 영역으로 확장에 대한 내용을 포함하고 있다. 협력 (Collaboration)은 컴퓨터과학이 본질적으로 협력적인 교과이므로 협력적으로 문제를 해결하는 과정을 경험하는 내용이 제시되어 있다. 컴퓨팅 연습과 프로그래밍 (Computing practice and programming)은 문제 해결을 위하

여 데이터를 수집하고, 프로그래밍과 다양한 테크놀로지의 활용을 통하여 문제 해결을 경험하는 내용과 컴퓨터 과학과 관련된 진로와 보안에 대한 내용으로 구성되어 있다. 컴퓨터와 커뮤니케이션 장비 (Computers and communications devices)는 컴퓨터와 네트워크의 구성 요소와 작동 원리, 인공지능 내용이 포함되어 있다. 마지막으로 커뮤니티와 세계화, 윤리적 영향력 (Community, global, and ethical impacts)은 학생들이 정보윤리를 갖춘 시민이 되기 위하여 개인정보 보호 원칙, 네트워크 보안, 저작권, 정보의 평가와 같은 내용을 배우는 영역이다 [6, 15].

2. 영국의 교육과정: The national curriculum in England: Computing

영국은 컴퓨터 교과를 1991년부터 가르치기 시작하였으며 처음에는 'Technology' 교과에 포함되어 있었다. 2000년에 'Technology' 교과에서 벗어나 'ICT' 교과로 분리되어 교육이 시작되었다. 하지만 과학·기술의 발전에 따라 현대 사회에서 요구하는 디지털 리터러시 (Digital literacy)의 변화가 진행되었다. 영국은 이러한 변화를 교육과정에 반영하여 컴퓨터 교과의 과목명을 'ICT'에서 'Computing'으로 바꾸고, 컴퓨팅 사고력을 중심으로 디지털 리터러시, ICT, 컴퓨터과학 내용으로 교과 내용을 개편하였다.

영국의 교육과정은 영유아 보육과정 (Childcare)과 초등 교육과정 (Primary school), 중등 교육과정 (Secondary school)으로 나누어져 있는데, 'Computing'은 초·중등 교육과정에서 필수로 가르치도록 편성되어 있다.

영국의 'Computing' 교과는 key process를 기반으로 key stage와 key concept로 구성되어 있다. Key stage는 초등 교육과정이 key stage 1과 2, 중등 교육과정이 key stage 3, 4이다. 또한, 'Computing' 교과에서 key concept은 언어, 기계, 그리고 계산 (Languages, machines, and computation), 데이터와 표현 (Data and representation), 통신과 조정 (Communication and coordination), 추상화와 설계 (Abstraction and design), 컴퓨터와 컴퓨팅 (Computers and computing are part of a wider context)으로 구성되어 있으며, key process는 모델링과 분해, 일반화를 포함한 추상화 (Abstraction: modelling, decomposing, and generalizing)와 프로그래밍 (Programming)으로 구성되어 있다. 영국의 'Computing' 교육 과정은 이와 같은 key stage에 따라 학생들이 알고 있어야 하는 내용을 제시하고 있다 [8, 16].

3. 한국의 교육과정: 2015 개정 교육과정

한국의 컴퓨터 교육은 초기에 직업 교육의 목적으로 시작되었다. 5차 교육과정부터 초등학교의 실과와 중·고등학교의 기술 교과에 컴퓨터 내용이 포함되기 시작하였다. 이러한 교육은 컴퓨터가 사회에 본격적으로 보급됨에 따라 미래 사회에 대비하기 위하여 컴퓨터 교육을 학생들에게 실시하였다. 6차 교육

과정에서는 최초로 독립 선택 과목으로 컴퓨터 교과가 신설되었다. 7차 교육과정에는 ICT 소양 중심으로 컴퓨터 교육이 시행되었다. 7차 교육과정에서 컴퓨터 교과의 영역을 살펴보면, 컴퓨터의 발달과 일과 생활에 미치는 영향을 설명하는 '인간과 컴퓨터', 컴퓨터와 소프트웨어의 구성과 조작에 대한 내용을 다루는 '컴퓨터의 기초', 문서 작업을 위한 '워드프로세서', 통신과 인터넷에 대한 내용이 포함된 'PC 통신과 인터넷', 멀티미디어 콘텐츠를 제작하는 '멀티미디어'로 영역이 구성되어 있었다 [15].

2007 개정 교육과정부터는 한국의 컴퓨터 교육과정은 큰 기점을 맞이하게 된다. 과목의 연계성을 위하여 교과목 명을 '정보(Informatics)'로 바꾸고, ICT 소양 중심의 교과 내용에서 컴퓨터과학 개념과 원리 중심으로 교육과정의 개편이 이루어진다. 2007 개정 교육과정에서는 컴퓨터의 구성과 동작, 운영체제, 네트워크에 대한 내용을 다루는 '정보기기의 구성과 동작', 자료의 구조와 정보의 표현에 대한 내용이 포함된 '정보의 표현과 관리', 알고리즘과 프로그래밍을 통한 문제 해결을 설명하는 '문제 해결 방법과 절차', 정보 사회와 윤리, 보호, 멀티미디어 콘텐츠에 대한 내용이 제시된 '정보 사회와 정보 기술'로 영역이 변경되었다 [15].

2009 개정 교육과정에서 교과목이 선택 교과로 편성되는 등 컴퓨터 교육의 침체기를 맞이하였지만, 2015 개정 교육과정에서는 소프트웨어 교육의 중요성과 국외의 컴퓨터 교육의 강화에 발맞추어 컴퓨터 교육이 초·중등학교에서 필수로 가르치게 되었다 [15, 17]. 또한, 불필요한 내용을 줄이고, 정보문화 소양, 컴퓨팅 사고력, 협력적 문제해결력을 중심으로 교육과정 내용의 개편이 이루어졌다. 현재 고시된 2015 개정 교육과정은 중학교 정보와 고등학교 정보는 크게 4가지 영역으로 구성되어 있다: 정보문화, 자료와 정보, 문제해결과 프로그래밍, 컴퓨팅 시스템. 고등학교 정보과학은 프로그래밍, 자료 처리, 알고리즘, 컴퓨팅 시스템으로 내용이 구성되어 있다. 2015 개정 교육과정은 소프트웨어 교육의 중요성에 발맞추어 정보기술을 활용하여 문제 해결에 필요한 자료와 정보를 수집하고, 실생활 문제를 추상화하여 자동화를 통한 문제를 해결할 수 있는 능력에 초점을 맞추고 있다. 또한, 문제 해결을 위하여 직접 피지컬 컴퓨팅 시스템을 구축하는 능력을 기르는 것에 목표를 두고 있다 [12, 13, 18, 19]. 하지만, 정보 교과가 초·중학교 교육과정에서 필수 교과로 지정되었지만, 시수가 소프트웨어 인재 양성을 위해서는 부족하였다. 또한, 교육과정도 시수에 맞게 설계됨에 따라 보완이 필요하다는 의견이 있었다. 미래 사회에 필요한 소프트웨어 인재를 양성하기 위해서는 한국의 교육 환경에 대한 고려와 소프트웨어 역량을 갖출 수 있는 교육과정의 개발이 필요하다는 것을 확인할 수 있다.

4. 이스라엘의 컴퓨터과학 교육과정

이스라엘은 1992년 컴퓨터 과학을 고등학교에 가르치기 시작한 이후로, 현재 중학교와 고등학교에서 컴퓨터 과학을 가르

치고 있다. 컴퓨터 과학은 과학 교과에 합쳐진 교과로 존재하고 있으며, 학교에 따라 주당 15시간에서 18시간까지 배우고 있다. 이스라엘의 컴퓨터과학 교육은 컴퓨팅 사고력과 알고리즘을 배우는 것을 목표로 하고 있으며, 모듈을 중심으로 교육이 이루어지고 있다. 중학교는 5개의 모듈로 구성되어 있으며, 3개의 필수 모듈과 2개의 선택 모듈로 구성되어 있다. 필수 모듈은 컴퓨터 과학의 핵심을 배우는 'Introduction to CS', 그래픽과 수학적 함수, 계산에 대해 가르치는 'Spreadsheet with an emphasis on its usage for scientific research', 프로그래밍 언어나 로봇을 선택하여 문제를 해결하는 프로젝트를 수행하는 과목인 'Final programming project'로 구성되어 있다. 선택 모듈로는 논리적 사고와 공학적 사고를 키우기 위하여 로봇을 활용한 알고리즘적 문제를 해결하는 'Introduction to robotics'와 HTML5, CSS, 자바스크립트(Javascript)와 같은 프로그래밍 언어를 배우는 'Basic internet programming'이 있다.

고등학교 컴퓨터과학 교육은 학생이 선택한 과정에 따라 다르게 이루어진다. 문학이나 사회에 대한 내용을 배우는 일반(General)과 과학, 기술, 컴퓨터 과학에 대한 내용을 배우는 기술(Technology)로 과정이 구성되어 있다. 고등학교 교육과정에서 컴퓨터 과학은 총 450시간을 배우도록 구성되어 있다. 컴퓨터 과학 교과는 알고리즘을 통한 문제 해결 과정과 문제 해결 과정에 프로그래밍 언어를 적용해보는 'Fundamental 1 and 2', 시스템 설계와 데이터 구조에 대한 내용을 다루는 'Software design', 고급 프로그래밍 언어를 배우는 과정인 'Second paradigm', ICT를 실제 적용해보는 교과인 'Applications', 고급 컴퓨터 과학에 대한 내용을 배우는 'Theory'로 구성되어 있다 [20].

