

Development of Digital Contents to Improve Computational Thinking

Mi-Young Ryu*, Sun-Gwan Han**

Abstract

The purpose of this study is to design and develop of digital contents to improve computational thinking in the online education environment. First, we planned the design and development of contents with 19 experts of Software education. Digital content was designed from the point of view of improving the educational quality and the quality of contents for the improve of computing thinking. The content type is classified into the SW education area; computer science, programming, physical computing, convergent computing, computing thinking, and software education that improves the computing thinking. And we designed 45 learning programs for each SW education area. Designed learning contents were developed in 464 lessons to suit the online education environment. The content validity of the proposed content was verified by the expert group and the average CVI value was over .83. Through this, we could analyze that the developed contents will help learners to expand their computing thinking.

▶ Keyword: Digital Contents, Online Education System, Software Education, Computational Thinking

I. Introduction

디지털 기술과 SW를 중심으로 산업이 재편되면서 전 세계에서 코딩교육을 강화하고 있다. 영국은 2014년부터 초·중등 교육과정에 '컴퓨팅' 교육을 필수화하였고, 미국은 K-12 컴퓨터 과학 표준에 컴퓨팅 사고력을 명시하였다. 프랑스는 컴퓨터 교육을 중학교 정규 과목으로 개편하였으며, 인도는 '프로그래밍' 과목을 초·중등 필수 과목으로 지정하였다. 그리고 에스토니아는 2015년부터, 핀란드는 2016년부터 모든 초등학생을 대상으로 코딩교육을 실시하는 등 많은 나라들이 초등학교에서부터 정보 교육을 실시하고 있으며, 기존의 ICT 활용 중심에서 알고리즘을 중심으로 한 코딩교육으로 전환하고 있다[7].

우리나라에서는 2013년 창조경제 7대 실천전략 중 SW 인력 양성이 발표되고 2014년 SW중심사회 실현 정책을 추진하면서 SW교육이 주요 이슈로 제시되었다. 그 정책에는 초·중등 SW교육 활성화를 위해 '정보'교과의 신설을 포함하였다[8].

이에 교육부가 주도적으로 국가수준의 교육과정을 편성하며 모든 초·중등 학교의 교육과정에 편성하여 2018년부터 정보교

과를 운영할 예정이다. 새로운 교육과정을 적용하기에 앞서 일부 학교를 시범학교로 선정하여 운영하는데 소프트웨어교육은 기존의 정보교육과는 다른 성격으로 도입이 되다 보니 학교에서의 적용 할 수 있는 환경 구성뿐만이 아니라 교사 연수 프로그램과 학습자가 스스로 수업하기 위한 교육 콘텐츠가 매우 부족한 상황이다[10,11,14].

또한 소프트웨어교육이 단순히 프로그래밍을 하고 코딩 기법을 안내하거나 컴퓨터 과학 지식을 이해하는 교과 성격이라기보다 컴퓨팅 사고(Computational Thinking, 이하 CT)를 바탕으로 창의적인 문제해결력을 신장하는데 교육의 목표를 두고 있기 때문에 그에 맞는 교육 콘텐츠가 필요한 실정이다.

이러한 목표를 달성하기 위해 단기간에 초·중등 교사들과 학생들에게 SW교육을 전달하기 위한 방법은 오프라인 교재와 연수와 캠프 등의 교육실습보다는 온라인으로 제공하는 교육 콘텐츠와 교육 시스템이 효과적이다[13].

따라서 본 연구에서는 CT를 신장하기 위한 교육 자료로 디

*First Author: Mi-Young Ryu, Corresponding Author: Sun-Gwan Han

*Mi-Young Ryu (ddochi29@naver.com), Dept. of Computer Education, GyeongIn National University of Education

**Sun-Gwan Han (han@gin.ac.kr), Dept. of Computer Education, GyeongIn National University of Education

• Received: 2017. 11. 20, Revised: 2017. 11. 24, Accepted: 2017. 11. 28.

지털 콘텐츠를 구성하고 온라인 교육 시스템 환경에서 SW교육을 지원할 수 있는 다양한 콘텐츠를 개발하고자 한다.

II. Preliminaries

1. Backgrounds

1.1 Software Education

SW교육의 현장 정착을 위해 교육부에서는 2018년부터 공교육에서 SW교육을 의무적으로 배우도록 하고 각 유관기관으로 과학기술정보통신부, 교육부, 지역교육청 그리고 민간 연구소와 기업이 공조하여 진행을 하고 있다. 과학기술정보통신부는 한국과학창의재단을 실행기관으로 선도학교를 운영하고 교육부는 한국교육학술정보원을 실행기관으로 하여 연구학교를 운영하였다. 그리고 이에 따른 SW교육 교사 연수 프로그램을 개발하여 현장에 적용하고 있으며 이러한 노력이 시도교육청에서 함께 참여하여 확산하고 있다. 또한 민간 기관으로 교육관련 연구소에서 온, 오프라인 연수 프로그램을 제공하고 있고 민간 기업에서도 각각의 SW교육의 철학을 바탕으로 학생 교육과 교사 연수를 지원하고 있다[13].

