

컴퓨팅 사고력(Computational Thinking) 함양을 위한 대학에서의 SW교육에 관한 고찰

박성희
가천대학교 교육대학원

Study of SW Education in University to enhance Computational Thinking

Sung Hee Park
Graduate School of Education, Gachon University

요약 사회 전체가 소프트웨어 중심으로 운영되고, 사물인터넷을 통한 새로운 디지털 시대가 시작되면서 서로 융복합적으로 연계가 이루어지고 있다. 이러한 사회변화와 함께 실생활의 문제들이 복잡해지고 의사소통 능력과 문제해결능력이 점차 강조됨에 따라 이를 교육과정에 반영하기 시작하였다. 이러한 변화는 이미 해외에서 먼저 시작되었고, 국내에서는 2015년도 교육과정 개정을 통해 초중등에서도 SW교육을 교육과정에 포함하게 되었다. 반면 대학생들을 위한 SW교육에 대한 필요성은 최근에서야 그 움직임이 서서히 시작되었다. 현재 SW교육은 단순 프로그래밍교육이 아닌 컴퓨팅 사고력(Computational Thinking) 증진을 목표로 진행되고 있다. 따라서 최근 주목받고 있는 컴퓨팅 사고력에 대한 동향분석과 함께 대학SW교육에서 컴퓨팅 사고력 증진을 위해서 국내외 대학들이 관련 코스들을 어떻게 운영하고 있는지 분석하고, 컴퓨팅 사고력을 효과적으로 함양할 수 있는 교수학습모형을 제안하였다.

주제어 : 컴퓨팅 사고력, 대학 SW교육, 정보화 교육, 프로그래밍, 프로젝트학습

Abstract Society is operating with software and a new digital era through the Internet of Things started. A variety of fields are being in conjunction with each other based on computing. As problems in real life become more complicated and communication based on various knowledge and problem solving skills are emphasized, these changes are reflected in the curriculum. These changes started from overseas in advance then Korea includes SW education in elementary and secondary education through curriculum revision of 2015. On the other hand, SW education for university students just started after the curriculum revision of elementary and secondary education. The new SW education highlights and develops Computational Thinking beyond programming and it will be a key for the future. Therefore, this study analyzed trends of Computational Thinking and examples of CT courses in Universities. Suggestions and ideas for instructional model to develop Computational Thinking were discussed.

Key Words : Computational Thinking, University SW education, ICT education, Programming, Project-based learning

Received 21 December 2015, Revised 30 March 2016
Accepted 20 April 2016, Published 28 April 2016
Corresponding Author: Sung Hee Park
(Graduate School of Education, Gachon University)
Email: sungheepark@gachon.ac.kr

ISSN: 1738-1916

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

20세기 후반에는 ICT의 발달을 통해 정보통신기술과 관련된 미디어를 잘 활용할 수 테크놀로지 리터러시(technology literacy)의 중요성이 강조되었다. 이와 같은 테크놀로지 리터러시는 스프레드시트를 통해서 장부를 관리하거나 자신의 창의적 아이디어를 그래픽이나 동영상 편집프로그램을 통해 원하는 창의적 결과물을 제작하는 것과 같이 도구적 관점에서 신속하고 정확하게 테크놀로지를 활용할 수 있는 능력을 뜻하였다[1].

그러나 최근 들어 사회 전체가 소프트웨어를 중심으로 운영되는 사물인터넷의 시대가 도래함에 따라 독립되어 존재하던 분야들이 컴퓨팅을 기반으로 융합하는 융복합적 성격을 띠게 되었다. 이와 같은 시대적 흐름에 맞는 인재양성을 위해 세계가 컴퓨터교육에 많은 노력을 하고 있다. 이에 대한 구체적인 예를 들면 다음과 같다. 영국에서는 2000년에 ICT교과를 설치한 이래로 이에 관련한 교육과정을 지속적으로 포함시켰으며, 최근 2014년 9월부터 'Computing'이라는 교과를 포함하여 초, 중, 고에서 필수적으로 이수하도록 하고 있다. 이로써 영어, 수학, 과학, 체육과 더불어 주요 필수교과로 지정된 것이다[2,3].

이와 같이 미래인재 양성을 위해 SW 교육 필요성이 대두됨에 따라 미국에서는 2009년부터 컴퓨터과학교육 주간을 지정하여 이에 대한 중요성을 알리고 있으며, 시카고 교육청의 경우에는 3년 안에 모든 시카고 공립 고등학교에서 컴퓨터과학을 가르치도록 하는 방안을 2014년도에 발표하였다[4,5]. 한편, 핀란드에서는 초등학교부터 프로그래밍교육을 필수교육 지정하여 2016년부터 해당 교육과정을 시작하며, 프랑스에서도 프로그래밍 교육을 교육과정에 포함시키는 방안을 2014년도에 발표하였다[6,7].