5. 핀란드의 교육과정:KOODI 2016

핀란드는 글로벌 흐름과 소프트웨어에 대한 학생들의 요구를 반영하여 프로그래밍을 교육과정에 반영하는 교육과정 개정이 이루어졌다. 핀란드의 소프트웨어 교육을 위하여 2014년부터 소프트웨어 교육을 위한 교사 교육을 실시하고 있으며, 2016년부터는 기존의 ICT 교육을 대체하여 코딩 기반 소프트웨어 교육을 하고자 한다. 핀란드의 소프트웨어 교육은 컴퓨팅 사고력과 문제 해결능력을 기르기 위한 교육으로써, 1학년부터 9학년의 모든 학생이 배우도록 편성되어 있다. 이러한 소프트웨어 교육을 위해서 'KOODI 2016'이라는 프로젝트를 통하여 소프트웨어 교육을 위한 내용, 교수-학습 방안, 자료를 제안하였다.

'KOODI 2016'은 1~2학년, 3~6학년, 7~9학년인 3가지 그룹으로 나누어 제시되어 있다. 1~2학년에서는 문제 해결 과정을 쉽게 배울 수 있도록 놀이 중심으로 내용이 구성되어 있다. 프로그래밍이라는 개념이 명령을 전달하고, 컴퓨터가 수행한다는 내용을 놀이를 통하여 배우도록 제시되어 있다. 이를 위하여 언플러그드 활동 중심으로 교수-학습이 이루어져 있다. 3~6학년에서는 학생들이 직접 프로그래밍을 배울 수 있도록 visual

programming language를 활용하여 프로그래밍을 경험하는 내용으로 구성되어 있다. 프로그래밍을 학생들이 직접 경험할 수 있도록 교육용 프로그래밍 언어(Educational Programming Language, EPL)의 일종인 스크래치(Scratch), Kodu, Alice, 앱 인벤터(App inventor), Logo, Turtle roy를 통한 수업이 제시되어 있었다. 7~9학년에서는 텍스트 기반 프로그래밍 언어를 통하여 직접 프로그래밍을 하는 데 초점을 두고 있다 [21].

III. Method

1. 연구 절차

중등 소프트웨어 교육과정은 다음과 같은 연구 절차를 통하여 개발하였다. 소프트웨어 교육과정을 개발하기 위하여 국내외 컴퓨터 교육과정을 수집하였다. 수집한 컴퓨터 교육과정을 'CSTA K-12 Computer Science Standards'를 중심으로 비교·분석하였다. 분석한 결과를 공청회를 통해 발표하고, 소프트웨어 교육과정의 개발 방향에 대한 의견을 수렴하였다. 공청회를 통해 수렴한 의견을 바탕으로 한국의 교육 환경에 맞는 소프트웨어 교육과정을 개발하였다. 개발한 교육과정에 대한 전문가의 의견을 수집하기 위하여 공청회를 실시하여 교육과정에 대한 다양한 의견을 조사하였다. 조사한 의견을 바탕으로 교육과정을 수정 및 보완하여 중등 소프트웨어 교육과정을 최종 개발하였다.

2. 연구 대상

중등 소프트웨어 교육과정을 개발하기 위하여 국내외 컴퓨터 교육 관련 교육과정을 조사하였다. 소프트웨어 교육의 중요성이 강조됨에 따라서 다양한 나라에서 컴퓨터과학 교육을 실시하려는 움직임이 진행되고 있었다. 하지만, 교육과정 개발을 위해서는 구체적인 교육과정이 공개된 경우에만 연구에 활용할 수 있었다. 에스토니아, 핀란드, 중국, 일본, 미국, 영국, 인도 등이 컴퓨터과학 교육을 도입하려는 움직임이 있었지만, 구체적인 교육과정을 수집할 수 있는 국가는 미국과 영국뿐이었다. 따라서 영국의 'The national curriculum in England'에서 'Computing' 교과 내용과 'Computer Science: A curriculum for schools', 미국의 'CSTA K-12 Computer Science standards'를 한국의 '2015 개정 교육과정: 정보과'와 비교·분석하였다. 교육과정의 비교·분석을 위하여 기존에 해외 교육과정을 연구한 선행 연구 및 보고서를 참고하여 분석하였다 [6, 7, 8, 9, 11, 12, 13, 15, 16, 20, 21, 22, 23].

3. 연구 방법

중등 소프트웨어 교육과정을 개발하기 위하여 한국의 정보

교육과정과 미국과 영국의 교육과정을 비교하였다. 비교를 위하여 CSTA에서 개발한 ‘CSTA K-12 Computer Science standards’를 기준으로 영국과 한국의 교육과정이 어떻게 구성되어있는지 비교·분석하였다. 교육과정은 초등 부분은 제외하고, 중등 교육과정을 중심으로 분석하였다. 분석한 결과를 바탕으로 소프트웨어 교육과정을 개발하였다.

4. 전문가 검증

개발한 교육과정의 타당도를 확보하기 위하여, 전문가 검증과 공청회를 실시하였다. 공청회는 총 2회 실시하였다. 1차 공청회는 2015년 7월 한국교육과정평가원에서 실시되었으며, 컴퓨터 교육 관련 교육대학 및 사범대학 교수와 교육 연구 기관 연구원이 20여 명이 참여하여 진행하였다. 2차 공청회는 2015년 11월 고려대학교에서 진행되었으며, 컴퓨터 교육 관련 교육대학 및 사범대학 교수와 대학원생, 컴퓨터 교육 관련 연구기관 연구원 약 50여 명이 참여하였다. 공청회는 공개적으로 진행하였으며, 공청회를 통하여 연구 결과에 대한 다양한 의견을 수렴하였으며, 소프트웨어 교육과정에 대한 완성도를 높이고자 하였다.

또한, 전문가 검증은 공청회를 실시하기 전에 전문가 집단을 구성하여, 소프트웨어 교육과정에 대한 검증을 실시하였다. 전문가 집단은 컴퓨터 교육 전문가 8인으로 구성하였다.

IV. Result & Discussion

1. 국내·외 교육과정 비교·분석 결과

1.1 협력 (Collaboration)

‘CSTA K-12 Computer Science Standards’에서는 테크놀로지와 컴퓨팅 자원을 협력을 위해 활용하여 범교과적인 문제 해결과 협력적 문제 해결을 위한 내용과 협력의 가치를 이해하는 내용이 포함되어 있다. 이와 같은 내용에서 학생들은 협력적 문제 해결을 경험하고, 프로젝트 수행을 통하여 소프트웨어 개발 과정을 이해하게 된다. 영국의 ‘Computing’ 교육과정은 협력에 대한 내용이 전혀 포함되지 않은 것을 확인할 수 있다. 컴퓨팅 사고력과 핵심 개념을 기반으로 교육과정이 제작되었지만, 협력에 대한 내용은 찾아볼 수 없었다. 한국의 정보 교육과정은 협력에 대한 내용이 교육과정에는 명시되어 있지 않았다. 하지만 교육과정의 핵심 역량으로 협력적 문제해결력이 포함되어 있고, 프로그래밍 영역에서 pair programming에 대한 내용을 일부 수록하고 있었다. 따라서 교과서나 교사에 따라서 협력에 대한 내용이 다른 수준으로 가르칠 수 있는 상황이다. 그러므로 협력에 대한 내용이 포함된 교육과정이 필요하다는 것을 확인할 수 있었다.

1.2 컴퓨팅 사고력 (Computational Thinking)

컴퓨팅 사고력은 ‘CSTA K-12 Computer Science Standards’에서 문제 해결(Problem solving), 알고리즘(Algorithm), 자료 표현(Data representation), 모델링과 시뮬레이션(Modeling and simulation), 추상화(Abstraction), 다른 영역과의 연결(Connection to other)로 구성되어 있다. 문제 해결에서는 알고리즘적으로 문제를 해결하는 방법에 대해서 배우고, 문제 해결을 위해서 문제를 분석, 분해하는 내용, 소프트웨어 개발 과정에 따라서 문제를 해결하는 과정에 대한 내용이 포함되어 있다. 영국의 ‘Computing’ 교과에서는 알고리즘과 프로그래밍 영역에서 문제 해결에 대한 내용이 포함되어 있다. 하지만, 소프트웨어 개발 과정에 대한 내용은 포함되어 있지 않았다. 한국은 중, 고등학교 정보 교과와 문제해결과 프로그래밍 단원과 고등학교 정보과학의 알고리즘 단원에 포함되어 있었다. 영국과 마찬가지로 소프트웨어 개발 과정에 대한 내용은 포함되어 있지 않았다.

알고리즘은 컴퓨터의 실행과정과 관련된 알고리즘에 대한 기본적인 개념에 대해 이해하고, 알고리즘의 효율을 비교·분석하고, 자연을 이해를 돕기 위하여 자료를 활용하는 내용까지 포함하고 있다. 영국은 알고리즘에 대한 내용이 알고리즘과 데이터 영역에 포함되어 있다. 하지만 시스템과 자연의 이해를 돕기 위하여 자료를 활용하는 내용은 포함하고 있지 않았다. 한국은 문제해결과 프로그래밍 영역에서 핵심 개념으로 알고리즘이 포함되어 있다. 하지만 알고리즘 분석과 데이터 관리에 대한 내용이 부족하며, 미국과 영국에 비해 알고리즘에 대한 내용이 쉽게 구성되어 있었다.