2018년 시작되는 SW교육 정책은 매우 구체적이다. 초등학교에서는 실과 교과 내에 17시간 이상 SW교육을 실시할 수 있도록 하였다. 중학교에서는 '정보'교과를 신설하여 모든 학생들이 72시간을 배우도록 하였다. 고등학교는 심화 선택이었던 '정보'를 일반선택 과목으로 전환하고 그 내용 역시 SW를 중심으로 개편하였다[7].

1.2 Computational Thinking and SW Education

일반적으로 컴퓨팅 사고는 문제를 해결하는 능력으로 정의하고 다른 교과와는 다르게 컴퓨팅 역량을 활용하여 현실의 문제를 해결하는 사고력으로 보고 있다.

자넷 윙이 제시한 CT는 문제를 수립하고 해결책을 만들어 컴퓨팅 시스템을 통해 효과적으로 수행되도록 표현하는 사고 과정으로 정의하고 있으며 문제해결, 시스템설계, 인간행동과 지능의 이해를 포함하는 개념이라고 하였다[4]. 이를 간단하게 추상화(Abstraction)와 자동화(Automation) 능력으로 표현하였다[5].

이러한 두 가지 컴퓨팅 사고의 요소를 구글(Google)에서는 4가지 요소로 확대하여 분해(Decomposition), 패턴인식(Pattern Recognition), 추상화(Abstraction), 알고리즘(Algorithm) 그리고 자동화(Automation)으로 세분화하였다[20]. 이러한 세분화는 구글에서 필요로 하는 인재, 특히 코딩과 프로그래밍의 전문가와 컴퓨터 과학자의 문제해결 과정을 분석하여 구체화한 절차형 컴퓨팅 사고 모델이라고 할 수 있다.

이후 CSTA(Computer Science of Teachers Association)와 ISTE(International Society for Technology in Education)의 두 기관에서 발표한 CT는 구글의 CT 구성요소

를 더욱 세분화하여 확장하였다[17]. 두 기관에서 제시한 컴퓨팅 사고의 핵심 개념과 능력은 자료수집(Data Collection), 자료분석(Data Analysis), 자료표현(Data Representation), 문제분해(Problem Decomposition), 추상화(Abstraction), 알고리즘과 절차(Algorithms-Procedures), 자동화(Automation), 시뮬레이션(Simulation), 병렬화(Parallelization)의 9가지 개념으로 제시하고 그 개념을 K-12의 컴퓨터과학 표준 교육과정에 포함시켜 구체적인 사례를 제시하였다[14].

2. Related Works

온라인 사이트를 활용하여 SW교육을 활성화시키고자 하는 노력은 각 나라에서도 활발히 이루어지고 있다.

스크래치 온라인 커뮤니티(<http://scratch.mit.edu>)는 전 세계 스크래치 개발자들이 프로젝트를 공유하고 협력, 토론 등의 상호작용이 가능한 환경을 제공하고 있다. Code.org는 전 세계적인 SW교육용 웹기반 플랫폼으로 코딩교육을 무료로 제공하고 있다. 학생, 교사, 일반인 모두를 대상으로 하며, 컴퓨터과학 원리를 언플러그드 활동과 블록프로그래밍 형식의 코딩 학습 두 가지 형태로 학습내용을 제공하고 있다.

영국의 경우 교사 연수를 위하여 온라인 사이트를 활용하고 있다. 거점(Hub) 학교와 선도 교사(Master Teacher)를 선발하여 이들이 각자 자신의 소속지역의 교사들을 도와주는 방법을 이용한다. 공적인 연수 방법 외에도 컴퓨터 교사 연합(Computer At School, CAS)에서 운영하는 커뮤니티에는 교사들이 주축이 되어 서로의 경험과 노하우를 나누고, 직접 제작하거나 외부에서 가져온 방대한 분량의 교수-학습자료들을 온라인으로 공유한다. 컴퓨터 과학의 지식 평가를 위한 새로운 시도로 비브라스 챌린지(Bebras Challenge)가 있다. 컴퓨터 과학의 지식을 현실 세계의 문제로 구성하여 학습자들이 기초 CS 지식이 없이도 CS에서 필요한 지식과 알고리즘 역량을 측정하는 평가문항을 구성하여 제공하고 있다.

이러한 사이트들의 특징은 각기 제공하고자 하는 영역이 구분되어 통합적으로 CT를 신장시키기 위한 다양한 콘텐츠가 부족하다는 점이다. 컴퓨터 과학 지식 위주의 사이트, 코딩 실습 위주의 사이트 등으로 구성되며 학습자나 교사들이 컴퓨터 과학 지식과 코딩 기능, 컴퓨팅 프로젝트 등의 다양한 학습 콘텐츠를 통해 CT를 통한 문제해결력을 신장시키기 위해 사이트를 찾아보기는 힘들다.