이와 같은 추세는 국내에도 반영되어 교육부가 발표한 2014년 9월 교육과정 개정안에는 2015년 문·이과 통합형 교육과정 총론과 함께 소프트웨어 교육강화 내용이 포함되어 있다. 이를 바탕으로 2018년도부터 초등학교 실과수업에서 17시간 이상의 소프트웨어 교육을 실시하고, 중학교에서는 정보교과를 필수과목으로 지정하여 34시간 이상의 소프트웨어 교육을 실시하게 되었다. 고등학교에서는 현재 심화선택인 정보과목을 일반선택으로 전

환하도록 하였다.

특히 국내에서는 SW 전문인력 부족과 함께 SW융합 인력 수요증가로 인해 국내 기업들은 SW인력 확보에 대한 어려움을 호소하고 있는 실정이다. 앞서 소개했듯이 2018년도부터 SW교육이 초·중등교육에서 진행이 될 예정이고 일부 대학에서 자발적인 SW교육을 시키고 있으나, 이것만으로는 SW분야 인력수급으로는 역부족인 상황이다. 이에 교육부와 미래창조과학부가 공동으로 SW 중심사회를 위한 인재양성 추진계획을 2015년 7월에 다음과 같이 발표하였다[8].

초·중등 SW교육 본격 확산, 산업현장의 요구를 반영한 대학 SW교육 혁신, 민관협력으로 친 SW문화확산 등 3대 분야 12개 과제를 청사진으로 제시하였다. 특히 대학을 대상으로는 한 계획에서는 산업현장에서 요구하는 문제해결 역량을 갖춘 창의적 SW인재 향성을 위한 대학 SW교육의 혁신을 도모하며, 이에 일환으로 'SW중심대학' 사업을 2015년도부터 본격적으로 시작하였다. 특히 대학SW교육 중 비전공자들과 관련된 내용을 구체적으로 살펴보면 대학 내 설치된 교양교육 전담기구 등을 활용한 인문계 학생들 대상의 대학 내 SW기초교육을 확대하며, 모든 비전공자들에 대해서도 전공별 특성에 맞는 SW기초교육을 의무화하는 내용이 반영되어 있다.

초·중등교육에서는 이미 다양한 관련 교육분야의 전문가들이 협력하여 다양한 연구 등을 통해 교육과정 수립과 함께 행정적 지원 체계를 마련하는 활발한 활동이 일어나는 반면, 대학교육에서는 SW교육에 관한 연구가 매우 미비한 실정이다. 무엇보다도 개정된 초·중등교육과정을 받지 못하고 이미 대학에 진학해 대학에서만 제공되는 SW교육을 받고, 관련 산업계로 진출할 현재 대학생들의 상황을 감안해본다면, 이에 대한 교육체계나 연구가 매우 시급한 실정이다.

특히 기존의 대학에서 제공한 IT관련 교양교육들이 문서작성, 프리젠테이션, 통계표작성 프로그램 위주로 자격증과 연계한 교육에 국한되었다는 문제점이 지적되면서, 대학교육에서의 정보기술 유창성(information fluency)에서의 지적역량이 크게 강조되고 있다. 이는 문제해결을 위하여 복잡하고 일관된 상황 속에서 IT를 적용할 수 있는 능력을 일컫는데, 이는 현재 활발하게 논의되고 있는 컴퓨팅 사고력과 연결된다고 볼 수 있다. 궁극적으로 컴퓨팅 사고력은 읽기, 쓰기, 셈하기와 같은 기본소양의

3R교육처럼 컴퓨팅사고력이 여기에 추가되어야 한다고 주장되고 있다[9,10]

따라서 본 연구에서는 대학에서의 SW교육, 특히 컴퓨팅 사고력에 관한 국내외 동향분석과 함께 국내외대학에서 현재 시행되고 있는 컴퓨팅 사고력 중심의 SW교육 사례분석을 통해 컴퓨팅 사고력 교육과정에 대한 다양성을 살펴보고, 이에 적합한 교수학습모형을 제안하고자 한다.

2. 컴퓨팅 사고력

2.1 컴퓨팅 사고력 개념 및 특성

앞서 제시된 국내외에서의 SW교육에서의 큰 움직임은 프로그래밍 교육을 통한 사고력 향상에 있으며, 구체적으로 컴퓨팅 사고력 함양에 그 목적이 있다[11,12].

컴퓨팅 사고력은 수십 년에 걸쳐 다양한 이름과 정의를 통해 학문적 논의가 이루어져왔다. 1960년대 Alan Peril은 대학의 모든 전공의 학생들이 프로그래밍을 배워야한다고 하면서 ‘컴퓨테이션 이론’ 교과가 필요하다고 제안하였다[13,14].

Papert(1980)는 1980년대 K-12에서 수학적 개념학습 연구에서 LOGO 프로그래밍 교육을 통한 절차적 사고(procedural thinking)를 향상시킨 것을 입증하였다[15].