자료 표현은 문제에 대한 정의, 다양한 자료형과 표현 방법, 다양한 진수 표현에 대한 내용으로 구성되어 있었다. 영국은 데이터 영역에서 미국 교육과정과 같은 수준의 내용으로 구성되어 있었다. 한국은 중, 고등학교 정보에서 자료와 정보 영역과 고등학교 정보과학에서 자료 처리 영역에 관련된 내용이 포함되어 있지만, 이진수나 16진수에 대한 내용은 없는 것을 확인할 수 있었다. 2009 개정 교육과정에서는 이러한 내용이 포함되어 있었지만, 학생들과 교사의 어려움을 느끼는 문제와 정보 교과의 시수 문제 때문에 2015 개정교육과정에서 제외되었다 [15].

모델링과 시뮬레이션은 모델링과 시뮬레이션을 이해하기 위하여 기본적인 내용을 학습하고, 실생활에 적용해보고, 모델링과 시뮬레이션을 활용하여 실제 문제를 해결하는 내용으로 구성되어 있다. 영국은 컴퓨팅 사고력을 key process로 제시하고 있지만, 모델링과 시뮬레이션에 대한 내용이 교육과정에 포함되어 있는지는 않았다. 반면에 한국은 고등학교 정보에 문제해결과 프로그래밍 영역에서 중학교 수준의 모델링과 시뮬레이션에 대한 내용을 다루고 있으며, 고등학교 정보과학의 컴퓨팅 시스템 영역에서 시뮬레이션의 설계와 구현에 대한 내용이 포함되어 있다. 한국의 정보 교육과정은 영국과 다르게 모델링과 시뮬레이션에 대한 내용을 포함하고 있지만, 미국에 비해 성취 기

준이 매우 낮은 것을 확인할 수 있었다.

추상화의 경우에는 문제를 작은 문제를 분해해 보는 경험에서부터 추상화의 가치에 대한 토론, 새로운 함수와 클래스를 통한 문제 분해, 문제 해결을 위한 병렬 처리에 대한 내용이 제시되어 있었다. 영국의 교육과정에서는 알고리즘과 프로그래밍 영역에서 추상화에 대한 내용이 언급된 것을 확인할 수 있었다. 한국은 고등학교 정보 교과에서 문제를 분석 및 분해하는 내용이 포함되어 있고, 문제해결과 프로그래밍 영역에서 핵심 개념으로 추상화가 포함되어 있다. 하지만 추상화의 내용이 미국의 교육과정의 Level 3A까지 포함되어 있고, Level 3B 내용까지는 포함하고 있지 않았다.

마지막으로 다른 영역과의 연결은 수학과 컴퓨터과학 간의 연결에 대한 내용과 다른 영역에 컴퓨팅 사고력이 응용되는 사례를 배움으로써 기초 지식을 다지고, 음악이나 미술에 적용되는 실제 사례를 찾아보는 것으로 구성되어 있다. 영국의 'Computing' 교과에서는 수학과 컴퓨터과학 간의 연결에 대한 내용과 다른 영역에 컴퓨팅 사고력이 응용되는 사례를 찾아보는 내용은 없었지만, 음악과 미술에 적용되는 실제 사례를 찾아보는 내용은 데이터 영역에 포함되어 있었다. 반대로 한국은 컴퓨팅 사고력이 응용되는 사례와 수학과 컴퓨터과학 간의 연결을 살펴보는 내용은 고등학교의 프로그래밍에 포함되어 있었지만, 예술에 접목하는 사례로 교육과정에 포함되어 있지 않았다.

컴퓨팅 사고력은 소프트웨어 교육의 핵심 요소로써 모든 교육과정의 핵심 역량으로 포함되어 있었다. 하지만 미국과 영국 교육과정에 포함된 내용에 비해 한국의 정보 교육과정은 많이 부족한 것을 확인할 수 있었다. 이러한 문제는 2015 개정 교육과정에서 정보 교과가 필수로 바뀌었지만, 차시가 많이 할당되지 않았으므로 교육과정의 내용이 한정적이었기 때문에 발생한 것으로 생각된다 [24, 25]. 따라서 본 연구에서 개발하고자 하는 소프트웨어 교육과정은 디지털 리터러시(Digital literacy)와 소프트웨어 역량을 갖춘 인재 양성을 위하여 컴퓨팅 사고력에 대한 내용은 보완하여 작성하였다.

1.3 컴퓨팅 연습과 프로그래밍

(Computing Practice and Programming)

컴퓨팅 연습과 프로그래밍은 학습을 위한 테크놀로지 자원 활용(Using technology resources for learning), 디지털 작품 제작을 위한 테크놀로지 도구 활용(Using technology tools for the creation of digital artifacts), 프로그래밍(Programming), 원격 자료와의 상호작용(Interacting with remote information), 직업(Careers), 자료 수집 및 분석(Data collection and analysis)으로 구성되어 있다. 학습을 위한 테크놀로지 활용은 중학교 수준까지 제시되어 있었으며, 다양한 과제나 문제 해결에 적절한 테크놀로지 자원을 활용하고, 학습의 효과를 높이기 위해 멀티미디어 도구를 활용하는 내용으로 구성되어 있었다. 이와 같은 내용은 영국 교육과정에는 제시가 되어 있지 않았으며, 테크놀로지의 의미를 포괄적으로 본

다면 한국의 교육과정에서는 고등학교 정보와 정보과학에서 피지컬 컴퓨팅 구성과 구현 요소에 포함되어 있었다.

디지털 작품 제작을 위한 테크놀로지 도구 활용에서는 결과물을 만들기 위해 학생들이 다양한 테크놀로지 도구를 활용한 내용을 포함하고 있다. 수준에 따라 테크놀로지 도구는 웹 프로그래밍 언어, 모바일 컴퓨팅 기기, 고급 테크놀로지 도구를 제시하고 있다. 영국 교육과정에서는 웹 프로그래밍 언어는 통신 및 인터넷에 포함되어 있었고, 모바일 컴퓨팅 기기, 고급 테크놀로지 도구에 대한 내용은 프로그래밍에 포함되어 있었다. 하지만 중학교 수준의 테크놀로지 도구 활용은 중학교 수준에 포함되어 있지 않았다. 한국의 경우에는 컴퓨팅 시스템 영역에서 이와 관련된 내용은 대부분 포함하고 있었다. 하지만 웹 프로그래밍 언어에 대한 내용은 포함되어 있지 않은 것을 확인할 수 있었다.

프로그래밍에 대한 내용은 설계한 알고리즘의 이해와 적용에 대한 내용과 반복, 조건, 논리, 변수, 함수 등을 활용하여 문제 해결부터 문제 해결에 API와 라이브러리 활용, 다양한 프로그래밍 언어 및 자료 활용, 디버깅과 프로그램 실행 과정을 설명하고 그것을 응용하는 과정으로 구성되어 있다. 영국의 'Computing' 교육과정은 알고리즘과 프로그래밍에 프로그래밍과 관련된 내용이 포함되어 있으며, 프로그램 실행 과정을 설명하기 위해서 문서화하는 내용까지 포함되어 있었다. 또한, 프로그래밍 언어를 배우고 문제 해결을 위한 프로그램 개발에 대한 내용에 강조하고 있고, 추상화를 통한 문제 분해에 대한 내용은 미국 교육과정에 비해 적은 것으로 나타났다. 한국은 프로그래밍에 대한 내용이 중, 고등학교 정보 교과와 고등학교 정보과학 교과에 포함되어 있지만, 미국 교육과정에 비하면 프로그래밍 내용에 대한 수준이 낮았다. 또한, 디버깅과 프로그램 실행 과정을 설명하는 내용은 부족한 것으로 나타났다. 따라서 한국의 교육과정은 영국과 미국에 비해 프로그래밍에 대한 내용의 수준이 낮은 것으로 나타났다.

원격 자료와의 상호작용은 보안에 대한 내용 중심으로 구성되어 있었다. 개인 정보 보호를 위한 암호, 보안에 대한 내용부터 암호화나 개인 인증에 대한 원리에 대한 이해, 실생활에 활용까지 다루고 있었다. 영국은 통신 및 인터넷에서 보안에 대한 내용을 다루고 있었고, stage 4부터 다루고 있었다. 한국에서는 고등학교 정보 교과의 정보 윤리 영역에서 정보의 보호와 보안에 대한 내용으로 다루고 있었지만, 원리에 대한 이해와 실생활의 활용에 대한 내용은 없었다. 따라서 미국에 비해 한국과 영국은 보안에 대한 내용은 고학년에서 다루고 있었으며, 포함된 내용도 쉬운 내용 위주로 구성되었다고 생각된다.

직업은 컴퓨터과학이 직업에 미치는 영향과 컴퓨팅과 관련된 다양한 직업, 컴퓨터과학에 따라 미래 직업과 테크놀로지의 변화에 대한 내용을 다루고 있었다. 영국에서는 직업과 관련된 내용은 전혀 다루고 있지 않았다. 반면에 한국에서는 정보 문화 영역에서 정보 교과와 관련된 진로에 대한 내용, 컴퓨터과학에 따라서 사회가 변화하는 모습을 생각하는 내용이 있었다. 영국

은 정보 문화와 관련된 영역이 없었지만, 한국은 핵심 역량으로 컴퓨팅 사고력뿐만 아니라 정보 문화 소양이 포함되어 있으므로 이와 같은 교육과정이 나타난 것으로 생각된다.