서봉수와 박성균은 프로그래밍 학습을 위한 웹콘텐츠의 개발에 관한 연구에서 C와 Java 프로그래밍 언어를 학습하기 위해 웹 콘텐츠의 효과성을 분석하고 학생들의 이해도, 만족도 및 프로그래밍 능력이 현저히 향상되었음을 밝혔다[1]. 송의성과 길준민은 컴퓨팅 사고 향상을 위한 블렌디드 학습기반의 교육 프로그램을 개발하는 연구를 통해 교육용 프로그래밍에 관한 블렌디드 학습의 콘텐츠를 통해 학습자들이 소프트웨어교육에 대한 인식 수준이 높아지고 컴퓨팅 사고력의 자가 평가에서도 신장됨을 밝혔다[2]. 김경미와 김현숙은 디지털시대에 플립드

러닝을 활용한 학습자 맞춤형 소프트웨어 교육 방안 연구를 통하여 프로그래밍의 초보학습자들에게 적절한 플립드 러닝 교육 콘텐츠와 운영 방법을 설계하여 적용하였다. 이를 통해 온라인 교육과 오프라인 교육에서의 적용 전략과 교육과정 설계의 방안을 제시하고 수업의 과정별 고려사항에 대해 제안하였다[6]. 서정현과 김영식은 초등학생의 융합적 산출물 기반 소프트웨어 교육용 콘텐츠 개발에 관한 연구를 통하여 컴퓨팅 사고를 신장시키기 위한 교육용 콘텐츠로서 CT의 개념을 활용한 융합적 산출 기반 프로그램을 개발하고 학생들에게 적용하여 긍정적 영향을 주는 것을 밝혔다[3].

이상의 선행연구를 분석해보면 많은 연구자들이 소프트웨어 교육으로서 컴퓨팅 사고를 신장시키기 위한 교육 콘텐츠들을 개발하여 적용함으로써 그 효과를 밝히고 있으나 단편적인 프로그래밍 언어 한가지나 제한된 학습방법을 통해 접근하고 있다. 이처럼 통합적인 교육 영역과 교육 콘텐츠의 설계와 개발에 관한 연구는 많이 부족한 것으로 분석되었다.

III. Design and Development of Digital Contents

1. Overview of Contents Design

1.1 Process of Study

연구절차 및 콘텐츠 개발의 절차는 다음과 같이 구성하였다. 개발의 단계는 소프트웨어공학과 교육공학의 콘텐츠개발 단계인 ADDIE 모형에서 개발과 구현을 통합하여 4단계로 구성하였다.

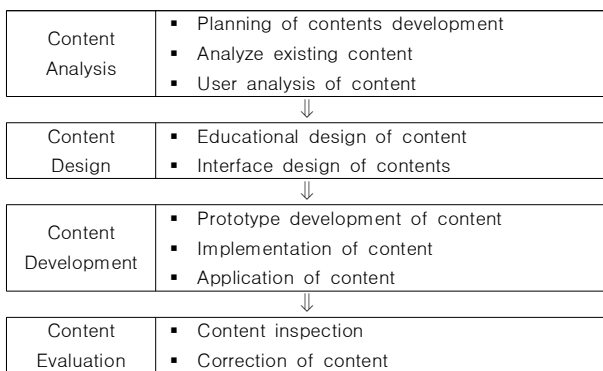


Fig. 1. Development Process of Educational Contents

우선 콘텐츠 개발을 위한 기초 분석과 기획을 위해 전문가를 구성하였다. 컴퓨터교육학, 컴퓨터공학, 교육공학, 교육콘텐츠 개발 전문가, 소프트웨어교육 전문가를 포함하여 총 19명이 참가하였다. 전문가 집단과 3회에 걸쳐 콘텐츠의 설계 및 개발 기획을 실시하였다. 브레인스토밍, SWOT분석을 통해 콘텐츠의 성격을 규정하고 미디어의 특징, 학습관리 시스템과의 연동에 대한 논의를 실시하였다. 그리고 기존 개발된 온라인 교육

콘텐츠를 분석하고 코딩교육의 사용자로 초중등학생과 교사를 중심으로 사용자 분석을 하였다.

콘텐츠의 설계는 교육적인 요소의 설계와 함께 시스템과의 인터페이스 디자인 그리고 학습 방법론에 따른 미디어 구성에 관한 설계를 실시하였다. 콘텐츠 교육요소 설계는 교육목표의 적절성, 교육과정의 구성, 교육내용에 따른 세부 콘텐츠 목록, 학습의 단계에 따른 교수학습 전략, 그리고 학습 평가 전략 구성을 중심으로 설계에 반영하도록 하였다. 설계된 내용은 소프트웨어교육 전문가들에게 타당도 검사를 통하여 적합성을 확인하였다.