이후 Wing(2006)을 통해 이에 관한 학문적 논의가 컴퓨팅 사고력으로 전환되었고, 컴퓨팅 사고력이란 “21세기를 살아가는 모든 사람들이 갖추어야 할 기본적인 사고 능력으로서 컴퓨터 과학의 기본 개념과 원리를 활용한 문제 해결, 시스템 설계, 인간 행동의 이해를 포함하는 추상적 사고 능력”으로 정의하였다. 이를 위해 컴퓨팅 사고력은 로직, 알고리즘 사고, 순환적 사고, 추상화, 병렬적 사고, 패턴 매칭과 과정들을 사용하게 하는 데, 이들 중 가장 높은 수준의 사고과정은 추상화 과정으로 추상화 과정을 통해 복잡성을 줄이고 기본 주요개념의 정의를 설정하는 단계이기 때문이다[10].

Aho(2012)는 좀 더 간략하게 “컴퓨팅 사고력이란 문제들을 공식화(formulating)하는 것에 관여하는 사고과정으로 문제해결안이 컴퓨터화된 단계와 알고리즘으로 표현될 수 있다”라고 정의하였다[16].

궁극적으로 컴퓨팅 사고력은 컴퓨터로 조작가능한 문

제들을 인식하고 능숙하게 문제해결에 접근할 수 있도록 교육함으로써 학습자들이 더욱 효과적인 문제해결자가 될 수 있는 능력을 갖추게 한다[17].

앞서 제시했듯이 컴퓨팅 사고력은 정보과학교육에 화두로 등장하면서 미국에서는 컴퓨터 과학교사 협의회(CSTA)와 ISTE가 공동으로 컴퓨팅 사고력의 구성요소를 선정하여 컴퓨팅 사고력을 통해 K-12에서 모든 교과를 거쳐 문제를 해결하고 시스템을 설계하여 새로운 지식을 창출할 수 있는 컴퓨팅 능력과 한계에 대한 이해를 할 수 있도록 하였다. 다음의 <Table 1>은 CSTA와 ISTE에서 정의한 컴퓨팅 사고력의 주요 9가지 구성요소와 요소별 정의에 대한 설명이다[18].

<Table 1> Components of CT

Component	Definition
data collection	the process of gathering appropriate information
data analysis	making sense of data, finding patterns, and drawing conclusions
Data Representation	Depicting and organizing data in appropriate graphs, charts, words, or images
Problem Decomposition	Break down tasks into smaller manageable parts
Abstraction	reducing complexity to define main idea
Algorithms & Procedures	Series of ordered steps taken to solve a problem or achieve some end.
Automation	Having computers or machines do repetitive or tedious tasks
Simulation	Representation or model of process. Simulation also involves sunning experiments using models.
Parallelization	Organize resources to simultaneously carry out tasks to reach a common goal

본 연구에서는 이러한 컴퓨터 사고력의 주요개념을 통해 실생활의 문제 상황을 분석하여 학습자들이 컴퓨터를 활용하여 문제해결을 할 수 있는 능력 함양에 초점을 두어 대학에서의 SW교육을 살펴보았다.

3. 국내외 컴퓨팅 사고력 교육 사례

3.1 국내대학 컴퓨팅 사고력 교육

그동안 국내에서 시행되었던 대학 정보화 교육에 대한 최근 연구를 살펴보면 기기를 중심으로 한 대학생들의 스마트기기의 교육적 활용이나 기술수용모형과 관련된 연구들이 있었다[20,21] 정보화 교육의 내용적인 측면

에서는 대부분 사무용 응용프로그램에 관한 디지털리터러시 내용이었을 뿐만 아니라, 교육내용이 10여 년 전에 설계된 내용들이라는 문제점을 지적하면서 대학 SW교육 미래방향을 위해서는 컴퓨팅 사고력과 프로그래밍 교육이 포함되어야 한다는 의견이 제시되기 시작했다[21].

또한 대학 내 학생들의 SW 교육에 대한 관심과 수요가 점차 증가하고 있으며, 구체적인 예로 서울대 코딩교과목에 비전공자의 비율이 2012년에 9%였는데, 2015년에는 55%로 급증했다[22]. 그러므로 현재 비전공자도 컴퓨팅 사고력에 기반을 두어 해당분야에서 부가가치를 높이고 융합SW인재로 성장할 수 있도록 기회를 제공하는 노력이 필요하다.

현재 국민대는 국내 최초로 전체 학부생을 대상으로 SW교육을 의무화하였고, 컴퓨터개론, 프로그래밍 언어 등 1년간 6학점 이상 SW과목을 수강하도록 하였다. 연세대에서는 컴퓨팅 사고력 과목을 2015학년도에 신설하여 시범적으로 운영하고, 2016학년도 신입생 전체에게 필수교과목으로 실시할 예정이다.