마지막으로 자료 수집 및 분석을 살펴보면, 자료 수집에 대한 내용은 문제 해결을 위해서 프로그램을 통한 자료와 대량의 자료, 다양한 문제 해결을 위해서 필요한 자료 수집에 대한 내용으로 구성되어 있다. 자료의 분석은 수학이나 통계적인 내용과 자연과 실생활을 이해하는데 필요한 자료 분석의 활용에 대한 내용을 포함하고 있었다. 영국에서는 stage별로 데이터 영역에 포함되어 있었다. 컴퓨팅 사고력을 기반으로 교육과정이 개발되었기 때문에 문제 해결에 필수적인 자료 수집과 분석에 대한 내용이 stage별로 적절하게 내용 구성이 이루어졌다고 생각된다. 한국의 경우에는 자료 수집과 분석에 대한 내용이 고등학교부터 등장하고 있었다. 고등학교 정보에서 자료와 정보 영역에서 자료 분석과 자료 관리에서 자료의 수집과 분석에 대한 내용이 나타나 있었고, 고등학교 정보과학의 자료 처리 영역에서 자료 표현과 구조에 대한 내용이 포함되어 있었다. 하지만 자연과 실생활을 이해하기 위해서 자료 분석을 하는 것이나 대량의 자료와 다양한 문제를 해결하는 데 필요한 자료 수집에 대한 내용은 없었다. 따라서 자료 수집 및 분석에 대한 내용은 미국과 영국에 비해 한국은 부족한 것으로 나타났다.

컴퓨팅 연습과 프로그래밍에 대한 내용은 미국 교육과정에 비해 영국과 한국의 교육과정은 특정 영역이 부실한 것으로 나타났다. 한국은 정보 문화 소양이 핵심 역량으로 존재하기 때문에 직업에 대한 내용이 충분히 있었지만, 보안과 프로그래밍, 자료 수집 및 분석에 대한 내용은 상대적으로 적게 수록되어 있었다. 영국은 컴퓨터과학 내용에 대해 충분히 수록되어 있었지만, 직업에 대한 내용과 테크놀로지 자원을 활용하여 학습하는 것과 학생들이 작품을 만드는 내용에서는 다른 나라에 비해 부족한 것으로 나타났다.

1.4 컴퓨터와 커뮤니케이션 장비

(Computers and Communication Devices)

컴퓨터와 커뮤니케이션 장비는 하드웨어적인 내용을 중심으로 구성되어 있었다. 세부 영역을 살펴보면, 컴퓨터(Computers)와 Troubleshooting, 네트워크(Networks), 인공지능(Human vs. Computers)으로 구성되어 있다.

컴퓨터는 크게 응용 컴퓨팅(Applied computing)과 컴퓨터 구성요소(Computer components)로 구성되어 있다. 컴퓨팅 응용은 컴퓨터와 프로그램 실행에 대한 이해를 바탕으로 다양한 컴퓨팅 디바이스와 모바일 기기에 대한 특징과 컴퓨터 구성요소의 기준 등 컴퓨팅과 관련된 다양한 내용이 포함되어 있었다. 컴퓨터 구성요소는 컴퓨터와 관련된 구성 요소에 대한 내용을 중심으로 구성되어 있다. 하드웨어와 소프트웨어의 관계 이해부터 다양한 형태의 입·출력, 파일 포맷에 대한 내용과 컴퓨터를 구성하는 요소와 프로그램 실행을 위해 다양한 수준의 하드웨어와 소프트웨어, 하드웨어 레이어에 대한 내용이 포함되

어 있었다. 영국에서는 컴퓨터 구조를 중심으로 이와 같은 내용이 포함된 것을 확인할 수 있었다. 한국의 경우에는 중학교 정보 교과에서 컴퓨팅 시스템 영역에 컴퓨팅 기기의 구조와 동작 원리에 컴퓨터 구성요소에 대한 내용이 포함되어 있고, 고등학교 정보과학의 컴퓨팅 시스템 영역에서 응용 컴퓨팅 내용이 일부 포함되어 있다. 하지만 하드웨어 요소에 대한 내용이 미국과 영국에 비해 적은 것을 확인할 수 있었다. 한국은 컴퓨팅 사고력을 중심으로 교육과정의 개정이 이루어졌고, 한정적인 시수 때문에 많은 내용을 교육과정에 담을 수 없었다. 이에 따라서 컴퓨터 하드웨어에 대한 내용이 다른 교과에 비해 적게 나타난 것으로 생각된다. 물론 피지컬 컴퓨팅 기기의 실습을 통하여 학생들이 컴퓨팅 기기를 다루는 내용을 경험하게 되지만, 다양한 영역에서 많이 활용되고 있는 컴퓨터 자체에 대한 내용도 학습의 필요성이 있다고 생각된다.

Troubleshooting은 일상생활에서 발생하는 소프트웨어와 하드웨어 문제를 해결하는데 필요한 루틴에 대한 내용을 포함하고 있다. 영국과 한국 모두 하드웨어 문제를 해결하기 위한 내용은 포함하고 있었지만, 일상생활에서 발생하는 소프트웨어 문제를 해결하기 위한 내용은 교육과정에 포함되지 않은 것으로 나타났다.

네트워크는 네트워크 시스템의 구성 요소뿐만 아니라 네트워크의 시스템 비교에 대한 내용까지 다루고 있었다. 영국은 통신과 인터넷이라는 영역이 따로 있으므로 네트워크에 대한 내용이 많이 수록되어 있었다. 한국도 고등학교 정보의 컴퓨팅 시스템 영역에 네트워크에 대한 내용이 있었다. 하지만 미국이나 영국에 비해 많은 내용이 포함되어 있지 않았다.

마지막으로 인공지능은 인간의 지능과 기계의 지능에 대해 비교해보는 내용부터 인공지능과 로봇(Robotics)의 활용 방안에 대한 내용까지 포함되어 있었다. 인공지능에 대한 내용은 영국에서는 프로그래밍 영역에서 advanced 수준에서 다루고 있었고, 한국의 교육과정에서는 아예 없는 것으로 나타났다. 알파고와 이세돌의 대결을 통하여 인공지능에 대한 관심이 증가하고 있는 상황에서 한국의 정보 교과에서는 인공지능에 대한 내용이 하나도 없는 상황은 인공지능 관련 인재를 양성하고 국가 주도의 프로젝트가 활발히 시행되는 상황과는 상반되는 결과이다. 따라서 인공지능에 대한 내용이 반영한 교육과정의 개발이 필요하다고 생각된다.

1.5 커뮤니티와 세계화, 윤리적 영향력

(Community, Global, and Ethical Impacts)

커뮤니티와 세계화, 윤리적 영향력은 책임감 있는 사용(Responsible use), 테크놀로지의 영향력(Impacts of technology), 정보의 정확성(Information accuracy), 정보 윤리(Ethics, laws, and security), 평등(Equity)으로 구성되어 있었다.

책임감 있는 사용은 정보나 테크놀로지의 부적절한 사용에 따른 법적 책임과 네트워크와 소셜미디어 및 기기에 대한 올바

른 사용에 대한 내용으로 구성되어 있었다. 영국의 교육과정에서는 이와 관련된 내용이 존재하지 않았다. 한국의 경우에는 정보 문화 영역에서 사이버 윤리와 저작권에 대한 내용에 포함되어 있었다. 한국의 교육과정에는 이와 같은 내용에 게임과 스마트폰 중독에 대한 내용이 추가로 포함되어 있었다.

테크놀로지의 영향력에서는 테크놀로지와 컴퓨팅의 발전을 살펴보고, 발전에 따라서 사회 전반과 문화에 미친 영향을 비교·분석하는 내용이 있었다. 영국은 많은 내용이 교육과정에 포함되어 있지 않았지만, 컴퓨팅의 혁신이 미친 영향에 대한 내용으로 오토마타, 생물학 연구, 행동 및 심리학 연구 등 다양한 사례를 알고리즘 영역에서 제시하고 있었다. 한국의 경우에는 정보문화 영역에서 소프트웨어의 가치와 정보과학과 사회 내용에 테크놀로지의 영향력이 포함되어 있었다. 하지만, 테크놀로지가 사회에 미치는 영향으로 한정되어 있으며, 자동화와 컴퓨팅 등이 사회에 미친 영향을 제시하는 내용은 미국에 비해 부족한 것으로 나타났다.

정보의 정확성에 대한 내용은 문제 해결을 위하여 인터넷을 통해 얻은 정보의 정확성이나 관련성, 적절성이 포함되어 있었다. 정보의 정확성은 영국과 한국의 교육과정 모두에서 포함되어 있지 않았다.

정보 윤리는 도덕과 법, 윤리에 대한 내용을 소프트웨어의 저작권과 보안 중심으로 서술되어 있었다. 영국의 'Computing' 교육과정은 위와 마찬가지로 정보 윤리와 관련된 내용이 하나도 포함되어 있지 않았다. 한국의 정보 교과에서는 저작권을 내용 요소로 포함하고 있으며, 정보 보안과 관련된 내용이 정보 문화에 포함되어 있어서 정보 윤리에 대한 내용이 많이 수록된 것을 확인할 수 있었다.