콘텐츠의 전체 개발에 앞서 코딩교육을 위한 교육 콘텐츠의 프로토타입을 개발하고 소프트웨어 전문가들을 대상으로 적합성을 검증받고 교육과정에 맞는 콘텐츠를 구현하였다. 구현된 콘텐츠는 교사와 학생들에게 적용하였으며 콘텐츠 검사와 피드백을 통하여 콘텐츠를 수정, 보완하였다.

1.2 Considerations for Design

교육 콘텐츠는 수업의 목적을 달성시킬 수 있도록 효율적이고 구체적으로 구성되어야 한다. 온라인 학습 환경에서 사용되는 디지털 콘텐츠의 경우에는 학습이 지향하는 목표 즉 컴퓨팅 사고를 신장시키는가와 콘텐츠의 학습 적합성 즉, 시스템과 사용자, 학습 환경과 기기 등을 고려하여 설계 하여야 한다. 컴퓨팅 사고를 신장시키기 위한 교육 콘텐츠의 조건을 15가지로 설계하였다. 그 조건은 교육목표, 교육과정, 교육내용, 교육전략, 교육평가, 학습자 수준, 교사 수준, 콘텐츠 유형, 콘텐츠 학습 시간, 인터페이스, 학습 시스템 연동, 학습환경, 상호작용, 학습 자료, 유지보수이다.

15가지 고려사항 중 1~7사항은 콘텐츠가 학습자의 컴퓨팅 사고를 신장시키는가에 대한 교육적 분석부분이고 8~15번 사항은 온라인 학습에서 콘텐츠가 적합한지에 대한 부분이다. 본 연구에서는 이러한 사항을 고려하여 디지털 콘텐츠를 설계의 기준으로 삼고 전문가들의 타당도 검사를 통하여 그 기준에 맞게 개발하였다.

2. Design of Digital Contents

2.1 Curriculum and Contents

컴퓨팅 사고를 신장시키기 위한 영역을 크게 6가지로 구분하여 제시하였다. 6가지로 구분하는 기준은 미래인재연구소에서 제시한 창의컴퓨팅 교육 영역을 바탕으로 영역을 선정하여 개발에 참여한 19명의 전문가들이 브레인스토밍과 SWOT 검사를 통하여 선정하였다.

각 영역은 컴퓨팅 사고를 신장시키기 위한 지식 개념과 기능 실습 영역을 학습하는 콘텐츠로 구성된다. 태도와 가치에 관한 부분은 각 콘텐츠의 지식과 기능 학습 중에 발문과 자료의 제시를 통하여 자연스럽게 학습하도록 하였다. 각 영역에 따른 세부 교육내용의 콘텐츠는 문헌 분석과 소프트웨어교육 선도학교의 운영사례 그리고 영국 및 미국의 SW교육 과정을 바탕으로 분석하였다.

Table 1. Education area of content

Area	Division	Contents
Computer Science	Knowledge, Attitude	Knowledge, concept, theory, procedural principle, algorithm of computer science
Programming	Skill, Attitude	Coding Basics Various programming
Physical Computing	Skill, Attitude	Project type development Coding practice
Computing Thinking	Knowledge, Skill	Computing Thinking Theory Coding practice
Software Education	Knowledge, Attitude	Introduction to Software Education Educational contents, education methods, evaluation
Convergent computing	Skill, Attitude	Coding for other subjects Solving real life problems

컴퓨터과학의 강좌 내용은 주로 CS 지식에 관련된 개념을 학습하는데 딱딱한 강좌의 내용 전달보다는 놀이를 통한 원리와 알고리즘의 발견을 하도록 설계하였다. 언플러그드 CS 활동과 비브라스 평가 문항 등을 통한 실생활 예시 문제를 해결함으로써 자연스럽게 컴퓨터 과학을 이해하도록 하였다. 프로그래밍의 영역은 초중등학생들이 쉽게 코딩을 할 수 있는 교육용 프로그래밍 언어 즉, 엔트리, 스크래치, 스텝, 앱인벤터, 포켓코드 등의 블록형 코딩 언어를 주로 사용하여 학습하도록 설계하였으며 스크립트 언어와의 연계를 위하여 스몰베이직, 파이썬 그리고 HTML을 학습하도록 내용을 선정하여 설계하였다. 퍼지컬 컴퓨팅 영역은 아두이노를 바탕으로 기초, 초급, 중급, 고급 콘텐츠를 구성하고 상업적으로 사용하기 편리한 교구제를 선정하여 다양한 메이커 활동이 가능한 콘텐츠로 설계하였다.

융합 컴퓨팅 영역은 앞서 배운 교육 프로그램을 바탕으로 코딩과 컴퓨터 과학을 이해하도록 타고과 교육과정과 연계하여 구성하였다. 컴퓨팅 사고 영역은 실생활의 문제해결전략을 위주로 정보기술과의 연계 및 코딩을 활용한 창의적인 사고를 이끌어 내도록 구성하였다. 소프트웨어교육 영역은 초중등 학생들을 대상으로 구성하지 않고 교사들을 대상으로 설계하였다. 실제 소프트웨어교육의 필요성과 인식 전환 그리고 그에 따른 기초 이론을 제시하고 학생들을 가르치기 위한 교수학습 방법 중 한국교육학술정보원에서 개발된 모형을 중심으로 선정하여 설계하였다.