또한 동국대학교에서도 2016학년도 신입생 중 인문·사회·예체능·자연계열 등 비전공자 전원을 대상으로 SW 강좌 ‘소프트웨어와 미래사회’ 교과 수업을 의무화를 통한 SW융합형 교육을 목표로 하고 있다.

그 외에도 SW중점대학으로 선정된 가천대학교와 고려대학교 등 에서도 2016학년도 신입생 전체 대상으로 필수 교양과목으로 SW관련 교육을 실시할 예정이다 [8,23,24].

3.1.1 연세대학교 컴퓨팅 사고력 교육

2015학년도 2학기에 시범적으로 운영되고 있는 연세대학교의 컴퓨팅 사고력 교과목 개발과 운영 및 개편내용을 살펴보면 다음과 같다[25].

우선 수강규모는 30명으로 이과계열을 제외한 1학년을 대상으로 하고 있다. 전공별 구성인원은 인문9명, 교육 1명, 상경 4명, 경영 4명, 사회과학 4명, 생활과학 3명, 공학 2명, 글로벌인재 1명, 자유전공 2명으로 나타났다. 수업목표는 컴퓨터를 배우지 않는 계열의 학생들에게 컴퓨팅의 기본개념을 소개하여 논리적으로 추리하고 문제를 해결하는 방법을 익히고 스스로 문제를 정의하고 해결하는 실습과정을 병행하였다. 특히 스스로 문제를 정의하고 해결하는 프로젝트 과정은 Solving Problem을 다

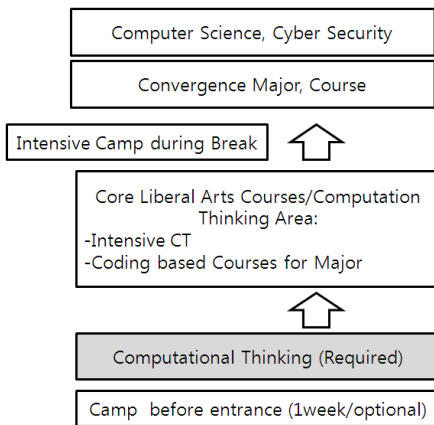
루는 주차에 시작해서 학기말까지 이어진다. 2015학년도 2학기 현재 운영 중인 교과 과정은 다음 <Table 2>와 같다[26].

<Table 2> Computational Thinking Course Syllabus of Yonsei University

Week	Topics
1	What is Computational Thinking?
2	Computational Thinking in Daily Life
3	The Story of the Computer (I)
4	Assignment (I): Independent Study
5	The Story of the Computer (II)
6	Solving Problems
7	Midterm Exam
8	Algorithmic Thinking
9	Modeling Solutions
10	Concurrent Activity
11	Inventive Problem Solving
12	Creative Problem Solving
13	Term Project: Problem Solving
14	Presentation and Poster Session
15	Assignment (II): Independent Study

3.1.2 고려대학교 컴퓨팅 사고력 교육과정

고려대학교의 경우에는 2015년도에 SW중점대학으로 선정되어 2016학년도 신입생을 대상으로 다음과 같은 체계로 SW교육을 실시할 예정이다. 입학 전 신입생 중 1000명 정도가 선택해서 수강할 수 있는 1주일 기간의 ‘입학 전 캠프’를 제공하고, 입학 후에는 신입생 전체 필수 교양과목으로 ‘정보적 사고(컴퓨팅 사고력)’ 1학점을 실시할 예정이다. 필수 컴퓨팅 사고력 과목을 이수한 후에는 심화과정을 희망하는 학생들을 위해서 핵심교양에 컴퓨팅 사고력 영역 수업군을 만들어서 심화 컴퓨팅 사고력 과정과 전공별 선호되는 코딩기반 과정 등을 두어 각 과정별로 다양한 수업들을 제공할 예정이다. 그 이후에는 3학년에서는 소규모그룹의 SW비전공자들을 대상으로 연계/융합전공 과정이나 코스를 통해 2년 동안 심도 있는 교과과정이 운영될 것이며, 추후 컴퓨터과학이나 사이버보안과 같은 분야 전공으로 연계될 수 있다. 이와 관련된 교육과정 체계도를 보면 다음의 [Fig. 1]과 같다[24].



[Fig. 1] Computational Thinking Curriculum Framework of Korea University

3.2 해외대학 컴퓨팅 사고력 교육

3.2.1 하버드대학교 CS50

하버드대학교의 CS50코스는 알고리즘적으로 생각하는 방법과 문제를 해결하는 법을 익히는 것에 목적이 있다. 수업에서 다루는 주제도 추상화, 알고리즘, 데이터 구조, 자원 운영, 정보보안, 소프트웨어 엔지니어링, 웹개발 등을 다루고 있으며, 프로그램 언어의 경우에는 C, PHP, JavaScript plus SQL, CSS, HTML을 포함하고 있다. 주 차별 수업내용은 다음의 <Table 3>과 같다[27].