마지막으로 평등은 컴퓨팅 자원과 정보의 분배 때문에 생기는 문제를 찾아보고, 미치는 영향과 실제 사례에 대해 생각해보는 것으로 구성되어 있다. 영국과 한국의 교육과정 모두 평등에 대한 내용은 거의 수록되지 않았다. 영국은 평등과 관련된 내용이 하나도 존재하지 않았고, 한국은 중학교 정보 교과의 정보 문화에서 정보와 소프트웨어의 가치에 관해 설명하는 부분이 제시되어 있었다.

이와 같은 분석을 살펴보면, 영국은 커뮤니티와 세계화, 윤리적 영향력에 대한 내용은 거의 수록되어 있지 않은 것을 확인할 수 있었다. 한국은 정보 문화 소양이 핵심 역량 중 하나이므로 많은 내용이 포함되어 있었지만, 내용 수준은 미국의 교육 과정에 비해 낮은 것을 확인할 수 있었다.

한국의 정보 교과는 다른 나라와 비교하였을 때, 내용 수준이 낮은 것을 확인할 수 있었다. 정보 윤리에 대한 내용도 'CSTA K-12 Computer Science Standards'에 비해서 낮았으며, 프로그래밍이나 컴퓨팅 사고력에 대한 내용도 성취 기준이 미국과 영국에 비해 낮은 것으로 나타났다. 미국 교육과정에 포함되어 있었던 협력에 대한 내용은 중등 교육과정에서 아예 포함되지 않았다. 그뿐만 아니라 컴퓨팅 사고력과 협력적 문제 해결을 중심으로 교육과정 개편이 이루어짐에 따라 네트워크와

운영체제, 하드웨어와 관련된 내용은 많이 포함되어 있지 않은 것을 확인할 수 있었다. 마지막으로 인공지능에 대한 내용이 전혀 포함되어 있지 않았다. 뇌 과학과 신경과학이 컴퓨터과학과 융합됨에 따라서 인공지능의 발전은 급속도로 이루어졌고, 2016년 다보스포럼에서는 인공지능이 로봇과 함께 제4차 산업혁명을 이끌 요소로 발표되었다 [1]. 이러한 인공지능은 음성 인식, 무인 자동차 등 다양한 분야에 활용되고 있다. Google의 Deepmind에서 개발한 'Alphago'는 2016년 이세돌과의 바둑 경기에서 승리함에 따라 인공지능에 대한 관심은 더욱더 증가하였다. 미래 사회에서는 소프트웨어의 발전에 따라서 일상생활에서 인공지능의 활용이 증가할 것이므로, 소프트웨어 교육에서 인공지능에 대한 내용을 포함하고 학생들에게 교육을 실시하는 것이 필요하다고 생각된다.

영국의 'Computing' 교과는 한국 교육과정과 마찬가지로 Collaboration 영역의 내용이 아예 존재하지 않았다. 또한, 한국과는 반대로 컴퓨팅 관련 진로, 사이버 윤리, 저작권, 법, 평등과 같은 정보 윤리 및 소양 관련 내용이 교육과정에서 거의 존재하지 않았다. 하지만 key concept가 컴퓨터과학 관련 요소로 구성되어 있으며, key process가 컴퓨팅 사고력을 기반으로 교육과정이 개정됨에 따라 알고리즘, 프로그래밍, 자료구조, 컴퓨터구조, 네트워크와 관련된 내용이 교육과정에 많이 포함된 것을 확인할 수 있었다. 이러한 컴퓨터 과학 내용은 한국과 미국보다 수준이 높은 것을 확인할 수 있었다.

2. 중등 소프트웨어 교육과정 개발

미국과 영국, 한국의 컴퓨터과학 교육과정 비교를 통하여 미국의 컴퓨터과학 교육이 다른 나라의 교육과정에 비해 다양한 영역의 내용이 포함되어 있다고 판단하였다. 이에 따라 중등 소프트웨어 교육과정의 개발을 위해서 'CSTA K-12 Computer Science Standards'를 기반으로 한국의 교육 현실에 맞는 소프트웨어 교육과정을 개발하였다. 개발한 소프트웨어 교육과정은 중학교, 고등학교, 고등학교 심화 수준으로 제시하였고, 협업과 융합 (Collaboration and Convergence), 컴퓨팅 사고력 (Computational Thinking), 컴퓨팅 실습과 프로그래밍 (Computing Practice and Programming), 컴퓨터와 커뮤니케이션 장비 (Computers and Communication Devices), 커뮤니티, 세계화, 윤리적 영향력 (Community, Global, and Ethical Impacts)으로 영역을 구성하였다.

2.1 협업과 융합 (Collaboration and Convergence)

협업과 융합 영역은 컴퓨팅 기기나 테크놀로지 기기를 활용하여 동료와 협업할 수 있는 능력을 기르고, 다른 분야에 컴퓨팅 기기를 접목해 문제를 해결하는 내용을 포함하고 있다. 협업과 융합 영역에서 하위 영역으로 협업을 위한 기술과 자원 활용 (Using technology tools and resources for collaboration) 과 협업을 위한 컴퓨팅 (Computing through collaboration), 다른 분야와 연결 (Connections to other parts)이 있다. 협업

을 통한 기술과 자원 사용에서는 협업이나 문제 해결 활동, 프로젝트 수행 과정에서 테크놀로지와 자원을 활용하여 협업을 경험하는 내용을 포함하고 있다. 협업을 통한 컴퓨팅에서는 소프트웨어 개발 과정과 관련된 협업에 대해 살펴보고, 협업 도구들이 이에 미치는 영향을 살펴보는 내용을 포함하고 있다. 마지막으로 다른 분야와 연결에서는 수학이나 예술과 같은 다른 학문에 컴퓨팅이 어떻게 활용되는지 알아보는 내용이 포함되게 구성하였다. 2015 개정 교육과정에서 정보 교과에서는 융합에 대한 내용이 피지컬 컴퓨팅 영역에서 일부 포함되어 있었지만, 고등학교에 한정되어 있으며, 협업에 대한 내용은 포함되어 있지 않았다. 소프트웨어 개발 과정에서는 협업할 수 있는 필수 역량이다 [13]. 따라서 소프트웨어 교육과정에서 학생들이 다양한 프로젝트에서 협업할 수 있는 능력을 배양할 수 있도록 관련 내용이 포함될 필요가 있다. 본 소프트웨어 교육과정은 학생들이 협업 과정을 활용할 수 있는 도구를 살펴보고, 협업을 통해 문제를 해결해나가는 과정을 경험하도록 구성하였다. 또한, 다양한 영역에 컴퓨팅이 적용 및 융합된 사례를 살펴봄으로써, 컴퓨팅이 실생활에서 어떻게 적용되고 있으며 어떻게 활용할 수 있는지 알 수 있도록 개발하였다. 협업과 융합에 대한 교육과정 내용은 Table 1과 같다.

2.2 컴퓨팅 사고력(Computational Thinking)

컴퓨팅 사고력은 Wing(2006)의 연구를 이후로 컴퓨터과학과 소프트웨어 교육에 필수 역량으로 강조되고 있다 [26]. 이에 따라 미국, 영국, 한국의 소프트웨어 교육과정에서는 컴퓨팅 사고력을 key concept이나 핵심 역량으로 제시하고, 학생들이 컴퓨팅 사고력을 향상시키기 위한 교육과정 내용을 구성하고 있었다. 본 연구에서도 컴퓨팅 사고력의 중요성을 소프트웨어 교육과정에 반영하기 위하여 컴퓨팅 사고력의 하위 요소들을 교육과정 내용에 포함하였다. 컴퓨팅 사고력의 하위 영역을 살

펴보면, 문제 해결 (Problem solving), 알고리즘 (Algorithm), 자료 표현 (Data representation), 모델링과 시뮬레이션 (Modeling and Simulation), 추상화 (Abstraction), 자료 수집과 분석 (Data collection and analysis)으로 구성하였다. 문제 해결은 문제 해결 과정에 알고리즘이나 프로그래밍, 추상화를 통한 문제의 분해 및 분류, 휴리스틱 알고리즘 등에 대한 내용으로써, 배운 내용을 활용하여 실제로 문제를 해결하는 과정을 수준에 따라 경험할 수 있도록 구성하였다. 알고리즘은 알고리즘에 대한 이해와 문제 해결 과정에서 적용부터 문제 해결을 위하여 활용되는 다양한 알고리즘에 대한 평가와 선택, 구현에 대한 내용을 포함하고 있다. 세 번째로 자료 표현은 중학교 수준에서는 다양한 자료를 글이나 소리, 사진으로 표현하거나 그림이나 도형을 활용하여 문제를 표현하는 내용을 제시하였고, 고등학교에서는 디지털 정보를 진수 표현과 컴퓨터 시스템에서 사용되는 다양한 형태로 표현하는 내용을 포함하였다. 마지막으로 심화 과정에서는 배열이나 리스트, 그래프, 트리와 같이 간단한 데이터 구조를 이해할 수 있는 내용을 제시하였다. 그다음 하위 영역인 모델링과 시뮬레이션은 문제 해결에 활용하는 모델링과 시뮬레이션에 대한 내용을 수준에 따라 컴퓨터 과학 문제, 자연 현상과 실생활 문제, 과학적 가설의 검증을 위해 활용하는 내용으로 구성하였다. 컴퓨팅 사고력에서 핵심 요소인 추상화는 문제를 분석하여 작은 문제로 분해하는 내용부터, 추상화의 개념과 가치를 이해하고 실생활 문제를 분해하는 내용, 함수와 클래스, 병렬처리를 활용하여 어렵고 큰 문제를 분해하는데 추상화를 활용하는 내용을 제시하였다. 마지막으로 데이터 수집과 분석은 인터넷, 응용 소프트웨어, 프로그램을 통한 자료 수집 및 분석에 대한 내용을 기초로 문제를 해결하기 위하여 다양한 데이터를 수집하고 분석하는 내용을 학생들이 이해할 수 있도록 구성하였다. 컴퓨팅 사고력 영역은 학생들이 컴퓨팅 사고력의 하위 영역에 대한 이해뿐만 아니라 실생활 문제

Table 1. Collaboration and convergence in software education curriculum.