2.2 Educational Strategy and Evaluation

컴퓨팅 사고를 신장시키기 위한 교육 전략은 콘텐츠의 학습 유형에 따라 다르게 설계하였다. 지식위주의 학습 콘텐츠, 예를 들면 컴퓨터 과학 등의 콘텐츠는 학습자가 어려움을 극복하기 위해 스스로 개념이나 알고리즘의 원리를 발견할 수 있도록 발견탐구학습의 전략과 놀이학습 전략을 사용하여 설계하였다. 기능 영역의 학습 콘텐츠는 기초가 되는 직접교수법과 문제해결학습 그리고 프로젝트 학습을 통해 기능을 이해하고 스스로 적용하여 실생활의 문제를 해결하도록 구성하였다. 태도 영역의 학습콘텐츠는 단위모듈로 구성되지 않았지만 지식과 기능의

학습콘텐츠 내에 일부 발문과 학습 요소로 제시되어 가치갈등 또는 토론학습의 방법으로 문제를 해결하도록 설계하였다.

Table 2. Educational strategy of content

Div	Teaching and Learning Strategy	Learning Procedure	Example content
Knowledge	Discovery inquiry learning method Learning to play	Task recognition → Rule discovery → Generalization → Apply	Unplugged CS Bebras Computing
Skill	Direct teaching Problem solving method Design learning method Project learning method	Presentation of the task → Demonstration explanation → Follow-up → Expansion Problem-solving → solution strategy → problem solving → generalization	Scratch foundation CT recipe design Physical Computing
Attitude	Value conflict learning method Discussion learning method Value inquiry learning method	Present situation → Problem Discussion → Solution strategy → Valuation	Computing Thinking Convergent computing

콘텐츠 내에서 평가는 가급적 객관식 형태의 단순 지식을 묻는 평가 방법은 지양하고 과정 평가 또는 프로젝트형으로 대변되는 수행평가 형식으로 제공하여 학습콘텐츠에서 배운 내용을 스스로 만들어보고 제시된 문제를 해결하여 스크래치 또는 엔트리 등의 사이트에 결과물을 올려 다른 학습자들과 공유하고 피드백을 받는 형태로 구성하였다.

2.3 Suitable of Education Contents

디지털 콘텐츠가 학습자들에게 편리하게 학습을 제공하기 위해서는 사용자의 유형과 수준 그리고 미디어의 특징, 학습시간, 인터페이스 디자인, 학습관리시스템과의 연계 그리고 학습하는 환경으로서 사용기와 운영체제 등에 대한 설계가 이루어져야 한다. 또한 사용자와의 상호작용이 원활하게 되기 위한 인터랙션 부분과 자체 콘텐츠 외에 부가적으로 제공되는 교안 또는 학습자료가 필요하며 향후 업그레이드와 유지보수를 위한 설계를 하였다. 콘텐츠의 유형은 동영상을 바탕으로 하는 콘텐츠로 개발하되 저장공간과 네트워크 대역폭의 부담을 줄이기 위해 클라우드 시스템과 연동하여 그 문제를 해결하도록 콘텐츠를 설계하였다.

사용자와의 상호작용을 제공하기 위해 공개형 MOOC시스템의 기술과 오픈소스를 이용하여 구성하되 단위 학습 콘텐츠별로 작게 모듈화하여 학습 화면 내에 학습화면을 제시하고 진행 상황을 학습자가 스스로 조절하도록 하며 부가 학습자료를 제공하도록 설계하였다. 또한 학습 내용에 대한 질의와 결과물 제출을 위한 게시판의 기능을 제공하여 상호작용을 쉽게 하도록 설계하였다.

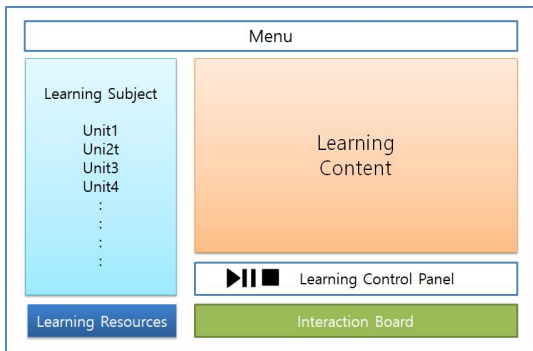


Fig. 2. Interface Design

3. Development of Content

설계된 콘텐츠는 2014년부터 2016년까지 전문가 19명이 참여하여 단위모듈을 가진 애니메이션 자료와 동영상자료의 형태로 개발하였다. 애니메이션 학습콘텐츠의 경우 교육콘텐츠 개발 전문팀에 의해 개발하였으며, 동영상 학습콘텐츠는 소프트웨어교육의 경험이 있고 초중등교사 연수 경험이 풍부한 현장 교사가 참여하여 개발하였다. 그림 3은 온라인 사이트의 메인화면과 콘텐츠의 구성 화면이다. 콘텐츠의 구성은 전문가 집단의 설계 과정을 거쳐 선정된 컴퓨터과학, 프로그래밍, 피지컬 컴퓨팅, 컴퓨팅 사고력, 소프트웨어 교육, 융합 컴퓨팅의 6개 영역을 중심으로 하여 강좌를 구분하여 개발하였다.