<Table 3> Course Syllabus of CS50

week	contents
1	Binary. ASCII. Algorithms. Pseudocode. Source code. Compiler. Object code. Scratch. Statements. Boolean expressions. Conditions. Loops. Variables. Functions. Arrays. Threads. Events.
2	Linux. C. Compiling. Libraries. Types. Standard output.
3	Command-line arguments. Searching. Sorting. Bubble sort. Selection sort. Insertion sort. O. Ω. Θ. Recursion. Merge Sort.
4	Stack. Debugging. File I/O. Hexadecimal. Strings. Pointers. Dynamic memory allocation.
5	Heap. Buffer overflow. Linked lists. Hash tables. Tries. Trees. Stacks. Queues.
6	TCP/IP. HTTP. Quiz 0.
7	HTML. CSS. PHP.
8	MVC. SQL.
9	JavaScript. Ajax.
10	Security. Artificial intelligence.
11	Artificial intelligence, continued.
12	Exciting conclusion

CS50에서는 프로그래밍 이외에도 수업에서 제시되는 문제세트가 있는데, 여기에는 생물, 암호해독, 금융, 게임 등 실제 분야내용이 반영되어 있어 학생들이 프로그래밍 과 실제 세계와의 연계성을 경험할 수 있다.

수업에서 비중이 큰 최종프로젝트는 학생들이 관심이 있는 캠퍼스나 세상에 영향을 줄 수 있는 실제 문제를 선정하고 이에 대한 해결안을 코스에 대해서 배운 어떤 프로그램 언어나 그 이외의 것을 이용해서 2~3명의 그룹을 이루어서 프로그래밍을 통해 소프트웨어를 개발하는 것이다. 만약에 다른 영역에 있는 수업의 프로젝트와 결합해서 양쪽 수업과제의 목적을 모두 맞춘다면 하나의 프로젝트로 양쪽 수업에 제출해도 된다. 단, 사진에 양쪽 수업에 모두 알리고 이에 관해서 허락을 받아야한다. 원활한 진행을 위해 사전 연구계획서, 최종 연구계획서, 과정 진행보고서 활동이 단계적으로 이루어지고 있으며, 심화 활동을 위한 CS50 해커톤과 함께 다른 학생들의 프로젝트 결과물을 볼 수 있는 CS50 전시회도 개최된다.

3.2.2 버클리대학교 Beauty and Joy of Computing

<Table 4> Course Syllabus of BJC

unit	theme
1	Welcome to Snap!
2	Looping and Custom Blocks
3	Putting it Together
4	Abstraction and Testing
5	Lists
6	Tic Tac Toe
7	Algorithms
8	Algorithm Complexity
9	Concurrency
10	Trees and Fractals using Recursion
11	Using Data
12	Recursive Reporters
13	The Internet
14	Practice with HOFs and Functions as Data
15	HOFs with Recursion
16	Simulation in Science
17	Simulation: BioInformatics
18	Applications that changed the world
19	Distributed Computing
20	Robots

버클리대학교의 Beauty and Joy of Computing (이하 BJC)은 대학교 1학년 CS비전공자들을 위한 CS입문코스 로 SNAP!이라는 스크래치 기반의 프로그램을 사용한다.

단순히 프로그래밍을 배우는 것을 넘어서 추상화, 설계, 시뮬레이션 등과 같은 컴퓨팅에서의 ‘큰 아이디어(big idea)’에 중점을 두며, 어플리케이션들이 만드는 세계 변화, 컴퓨팅의 역사, 현재와 미래에 관한 대화를 한다. 무엇보다도 코스 전반에 걸쳐 학생과 사회와의 관련성을 강조한다. 그래서 최종 프로젝트는 학생 스스로가 자신에게 가장 관심 있는 주제를 직접 선택해서 진행을 하게 하여 컴퓨팅의 아름다움과 즐거움에 학생들이 경험할 수 있게 하는 것에 그 목적이 있다. 2015-16 수업커리큘럼은 현재 업데이트 중에 있으며 그 이전의 수업내용은 <Table 4>와 같이 구성되어 있다[28].