Level	Domain		
	Using technology tools and resources for collaboration	Computing through collaboration	Connections to other parts
Middle school	<ul style="list-style-type: none"> - Utilize multimedia tools which increase productivity in personal and cooperate learning. - Make a design, development, product utilizing computer science technology resources like video, podcast, web-site, and paper. - Collaborate with colleague, expert through pair programming, programming training, group work in problem-solving activity. 	<ul style="list-style-type: none"> - Represent the elements needed for collaboration. 	<ul style="list-style-type: none"> - Understand the relationship with mathematics and computer science. - Represent an example that computer science merged with other subject.
High school	<ul style="list-style-type: none"> - Make a software with co-worker. - Use tools like discussion thread, wiki, blog, SNS, that facilitate collaboration, in communication with co-worker. - Explain how computing change the form of collaboration and how experiences, expressions, communication, cooperation vary. 	<ul style="list-style-type: none"> - Understand what effect collaboration has on software design and development. - Understand what effect computing has on collaboration and communication. 	<ul style="list-style-type: none"> - Represent an example that computational thinking applied to other study. - Explain how someone share one's thoughts and feelings in artistic way using computer.
Advanced	<ul style="list-style-type: none"> - Use project collaboration tools, version management system, and integration development system for software project. - Explain software development process involving software project. 	<ul style="list-style-type: none"> - Examine the readability and re-serviceability of a program someone else develop. 	

에 적용하는 내용을 포함하여 학생들이 실제로 컴퓨팅 사고력을 배양할 수 있도록 소프트웨어 교육과정을 개발하였다. 이와 같은 컴퓨팅 사고력의 교육과정에 대한 자세한 내용은 Table 2와 같다.

**2.3 컴퓨팅 실습과 프로그래밍
(Computing practice and programming)**

컴퓨팅 실습과 프로그래밍은 학생들이 미래 사회에 요구되

는 디지털 리터러시를 배양할 수 있도록 ICT 도구를 학습에 활용하고, 실제로 결과물을 만드는 내용을 포함하고 있다. 또한, 컴퓨팅 사고력에서 핵심 요소 중의 하나인 자동화 능력을 향상할 수 있는 프로그래밍에 대한 내용도 제시되어 있다. 컴퓨팅 실습과 프로그래밍의 하위 영역을 살펴보면, 학습을 위한 기술 자원 활용 (Use technology for learning), 디지털 작품 제작을 위한 테크놀로지의 활용 (Use technology for the development of digital artifact), 프로그래밍 (Programming)

Table 2. Computational thinking in software education curriculum.

Level	Domain		
	Abstraction	Algorithm	Modelling and simulation
Middle school	<ul style="list-style-type: none"> - Understand what is needed to analyse a problem. - Identify essential elements for a problem. - Decompose a problem into several simple problems through abstract. 	<ul style="list-style-type: none"> - Understand that one problem would be solved by various algorithm. - Explain and analyse sequential instruction. - Explore various solution and process in real-life problem. - Represent problem-solving process to text or picture. - Explain method and procedure for problem solving. 	<ul style="list-style-type: none"> - Interact with co-worker using specific model and simulation. - Use modelling and simulation for solving computer science problem. - Understand which problems could be solved by modelling and simulation.
High school	<ul style="list-style-type: none"> - Discuss about value of abstract for problem solving. - Understand high-level programming language, interpreter, instruction set, logical circuit, and abstract. - Decompose a real-life problem into several simple problems through abstract. 	<ul style="list-style-type: none"> - Explain how sequence, selection, iteration, and recursion to represent the algorithm. - Assess the performance of various algorithm useful to one problem. - Use search algorithm and sort algorithm frequently used. - Select the appropriate algorithm by data value and structure. 	<ul style="list-style-type: none"> - Analyse an accuracy of computer modelling used to describe real-life. - Use modelling and simulation to understand natural events and describe them.
Advanced	<ul style="list-style-type: none"> - Decompose a problem as new function and class and then analyse it. - Understand and explain parallel processing for big, complex problem solving. - Decompose a big-size problem into several simple problems through abstract. 	<ul style="list-style-type: none"> - Implement specific algorithm. - Assess algorithm by efficiency, accuracy, clarity. - Analyse data to understand nature and human system. 	<ul style="list-style-type: none"> - Use modelling and simulation to shape, re-define, verify the scientific hypothesis. - Recognize patterns and analyse data through modelling and simulation.
Level	Domain		
	Data representation	Problem solving	Data collection and analysis
Middle school	<ul style="list-style-type: none"> - Represent data into text, sound, image and number. - Schematize problem's state and structure, data using image or figure. 	<ul style="list-style-type: none"> - Apply algorithmic routine to solve the problem. - Explain parallel process related with problem solving. - Represent problem-solving algorithm into computer instruction code. 	<ul style="list-style-type: none"> - Collect data using Internet and application software. - Analyse data produced from computer program.
High school	<ul style="list-style-type: none"> - Describe relationship between binary literal and hexadecimal literal. - Understand that various forms used to describe digital data are complementary. - Explain how data are saved in computer system. 	<ul style="list-style-type: none"> - Split complex problem into simple problems using function, parameters, class and method. - Apply software development process to solve software problem. 	<ul style="list-style-type: none"> - Describe how data is collected, saved and expressed. - Explain how mathematical, statistical function, set, logic are used for computational calculation.
Advanced	<ul style="list-style-type: none"> - Describe data structure, like array and list, and how to use them. - Discuss interpretation of different binary literal. 	<ul style="list-style-type: none"> - Assort various problems into three kinds: easy-handled, difficult-handled, impossible-handled. - Use of heuristic algorithm to get an approximate answer of complex problem. 	<ul style="list-style-type: none"> - Use data analysis to understand nature and human system well. - Collect and utilize data to solve various formed problem.

으로 구성하였다. 학습을 위한 기술 자원 활용은 중학교 수준에서만 내용을 제시하였으며, ICT 도구를 학습에 활용하는 내용을 포함하고 있다. 디지털 작품 제작을 위한 테크놀로지의 활용에서는 웹 디자인을 위한 프로그래밍 언어, 모바일 컴퓨팅 프로그램 개발을 위한 언어 등 다양한 고급 프로그래밍 언어를 수준에 맞게 활용하여 결과물을 개발하는 내용이 포함하였다.

프로그래밍은 중학교 수준에서는 블록 기반 프로그래밍 언어, 고등학교 수준에서는 텍스트 기반 프로그래밍 언어, 심화 과정에서는 객체 지향 및 함수형 언어를 활용하여 문제를 해결하기 위한 내용으로 구성하였다. 이러한 문제 해결을 위하여 순차, 선택, 반복과 같은 기본적인 내용과 언어의 특징과 실행 과정에 대한 설명, 다양한 플랫폼 환경에 맞는 프로그램 제작에 대한 내용이 있다. 컴퓨팅 실습과 프로그래밍은 학생들이 다양한 ICT 도구와 프로그래밍 언어를 경험하고 실제 문제를 해결할 수 있는 능력을 기르는 데 초점을 맞추어 교육과정을 개발하였다. 컴퓨팅 실습과 프로그래밍에 대한 소프트웨어 교육과정 내용은 Table 3과 같다.

2.4 컴퓨터와 커뮤니케이션 장비(Computers and communication devices)

컴퓨터와 커뮤니케이션 장비는 2015 개정 교육과정에서 피지컬 컴퓨팅 영역에 대한 내용을 많이 포함하고 있는 영역이다. 세부 영역을 살펴보면, 응용 컴퓨팅 (Applied computing), 컴

퓨터의 구성 요소 (Components of computer), 문제 해결 (Troubleshooting), 네트워크 (Network), 인공지능 (Artificial intelligence)이 있다. 응용 컴퓨팅은 피지컬 컴퓨팅에 대한 기초적인 이해를 기반으로 센서를 활용하여 피지컬 컴퓨팅 기기를 구현하고, 실생활 문제를 해결하기 위하여 설계하는 내용까지 포함하고 있다. 컴퓨터의 구성 요소는 하드웨어와 소프트웨어에 대한 내용을 제시하고 있다. 컴퓨터의 구성 요소에서 배운 지식은 응용 컴퓨팅 분야에 활용되어 실생활 문제 해결에 필요한 컴퓨팅 장치들의 관계를 이해하는 데 도움을 주는 내용으로 구성하였다. 그다음으로 문제 해결 (Troubleshooting)을 살펴보면, 컴퓨팅 사고력의 문제 해결이 학생들이 추상화와 자동화를 통해 실생활 문제를 해결하는 데 초점을 두고 있지만, 문제 해결은 일상생활에서 겪는 하드웨어 문제를 해결하기 위한 전략을 배우는 내용으로 구성하였다. 네트워크는 중학교 수준에서는 컴퓨터와 네트워크를 구성하는 요소와 기능에 대해 살펴보고, 고등학교와 심화 수준에서 네트워크 환경과 성능에 미치는 요소를 살펴보면, 보안을 위한 암호화와 보안, 관련 기술의 원리에 대해 배우도록 제시되어 있다. 마지막으로 인공지능은 우리 사회에 미치는 영향과 인공지능과 관련된 기술을 접하는 내용으로 구성하였다.