Fig. 3. Interface of System and Contents

애니메이션 콘텐츠는 플래시를 사용하였으며 SCORM방식을 적용하여 학습 모듈을 분리하여 재활용할 수 있도록 개발하였다. 스마트 기기에서 구동이 되지 않는 관계로 초기 개발에 이루어졌으며 이후 개발은 클라우드에 탑재하기 편리하고 모든 스마트 기기와 PC, 랩탑 등에서 구동이 가능한 동영상 MP4 파일 타입으로 개발되었다.

개발된 교육 콘텐츠는 총 45개 강좌, 464차시를 개발하였다. 각 교육 영역별 콘텐츠는 블록형 프로그래밍의 특징을 활용하여 명령어 군의 색을 다르게 한 것을 이용하여 콘텐츠별로 교유의 색상을 갖도록 아이콘을 구성하였다. 이는 학습자들이 명령어 문법의 특징을 사이트와 콘텐츠에서도 잠재적으로 받아들일도록 설계

한 결과이다. 세부 콘텐츠의 목록은 <http://koreasw.org> 사이트에서 볼 수 있다.

콘텐츠의 학습 인터페이스는 설계에서 구성한 것과 같이 상단의 메뉴와 함께 화면 좌측에 학습 목차를 제시하여 단위학습을 사용자가 쉽게 선택하도록 하였다. 화면 중앙에 학습콘텐츠를 배치하여 학습에 집중하도록 하며 콘텐츠 하단에 콘텐츠 제어 모듈을 배치하여 학습을 사용자가 조절할 수 있도록 하였다. 콘텐츠 하단에는 학습 자료를 제공하고 화면 하단에 보편적인 SNS와 상용사이트를 활용하여 다른 학습자들과 공유하기 쉽도록 링크를 제공하고 있다. 또한 학습한 내용에 대한 질문과 학습 자료를 저장하기 위한 게시판 보드를 제공하여 상호작용이 쉽게 될 수 있도록 구성하였다.

콘텐츠의 학습 절차는 지식과 기능 그리고 태도의 학습 영역에 따라 다양하게 구성되 단위 모듈의 학습 콘텐츠임을 고려하여 기본 3단계의 수업틀을 적용하였다. 3단계는 동기유발 & 학습목표 제시- 학습하기(놀이를 통한 CT신장활동)-평가로 이루어지며 각각의 교수학습전략에 따라 학습하기의 내용이 직접 교수나 문제해결, 프로젝트 개발 등으로 다양하게 변형되어 개발되었다.

IV. Analysis and Results of Study

1. Validity Analysis of Content

개발된 콘텐츠에 대해 전문가들의 의견을 통해 개발의 기획과 내용이 타당한지 내용타당도 검사를 실시하여 분석하였다. 검사에 참여한 전문가 집단은 컴퓨터교육 전공자, 교육공학 전공자, 컴퓨터 공학 전공자, 소프트웨어교육선도학교 교사들로 구성되어 총 22명의 답변을 받아 분석하였다.

내용 타당도 검사 문항은 2개의 큰 카테고리로 구분되어 컴퓨팅 사고 신장을 위한 교육 콘텐츠의 적절성(1~7번 문항)과 콘텐츠 학습품질의 적합성(8~15번 문항)에 관한 부분으로 구분하여 15문항으로 구성되었으며, 세부적으로는 교육과정과 교육내용 그리고 교육전략은 영역별로 나누어 총 30문항으로 구성되었다.

2. Results of Validity Analysis

타당도의 검사는 전문가의 의견을 바탕으로 분석을 하기 때문에 내용 타당도 검사로 실시했으며 검사 문항에 대해 전문가 사이에서 동의하는 정도 즉, CVI 지수(Content Validity Index)를 사용하여 타당도를 산출하였다. 본 연구에서처럼 전문가의 수가 6명 이상인 경우 CVI 값이 .78 이상이어야 타당하다고 볼 수 있다. 내용 타당도 분석결과 CVI 값이 최소 .68에서 최대 .98을 나타냈고 총 CVI값의 평균이 .83 이상으로 나타나 타당도가 높다고 볼 수 있다.