4. 컴퓨팅 사고력 향상을 위한 교수-학습 모형

4.1 컴퓨팅 사고력 실행 교육과정

국내의 대학에서 진행 중인 컴퓨팅 사고력과 관련된 몇 가지 사례를 살펴보면, 수업의 규모나 학생들의 전공에 따라 다양성이 있을 수 있으며, 각 대학의 교육과정에 따라 전반적인 틀이 다르게 있을 수 있다는 것을 볼 수 있다. 무엇보다도 컴퓨팅 사고력과 관련되어서 중요한 점은 “컴퓨팅 사고력은 컴퓨터 전문가들처럼 생각하는 것”이 아닌 “컴퓨터 과학의 기본 개념과 원리에 기반한 문제해결 접근법”이라는 점이다[10,29]

최근 미국대학협회의와 하트연구소가 공동으로 기업인 및 고용주들을 대상으로 실시한 설문조사에 따르면, 기업인들과 고용주들의 95%가 기업현장의 혁신에 기여할 수 있는 지식과 스킬을 중요시하였으며, 대학의 전공교육보다는 비판적 사고능력, 명확한 커뮤니케이션 능력, 복잡한 문제해결능력을 손꼽았다. 그 외에도 80%의 기업인들과 고용주들이 전공에 상관없이 폭넓은 교양지식 획득과 함께 공동문제해결, 연구, 인턴십, 프로젝트, 커뮤니티 참여활동 등의 실무활동을 중요시 하며, 마지막으로 자신이 가지고 있는 지식과 기술을 보여줄 수 있는 e-포트폴리오의 중요성을 강조하고 있다[30].

위와 같은 능력을 함양하기 위해 Dede, Mishra, Voogt(2013)는 컴퓨터 사고력을 함양하기 위한 세부 교육과정 틀을 다음과 같이 제시하고 있다[31]. 첫째, 컴퓨팅 연결성(connecting computing)으로 컴퓨팅을 통한 혁신이

컴퓨팅이 개인·사회·상업시장 등에 미치는 영향력을 확인하도록 한다. 둘째, 컴퓨팅 창작물의 개발(developing computational artifacts)은 컴퓨팅을 통해 다양한 형태의 창작물을 만들어 내는 것으로 음악 리믹싱, 애니메이션 생성부터 웹사이트 구축 프로그래밍까지 여러 활동이 포함되어 있다. 이를 통해 문제해결을 창의적으로 하기 위한 컴퓨팅 기법을 적용하고 직접 컴퓨팅 작업에 참여한다. 셋째, 추상화(abstraction) 활동은 여러 수준에서의 추상화 적용과 이해에 바탕을 두어 모델개발과 자연 및 인공 현상에 대한 시뮬레이션을 위한 추상화를 이용할 수 있다. 넷째, 문제와 창작물에 대한 분석(analyzing problems and artifacts)에서는 컴퓨팅의 결과물들이 심미적, 수학적, 실용적 등의 다양한 기준들에 의해 평가되고 분석될 수 있는 것으로 컴퓨팅 작업을 통해 문제에 대한 해결안과 모델, 인공물 등을 만들고 자신의 작업을 뿐만 아닌 타인의 컴퓨팅작업도 평가하고 분석한다. 다섯째, 컴퓨팅 결과물의 공유와 소통(communicating)은 컴퓨팅에 관한 선택의 적합성과 디자인에 대한 정당화를 하는 활동으로서 상황과 맥락 안에서 결과물에 관한 의미를 설명하고 정확하고 정밀한 컴퓨터 언어, 기호, 시각화 기법 등으로 컴퓨팅 작업을 기술하고 궁극적으로 컴퓨팅 인공창작물의 목적을 요약 및 설명한다. 마지막으로 협업(collaboration)은 혼자 독립적으로 일할 때보다는 함께 일할 때 혁신을 더 많이 성취할 수 있다는 것을 강조한다. 다른 학생들과 공동작업 유형은 컴퓨팅 문제해결, 프로그램 개발, 대규모 작업에서의 공동작업 등으로 나타날 수 있다. 앞의 컴퓨팅 사고력 교육과정은 2016년도 미국 AP(advanced placement) 컴퓨터 과학교과에서의 컴퓨팅 사고력을 위한 6가지 실천사항으로 이미 교육과정에 반영되어 운영 중에 있다[32].

4.2 컴퓨팅 사고력 향상을 위한 교수학습모형

최형신(2014)은 컴퓨팅 사고력 세부역량을 정의하고 스크래치를 활용한 수업내용을 제시하였으나 구체적인 교수학습모형을 언급하지는 않았다[11]. 반면, 박정호(2015)의 연구에서는 스토리텔링기반의 SW교육을 실시하여 SW개념, 구현능력, 교육태도가 향상되었음을 실증적으로 규명하였고, 김수환, 한선관(2012)의 연구에서는 디자인기반학습을 적용한 컴퓨팅 사고력 효과성 연구를 통해 디자인 기반학습이 컴퓨팅 사고력과 함께 자기 프

로그래밍 능력 향상에 긍정적 영향을 미친 것으로 나타났다[33,34]. 그 외에도 전연주와 김태영(2015)의 최근 연구에서는 창의성향함에 초점을 두어 창의적 문제해결 모형과 접목을 한 CT-CPS 모형을 제시하였다[35]. 이와 같이 컴퓨팅 사고력 향상을 위한 교육은 지식위주의 교육이 아니라 다양한 실습과 활동 위주의 교육이므로 학습자의 자발적이고 주도적인 참여가 요구되는 학습자중심의 교수학습전략이 필요하다. 앞선 국내외 사례에서 컴퓨팅 사고력 교육과정이나 수업주제들은 볼 수 있었지만, 구체적인 교수학습모형을 제시한 내용은 찾아볼 수 없었다.