2015 개정 교육과정에서는 컴퓨팅 사고력, 정보 문화 소양, 협력적 문제 해결력을 중심으로 교육과정이 편성되었다. 하지만 부족한 시수와 초·중학교에 공통 교육과정으로 포함됨에 따라 컴퓨터과학 관련 내용이 많이 빠지게 되었다. 컴퓨터와 커뮤니케이션 장비에 대한 내용은 많이 포함하고 있는 영역이다.

Table 3. Computing practice and programming in software education curriculum.

Level	Domain		
	Use technology for learning	Use technology for the development of digital artifact	Programming
Middle school	<ul style="list-style-type: none"> - Select a proper tools and resources for problem solving. - Learn and making a product using multimedia tools and assist devices during whole curriculum. 	<ul style="list-style-type: none"> - Design, develop, make a product that enables presentation and communication with other concepts in curriculum using computer science resources. 	<ul style="list-style-type: none"> - Document the way how program operates. - Develop and debug program. - Create a program using block-based programming language. - Understand how program and algorithm works. - Understand variable, operator, sequential, selection, recursive.
High school		<ul style="list-style-type: none"> - Make, organize one's web page using various web development programming tools. - Design and develop mobile computing program using mobile devices and emulator. 	<ul style="list-style-type: none"> - Explain and understand the debug in various way for an accurate operating of a program. - Understand algorithm and how it is used. - Create a program including operator, sequential, selection, recursive, condition, logic, expression, function, control construct. - Use technique including analysis, design, develop to solve the problem. - Develop a program using text-based programming language. - Select an appropriate file format for using of various type of data. - Describe the characteristic of various programming language. - Describe how a program operates. - Develop a program merged with different study.
Advanced		<ul style="list-style-type: none"> - Create a work using high-tech tools as wish. 	<ul style="list-style-type: none"> - Develop a program using recursive functions. - Develop a program using object-oriented programming and functional programming language. - Develop a program adjusted to algorithm on different platform. - Describe the rule of scaling, efficiency, security.

니케이션 장비에서는 2015 개정 교육과정에서 피지컬 컴퓨팅 영역에 네트워크와 하드웨어 관련 컴퓨터 과학 내용을 보강하고, 최근 들어 중요성이 강조되고 있는 인공지능 관련 내용을 추가하여 개발하였다. 이를 통하여 학생들이 실생활 문제에 필요한 지식을 갖출 수 있도록 소프트웨어 교육과정을 구성하였다. 컴퓨터와 커뮤니케이션 장비에 대한 소프트웨어 교육과정 내용은 Table 4와 같다.

**2.5 커뮤니티, 세계화, 윤리적 영향력
(Community, global, and ethical impacts)**

커뮤니티, 세계화, 윤리적 영향력 영역에서는 2015 개정 교육과정에서 핵심 역량 중 하나로 제시되어 있는 정보 문화 소양을 중심으로 내용을 구성하였다. 하위 영역으로 컴퓨팅 기술의 영향 (Impacts of technology), 정보 정확 (Information accuracy), 평등 (Equity), 윤리, 법, 보안 (Ethics, laws, and security), 직업 (Career)으로 구성하였다. 컴퓨팅 기술의 영향은 학생들이 교육과정에서 배우는 다양한 컴퓨팅 기술들이 사회에 어떻게 영향을 미치는 살펴보고, 영향력을 긍정적, 부정적인 관점에서 분석하는 내용이 포함되어 있다. 정보 정확성에서

는 자료 수집을 통해 얻은 정보에 대한 정확도, 신뢰도, 적절성, 포괄성을 판단하기 위한 내용으로 구성되어 있다. 평등 영역에서는 컴퓨팅 기술이 발달함에 따라 발생하는 정보 격차에 대한 내용을 살펴보고, 학생들이 정보 격차에 대한 자신의 의견을 토론하고 설명할 수 있는 능력을 기를 수 있는 내용이 제시되어 있다. 윤리, 법, 보안에서는 정보 윤리와 관련된 내용뿐만 아니라 지적 재산권, 개인정보 보호, 관련 법률을 실생활 사례를 경험하도록 개발하였다. 마지막으로 직업은 학생들이 소프트웨어 교육을 통해 경험한 기술이 산업 현장에 어떻게 활용되고, 이와 관련된 직업이 무엇이 있는지 체험할 수 있도록 구성하였다. 본 영역에서는 학생들이 컴퓨팅 기술이 사회에 미치는 영향과 발생하는 문제에 대해서 알아보는 내용으로 구성되어 있다. 2015 개정 교육과정에서 정보 문화 영역이 포함되어 있으므로 교육 과정에 학생들의 수준에 따라 갖추어야 할 내용을 보강하는 방향으로 소프트웨어 교육과정을 개발하였다. 커뮤니티, 세계화, 윤리적 영향력에 대한 소프트웨어 교육과정 내용은 Table 5와 같다.

Table 4. Computers and communication devices in software education curriculum.

Level	Domain				
	Applied Computing	Components of computer	Trouble shooting	Network	Artificial intelligence
Middle school	<ul style="list-style-type: none"> - Understand that it is a computer that perform program. - Explain various electrical equipment that contain computing processor. - Design a data processing and operation control program based on sensor input. - Design and develop a physical computing system using sensors to solve real-life problem. 	<ul style="list-style-type: none"> - Understand and describe the relationship between hardware and software. - Take appropriate term when communicate with computer. 	<ul style="list-style-type: none"> - Figure out difficulties through when using computer. 	<ul style="list-style-type: none"> - Explain the component and function of computer and network. 	<ul style="list-style-type: none"> - Compare intelligence of human to machine and communication method of them.
High school	<ul style="list-style-type: none"> - Describe characteristic of embedded computer in mobile devices. - Make a criteria of purchasing or upgrading hardware. - Design and develop a physical computing system using sensors to solve a problem. 	<ul style="list-style-type: none"> - Explain fundamental elements of computer. - Compare various input/output devices. - Describe multi-level structured hardware and software for assisting program. 	<ul style="list-style-type: none"> - Figure out hardware or software problem and build problem-solving strategy. 	<ul style="list-style-type: none"> - Compare client-server and peer-to-peer network. - Explain the component of computer network. - Describe advanced in internet. - Set network environment. - Explain encryption as a method of data-security. 	<ul style="list-style-type: none"> - State function of AI and robot and its use. - Explain how computer imitate human's intelligent behaviour like motion, awareness.
Advanced	<ul style="list-style-type: none"> - Discuss the effect of functional change on application program. - Design a physical computing system to solve real-life problem. 	<ul style="list-style-type: none"> - Discriminate type of hardware and explain. - Discriminate and select the most appropriate file format in terms of balance. - Configure appropriate computing devices for problem solving. 		<ul style="list-style-type: none"> - Understand the component that affect network functionality. - Explain redundancy, error-collection as a method of data-security. - Describe principle of non-symmetric encryption and key transport and how to use them. 	<ul style="list-style-type: none"> - Discuss the effect AI influence on future world. - Explain the concept of intelligent behaviour in computer modelling and robot.

IV. Conclusion

소프트웨어의 중요성이 증가됨에 따라 전 세계적으로 소프트웨어 교육의 필요성이 증가되었다. 이에 따라 다양한 나라에서 소프트웨어 교육을 도입하였고, 한국도 2015 개정 교육과정에서 소프트웨어 교육을 초·중학교에 필수로 도입하였다. 하지만 부족한 차시 배정과 이에 따른 교육과정 내용의 부족으로 소프트웨어 산업의 인재 양성에 어려움을 겪는 것이 예상된다.

본 연구에서는 이러한 문제를 해결하기 위하여 중·고등학교에서 활용할 수 있는 소프트웨어 교육과정을 개발하기 위하여 미국, 영국, 한국의 컴퓨터 교육과정을 비교·분석하였고, 미국 교육과정을 기반으로 수준에 따라 학생들이 배워야 하는 내용을 핵심 개념과 학생의 수준에 따라 제시하였다. 이러한 연구를 통하여 다음과 같은 결론을 얻었다.

첫 번째, 한국의 2015 개정 교육과정을 미국, 영국 교육과정 비교·분석하였다. 미국과 영국, 한국의 소프트웨어 교육과정을 비교하여 새롭게 개정된 정보과 2015 개정 교육과정에서 보완해야 할 내용을 파악하였다. 또한, 컴퓨터 교육 전문가, 교육 전문가 등을 초청하여 실시한 공청회에서 3개 국가의 교육과정을

비교한 결과를 검증하는 과정을 실시하였다. 이를 통하여 한국의 상황에 맞는 소프트웨어 교육과정을 개발하기 위한 방향과 시사점을 도출하였다.