Table 3. Test results of Content Validity Items(n=22)

No	Item	CVI value
1	Educational Objectives for CT	.91*
2	Curriculum for CT (total)	.88*
	Computer Science	.79*
	Programming	.94*
	Physical Computing	.92*
	Convergent computing	.91*
	Computing Thinking	.95*
	Software Education	.77
3	Educational contents for CT (total)	.86*
	Computer Science	.81*
	Programming	.98*
	Physical Computing	.95*
	Convergent computing	.89*
	Computing Thinking	.79*
	Software Education	.75
4	Educational Strategy for CT (Total)	.79*
	Knowledge area content	.81*
	Skill area content	.88*
	Attitude area content	.70
5	Educational Evaluation for CT	.68
6	Level of Learners for CT	.79*
7	Level of Instructor for CT	.81*
8	Appropriateness of content type	.74
9	Appropriateness of content learning time	.89*
10	Relevance of the learning interface	.88*
11	Relevance of interworking with learning system	.85*
12	Appropriateness of learning environment	.91*
13	Relevance of interaction	.79*
14	Appropriateness of learning resource provision	.80*
15	Appropriateness of maintain	.88*

* : CVI > .78

세부적인 문항의 타당도를 분석해보면 CT를 위한 교육과정의 영역의 세부항목에서 소프트웨어 교육 영역이 .77로 나타났다. 이것은 콘텐츠가 학생용이 아닌 교사용의 교육자료로 개발하여 컴퓨팅 사고를 신장시키는 내용과는 다른 것으로 해석한 것으로 추론된다. CT를 위한 교육내용의 경우에도 마찬가지로 .75로 나타났는데 앞서 추론한 내용과 비슷한 결과로 보인다. CT를 위한 교육전략의 영역에서는 태도 영역의 값이 .70으로 낮게 나타났다. 이것은 학습콘텐츠 중에서 태도를 직접적으로 다루는 콘텐츠가 없기 때문인 것으로 보인다. 이 세 가지의 낮은 CVI 값에도 불구하고 총합을 하여 평균을 낸 상위 카테고리의 검사 문항이 각각 .88, .86, .79로 나타나 CT를 신장시키기 위한 콘텐츠의 교육과정, 교육내용, 교육전략의 전반의 타당도의 값은 적절한 것으로 분석되었다.

그리고 CT를 위한 교육평가의 CVI지수가 .68로 낮게 나왔는데 이것은 콘텐츠 내에 포함된 평가 문항의 유형이 주로 발문형, 프로젝트 개발자료 평가형, 자기주도학습 평가형이 주를 이루면서 평가 결과를 측정하기 어렵다는 의견이 많았다. 또한

이것은 콘텐츠 말고도 시스템의 기능적인 부분을 개선해야 한다는 의견을 통해 향후 연구에서 평가 시스템의 개발과 연동을 통한 콘텐츠의 학습 결과에 대한 처리 방법에 대한 개발이 이루어져야 함을 알 수 있었다. 콘텐츠 유형의 적절성에 대한 CVI값은 .74로 역시 낮게 나왔는데 주로 학습콘텐츠가 동영상으로 개발되면서 다양한 멀티미디어 자료를 활용해야 한다는 의견을 제시하였다. 앞서 제시한 것처럼 애니메이션 콘텐츠 또는 상호작용이 가능한 멀티미디어 기반 학습콘텐츠의 경우 개발의 어려움과 비용의 부담 그리고 시스템의 기능 개선, 네트워크 부하로 인한 문제점으로 인해 또 다른 방법의 대안을 찾아야 할 것으로 분석되었다.

이상으로 전체적인 타당성 분석 결과를 통해 컴퓨팅 사고를 신장시키기 위한 온라인 SW 교육 콘텐츠의 설계와 개발은 적절한 것으로 분석되었다.

V. Conclusions

본 연구에서는 컴퓨팅 사고를 신장시키기 위한 방법으로 온라인 SW교육 환경에서 학습자 스스로 학습할 수 있는 디지털 콘텐츠의 설계와 개발에 관한 연구 결과를 제시하였다.

우선 SW교육 전문가를 팀으로 구성하여 콘텐츠의 설계와 개발에 대한 전체 기획을 구성하였고 설계에 대한 핵심 고려사항을 선정하였다. 선정된 고려사항은 결국 콘텐츠의 설계와 개발에 대한 타당도 검사 문항으로 이어져 학습자가 컴퓨팅 사고를 신장시키기 위한 교육 설계와 콘텐츠의 학습 품질 향상을 통한 효율적인 학습 콘텐츠 개발로 이어졌다.

설계는 교육적 전략을 바탕으로 6가지의 SW교육 영역 즉, 컴퓨터과학, 프로그래밍, 피지컬 컴퓨팅, 융합 컴퓨팅, 컴퓨팅 사고, 소프트웨어교육으로 선정하여 각 영역별로 세부 교육 내용을 선정하였다. 총 45개 학습 콘텐츠를 설계하여 전문 개발팀과 소프트웨어교육 전문교사의 참여 하에 464час의 콘텐츠를 개발하였다.