컴퓨팅 사고력 향상을 위해 다양한 교수학습전략이 가능하지만, 그나마 앞서 제시된 컴퓨팅 사고력에 관한 교수학습모형 선행연구 경우에는 K-12 교육과정 안에서 특정 단원을 위한 연구들이었으며, 대학에서의 SW교육에 관한 구체적인 교수학습모형 내용은 거의 전무한 상태이다.

이에 본 연구에서는 앱개발 또는 스크래치 프로그램을 활용한 컴퓨팅 사고력 프로젝트학습 모형(CT-PBL)을 <Table 5>와 같이 제안하고자 하며 각 단계에 대한 구체적인 설명은 다음과 같다[36].

1단계에서는 해결될 필요가 있는 문제로 시작한다. 실제로 학습자들이 당면하고 있는 실제적인 문제들을 탐색하고 이를 규정한다. 2단계에서는 새로운 문제들에 대한 가능한 해결안들을 테스트하거나 현재 모델에서 결함이 있는지를 분석해본다. 3단계에서는 준거나 파라미터(parameters) 설계를 구체화할 수 있는 중요한 데이터를 수집하고 해당 설계를 테스트 해본다. 이를 위해 데이터 분석을 위한 변인을 규명하고 데이터 수집을 한다. 4단계에서는 수집된 데이터를 분석한다. 이를 통해 다른 해결안들끼리 비교하며, 준거에 어떻게 부합하는지 결정할 수 있다. 5단계에서는 디자인에 대한 시뮬레이션을 통해 향상시킬 수 있는 부분에 대해서 효과적인 테스트를 할 수 있는 기회를 제공한다. 6단계에서는 문제해결을 위해 과학적 지식을 기반으로 한 체계적인 과정을 설계한다. 요구되는 기능, 기술적 가능성, 비용, 안전성, 심미성, 합법성 등을 고려하여 다양한 해결안을 도출하되, 평가과정을 통해 최적의 선택을 한다. 7단계에서는 다양한 아이디어 경쟁에서 가장 유망한 해결안을 선택하는 중요한 매 단계에 걸쳐서 동료들과 협업을 한다. 대안들과 비교

하고 테스트한 데이터를 기반으로 증거들을 산출하고, 결론을 방어할 수 있는 증거들로부터 논쟁하는 등의 과정을 통해 직접 최선의 해결안을 도출한다. 8단계에서는 아이디어를 명확하게 구두 또는 글로 동료들에게 발표할 수 있어야 한다. 새로운 기술을 통해 협업과 소통을 확장할 수 있다.

1단계에서 4단계에 걸쳐서 교수자의 역할은 교과과정을 통합하는 역할을 한다. 그리고 5단계에서부터 7단계까지는 프로젝트 행동계획과 관리의 역할을 하고, 마지막 단계에서는 전체 프로젝트 공개행사(showcase)를 진행한다. 전체 과정에서 사용하는 앱개발 또는 스크래치 프로그램 사용에 관한 정보는 온라인 강의를 통해서 제공하여 기본적인 프로그래밍 지식과 기술을 습득하도록 한다.

<Table 5> Steps of CT-PBL

step	Description & activities
1	Asking questions and defining problems: Observation, relevant data search
2	Developing and using models: writing an inquiry report about detected problem with given form
3	Planning and carrying out investigations: Interview, web searching, survey, field investigation
4	Analyzing and interpreting data: Graph, table, chart, text
5	Using mathematics and computational thinking: Flow chart, use paper, presentation, block programming
6	Constructing explanations and designing solutions: Designing of the desired program using App inventor or Scratch within the program, writing a draft of block programming for the desired program
7	Engaging in argument from evidence: Debugging of the block programming of the program, completion of the program
8	Obtaining, evaluating, and communicating information: Presentation, demonstration of the completed program, feedback exchange about other's output(showcase)

CT-PBL 모형은 대학교육에서 학습자를 위한 스캐폴딩 구조와 함께 프로젝트 실행 매 단계에서 어떤 실천을 할 수 있는지를 제시하고 있다. 컴퓨터 사고력에 관한 수업의 주제와 학습대상자의 수준 및 프로그래밍 환경에 다양함에 따라 CT-PBL도 다양하게 적용될 수 있다. 이에 대한 자세한 프로그램은 미국 정보과학원리 홈페이지에서 스크래치에서부터 NetLogo에 이르기까지 다양하게 제시하고 있다(<http://apcsprinciples.org/>).