두 번째, 한국의 교육 상황에 맞는 중등 소프트웨어 교육과정을 개발하였다. 미국의 컴퓨터과학 표준을 기반으로 핵심 개념을 설정하고, 학생의 수준을 중학교, 고등학교, 고등학교 심화로 나눈 뒤, 소프트웨어 교육과정을 개발하였다. 개발한 소프트웨어 교육과정을 컴퓨터 교육 전문가 등 다양한 교육 전문가를 초청하여 공청회를 진행하였고, 소프트웨어 교육과정의 보완해야 할 점에 대한 의견을 수렴하였다. 의견을 반영하여 수정 및 보완하여 최종 소프트웨어 교육과정을 개발하였다.

개발한 소프트웨어 교육과정은 핵심 개념과 학생의 수준에 따라 내용 체계가 제시되고 있지만, 학교 현장에서 활용할 수 있는 교재나 학습 활동, 교수-학습 프로그램은 포함되지 않았다. 따라서 내용 체계에 따라 이러한 내용을 보완하는 연구가 필요하다. 또한, 소프트웨어 영역의 기술이 급속도로 발전하고 있으며, 교육 환경의 변화가 꾸준히 이루어지고 있으므로 이러한 변화를 반영한 소프트웨어 교육과정이 필요하다. 따라서 지속적으로 다양한 영역의 전문가 의견을 반영하여 지속적인 정보 과학 교육과정의 개선 작업이 필요하다.

본 연구는 중학교, 고등학교, 고등학교 심화 수준의 소프트

Table 5. Community, global and ethical impacts in software education curriculum.

Level	Domain				
	Impacts of technology	Information accuracy	Equity	Ethics, Laws, and Security	Career
Middle school	<ul style="list-style-type: none"> - Describe changing the computing in education, job, society with the passage of time - Analyse the impact of computer to human culture in positive and negative perspective. 	<ul style="list-style-type: none"> - Assess accuracy, reliability, appropriacy, diversity about real-life problem. 	<ul style="list-style-type: none"> - Discuss uneven distribution of computing resources by capital, accessibility, power. 	<ul style="list-style-type: none"> - Explain legal restrictions related with abusing computer skills and technology. - Describe ethical issue related with network. - Understanding type of software license and protect intellectual property. - Try to maintain data security and privacy policy in daily lives. 	<ul style="list-style-type: none"> - Describe what occupation newly rising as computer science is advanced. - Think flexibly, not be fixed on one idea while programming. - Explore one's career related regarding computer science.
High school	<ul style="list-style-type: none"> - Describe the impact of computing on business and commerce. - Discuss the role of computing in everyday life. - Compared to the impact of computing on a positive and negative perspective on culture. 	<ul style="list-style-type: none"> - Decide whether to use or not information gained from Internet estimation reliability. 	<ul style="list-style-type: none"> - Understand the ripple effect caused by information gap. 	<ul style="list-style-type: none"> - Explain ripple effects on the economy of cracking software. - Distinguish the right to access to information and the right to distribute information. - Recognize which is appropriate or inappropriate behaviour on social network service. - Describe pros and cons of the way software is created and shared. - Describe how protect software copyright. - Discuss issues about security and privacy policy related with computer network. 	<ul style="list-style-type: none"> - Explain various work and occupation which is key of computing.
Advanced	<ul style="list-style-type: none"> - Understand pros and cons of computing revolution. - Describe how automation change financial market's deal and prediction. 			<ul style="list-style-type: none"> - Use today's communication media and devices ethically. - Explain regulation that influence on software development and utilization. - Understand effects of privacy policy and system security. 	<ul style="list-style-type: none"> - Predict what occupation will be newly rising as computer science advanced.

웨어 교육과정을 개발하였다. 미래 사회에 필요한 인재를 양성하기 위해서는 초등학교 수준부터 연계가 필요하며, 정보과학에 대한 전문적인 내용을 배우며, 진로에 대한 직접적인 결정이 이루어지는 대학 교육과도 연계가 필요하다. 따라서 초등학교, 중학교, 고등학교, 대학교가 연계성을 가지며, 지속적인 소프트웨어 교육이 이루어질 수 있는 교육과정의 개발이 필요하다. 또한, 앞으로 소프트웨어 교육의 방향을 제시하기 위한 로드맵 제시와 이를 지원하기 위한 정책과 교육 연구가 지속적으로 필요하다.

REFERENCES

- [1] Hyundai Research Institute, The main contents and implications of the 2016 World Economic Forum (Volume 16-2). Retrieved from https://www.designdb.com/Share/fileDownload_1.asp?dCate=ContentsInfo&dID=PKID&pID=22765. 2016.
- [2] ysjeong. "Needs Analysis of Software Education Curriculum at National Universities of Education for the 2015 Revised National Curriculum," Journal of The Korea Association of Information Education, Vol. 20, No. 1, pp. 83-92, 2016.
- [3] ahsang & yjee, "Educational Objectives in Computing Education," Journal of the Korea society of computer and information, Vol. 21, No. 1, pp. 181-189, 2016.
- [4] jwchoi & yjee, "Educational Application of Puzzles for Algorithm Learning of Informatics Gifted Elementary School Students," Journal of the Korea society of computer and information, Vol. 20, No. 5, pp. 151-159, 2015.
- [5] skJeon & yjLee, " Art based STEAM Education Program using EPL," Journal of the Korea society of computer and information, Vol. 19, No. 4, pp. 149-158, 2014.
- [6] Computer Science Teachers Association, CSTA K-12 Computer Science Standards. Retrieve from http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf. 2011
- [7] K12CS, A Framework for K-12 Computer Science Education. Retrived from <http://k12cs.org>. 2016.
- [8] Department for Education in U. K., The national curriculum in England: Framework document. Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/210969/NC_framework_document_-_FINAL.pdf. 2013.
- [9] hylee, Japan, Compulsory of Computer Programming Education in Elementary school Curriculum . Retrieved from <http://www.yonhapnews.co.kr/>. 2016.
- [10] skshin & ykbae, "Review of Software Education based on the Coding in Finland," Journal of The Kora Association of Information Education, Vol. 19, No. 1, pp. 127-138, 2015.
- [11] jssung, & hckim, "Analysis on the International Comparison of Computer Education in Schools," The Journal of Korean association of computer education, Vol. 18, No. 1, pp.45-54, 2015.
- [12] Ministry of education, Practical Arts(Technology and Home Economics)/Informatics Curriculum, Proclamation of the Ministry of education: #2015-74 [Seperate Volume 10], 2015.
- [13] Ministry of education, Middle School Curriculum, Proclamation of the Ministry of education: #2015-74 [Seperate Volume 3], 2015.
- [14] Korea Institute for Curriculum and Evaluation, 2015 Revised Curriculum of Informatics draft hearing, ORM 2015-56-13, 2015.
- [15] jwchoi, sjan, & yjee, "Computing education in Korea—current issues and endeavors," ACM Transactions on Computing Education (TOCE), Vol. 15, No. 2, pp. 1-21, 2015.
- [16] Computing at School Working Group, Computer Science: A Curriculum for schools. Retrieved from <http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf>. 2012.
- [17] Korea Institute for Curriculum and Evaluation, The Characteristics of Korean Students in Computer and Information Literacy and an Analysis of the Effect of Educational Context Variables, RRE 2015-11-1, 2015.
- [18] Ministry of education, Science Vocational Subject Curriculum, Proclamation of the Ministry of education: #2015-74 [Seperate Volume 20], 2015.
- [19] Ministry of education, Operating instructions of Software Education, 2015.
- [20] Korean association of computer education, An Investigation of Current State and Requirement of Software Education. Retrieved from <http://board-2.blueweb.co.kr/board.cgi?id=mhchung&uname=news&unum=157&action=view>. 2014
- [21] skshin & ykbae, "Review of Software Education based on the Coding in Finland," JOURNAL OF The Korean Association of information Education, Vol. 19, No. 1, pp. 127-138, 2016.
- [22] jmkim & wglee, "China`s informatics curriculum to

- consider equality and excellence,” The Journal of Korean association of computer education, Vol. 18 No. 2, pp. 11-20, 2015.
- [23] Korea Institute for International Economic Policy, Organization and Operation of Elementary and Secondary School Curriculum in China, RRE 12-49, 2012.
- [24] Ministry of Education, Science and Technology, Practical Arts(Technology and Home Economics) Curriculum. Proclamation of the Ministry of education, science and technology: #2011-361 [Seperate Volume 10], 2011.
- [25] Ministry of Education, Science and Technology, Elective Subject Curriculum. Proclamation of the Ministry of education, science and technology: #2011-361 [Seperate Volume 10], 2011.
- [26] Wing, J. M., “Computational thinking”, Communications of the ACM, Vol. 49, No. 3, pp. 33-35, 2006.
- [27] swkim, & yjlee, “Development of the software education curriculum model for secondary education in Korea”, EDULEARN16 Proceedings, pp. 9129-9138, 2016.

Authors



Seong-Won Kim received the B.S. degree in Computer Education from Korea National University of Education, Korea in 2013. He received the M.S. degree in Biology Education from Seoul National University in 2015.

Mr. Kim is currently a doctoral course student in the Department of Computer Education, Korea National University of Education. He is interested in informatics education, robot programming education, STEAM education, and TPACK.



Youngjun Lee received the B.S. degree in Computer Science from Korea University, Korea, in 1988. He received the Ph.D. degree in Computer Science from the University of Minnesota, Minneapolis, in 1994.

He is currently a Professor in the Department of Computer Education, Korea National University of Education. His research interests include intelligent system, learning science, informatics education, technology & engineering education.