제안된 콘텐츠는 전문가 집단의 내용 타당도 검증을 통해 설계와 개발이 매우 적절한 것으로 나타났다. CVI 지수를 분석한 결과 CVI 값이 최소 .68에서 최대 .98을 나타냈고, CVI값의 평균값이 .83 이상으로 나타나 콘텐츠의 설계와 개발은 컴퓨팅 사고를 신장시키는데 많은 도움을 줄 것으로 분석되었다.

서두에서 제시한 것과 같이 SW교육이 초중등 현장에서 정식 교과로 활성화되기 위해서는 교과적인 노력도 필요하지만 교육의 주체인 교사와 학습자의 SW교육에 대한 인식과 기초 소양 그리고 코딩과 컴퓨터 과학의 지식을 통해 컴퓨팅 사고를 충분히 경험하는 것이 중요하다.

이에 본 연구에서 제시하는 온라인 교육체제에서의 학습 활동이 대안으로 대두되고 있으며 그 기초 작업으로 컴퓨팅 사고를 제시하는 온라인 학습 콘텐츠의 개발은 매우 중요하다.

향후 다양한 미디어 형태의 콘텐츠가 개발되어 학습자들과의 상호작용과 함께 맞춤형 서비스가 가능하도록 제시하는 연구가 필요하다. 또한 평가 부분에 있어서 학습자의 수준을 판단하고 학습 결과에 대해 적절하게 측정할 수 있는 평가 시스템과 과정 중심 수행 평가가 가능한 교수전략과 시스템의 개발이 필요하다.

REFERENCES

- [1] B. S. Suh, S.G. Park, "Design and Application of Web Contents for Integrated Programming Language Course", The Journal of Korea Association of Content, Vol 7, No 8, pp.226-233, 2007
- [2] E. S. Song, J. M. Kim, "Development and Application of Software Education Program Based on Blended Learning for Improving Computational Thinking of Pre-Service Elementary Teachers", The Journal of Korea Information Processing Society, Vol 3, No 7, pp.353-361, 2017
- [3] J. H. Seo, Y. S. Jung, "Development and Application of Educational Contents for Software Education based on the Integrative Production for Increasing the IT Competence of Elementary Students", The Journal of Korea Association of Information Education, Vol 20, No 4, pp.357-366, 2016.
- [4] J. M. Wing. "Computational Thinking", Communication of the ACM, Vol. 49, No. 3, pp.33-35, 2006.
- [5] J. M. Wing. "Computational thinking and thinking about computing". Philosophical transactions of the Royal Society, Vol. 366, pp.3717-3725, 2008.
- [6] K. M. Kim, H. S. Kim, "A Study on Customized Software Education method using Flipped Learning in the Digital Age", The Journal of Korea Association of Digital Policy, Vol 15, No 7, pp.55-64, 2017.
- [7] Ministry of Education, KERIS, "Guidelines for Operating Software Education", 2015.
- [8] Ministry of Science, ICT and Future Plan, "SW-centered Society and Software Education", SW Policy Reports, 2015.
- [9] M. Y. Ryu, S. K. Han, "Analysis of Software Image using Semantic Differential Scale in Elementary School Students". Journal of The Korean Association of Information Education, Vol 20, No 5, pp.527-534, 2016 .
- [10] M. Y. Ryu, S. K. Han. "The Structural Equation Modeling of Factors Affecting the Parent Willingness on Child's Software Education". Journal of The Korean Association of Information Education, Vol 20, No 5, pp.443-450, 2016.
- [11] S. B. Shin, "The Improvement Effectiveness of Computational Thinking through Scratch Education", Journal of the Korea Society of Computer and Information, Vol 20, No 11, pp.191-197, 2015.
- [12] S. K. Han, M. Y. Ryu, "Software Education", Seoul: Saengreung-Press, 2016.
- [13] S. K. Han, Teaching and Learning Strategies for SW Education, Creative Computing Issue Report 2015-1,2 2016-3, 2015, <http://computing.or.kr>
- [14] S. W. Kim, Y. J. Lee, "Development of a Software Education Curriculum for Secondary Schools", Computer and Information, Vol 21, No 8, pp.127-141, 2016.

Authors



Mi Young Ryu received the Master degrees in 2015 and is processing Doctoral course in Computer Education from GyeongIn National University of Education, Korea. Ms. Ryu joined the faculty of the Future Talent Institute at GyeongIn National

University of Education, Incheon, Korea, in 2015. She is currently a Teaching Professor in the Department of Computer Education, GyeongIn National University of Education. She is interested in Convergence Education, Software Education and Online Education.



Sun Gwan Han received the Ph.D. degrees in Computer Science and Engineering from Inha University, Korea, in 2002. Dr. Han joined the faculty of the Department of Computer Education at GyeongIn National University of Education, Incheon,

Korea, in 2002. He is currently a Professor in the Department of Computer Education, GyeongIn National University of Education. He is interested in Computer Education, Software Education and AI.