5. 결론 및 제언

사회 전반적인 영역에서 소프트웨어에 대한 역할비중이 커지면서 이에 대한 능력을 갖추고 자신의 아이디어와 가치를 소프트웨어를 통해서 구현하고 문제를 해결할 수 있는 능력을 지닌 인재양성에 미래의 국가경쟁력을 결정짓는 시기가 도래하고 있다. 이는 단순히 프로그래밍 능력 이상의 비판적 사고와 문제해결능력을 동시에 갖춘 것으로 현재 대학사회에서 양성하고자 하는 인재상과 공통적인 부분이 있다[37,38]. 본 연구를 통해 세계적인 SW교육의 흐름을 살펴보면 SW교육이 단순히 프로그래밍 교육이 아닌 문제해결능력까지 갖춘 수 있는 인재양성을 위해 컴퓨팅 사고력이 그 핵심을 이루고 있다는 점을 볼 수 있었다.

많은 국가들에서 초중등교육에서 컴퓨팅 사고력 교육 과정이 포함되어 시작된 반면 아직 대학SW교육에서의 컴퓨팅 사고력 교육에 대한 노력은 시작하는 단계라는 점은 국내외에서 공통적으로 나타나고 있다.

본 연구에서는 컴퓨팅 사고력 함양을 위한 대학교육에서의 SW교육이 나갈 방향을 위한 기초연구로써 전반적인 국내외대학에서의 트렌드 및 사례조사와 함께 이를 위한 교수학습모형을 제시했다는 것에 시사점이 있다고 볼 수 있다.

본 연구의 제한점으로는 대학SW교육이 태동하는 시기여서 관련자를 대상으로 한 요구조사나 실제 CT-PBL 교수학습모형을 대학교육에 적용한 효과성 연구와 같이 실증적 데이터를 기반으로 하지 않은 점이다. 그러므로 실효성 있는 대학 SW교육을 위해서 다음과 같은 제언을 하고자 한다.

첫째, 실증적 요구조사를 바탕으로 한 대학내 구성원들간의 융합이 반영되어야 한다. SW교육이라고 해서 SW전공자만 참여하는 것이 아니라 관련 교육전문가, SW비전공자들이 서로 협업해서 교육과정체계를 만드는 노력이 있을 때, 진정한 컴퓨팅 사고력을 향상시킬 수 있는 교육과정이 도출될 수 있다.

둘째, 전공별 맞춤형 컴퓨팅 사고력 교육과정, 학습자 중심의 다양한 교수방법이 적용된 코스가 설계되면 이를 현장에 적용하여 컴퓨팅 사고력 향상에 효과적인지를 실증적 데이터를 기반으로 검증할 필요가 있다.

셋째, 컴퓨팅 사고력 수업이외에도 이를 지원할 수 있

는 연계적인 교육과정도 마련되어야 한다. 필수 컴퓨팅 사고력 수업 외에도 더 높은 수준의 SW교육을 위해 다양한 수준의 관련 코스를 개설할 뿐만 아니라 창의적 캡스톤, 해커톤, 전시회, 융복합전공 개설, 학교 외부 커뮤니티 연계 프로그램, 로봇을 활용한 r-learning 프로그램 등 학생들이 다양한 선택을 할 수 기회를 제공해야한다 [39,40].

넷째, 이를 전공별로 효과적으로 지원해 줄 수 있도록 각 전공 분야별 강사 풀과 실습 지원 조교와 같은 인력시스템이 갖추어져야 하며, 적정수준 규모의 학생과 지원 인력 매칭이 이루어져야 한다. 동시에 대형 강의, 온라인 강의, 실습 등과 같은 융복합적인 방식의 교육이 동시에 이루어지므로 이를 지원해줄 수 있는 풍부한 교육자원이 갖추어진 교육환경 시스템도 같이 요구된다.

마지막으로는 행정적으로 기존 강의 및 학사운영 시스템으로는 이상적인 형태의 컴퓨팅 사고력 수업 운영과 불일치하는 부분들이 많이 발생한다. 그러므로 온오프라인 출석처리, 과제 제출과 자동 채점 등 수업 및 학습 과정이 효율적으로 이루어지도록 하는 전산 시스템과 함께 전공 및 수업간 연계를 지원할 수 있는 탄력적인 학사운영시스템이 필요하다.

앞으로 대학에서의 SW교육도 초중등교육과 마찬가지로 필수교육으로 확산될 것이다. 각 대학들이 만든 SW교육과정 운영에 대한 성과나 개선점 관계 공유하는 노력이 있다면 효과적으로 이를 안착시키고 확산시킬 수 있을 것으로 기대된다.

REFERENCES

- [1] J. S. Sung, H. Kim, "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.
- [2] J. Kim, W. Lee, "Controversial issues in knowledge and problem solving skills of information subjects observed after amending the curriculum in the U.K.", The Journal of Korean Association of Computer Education, Vol, 17, No. 3, pp.53-63, 2014.
- [3] Nacce & CAS, "Computing in the national curriculum: A guide for secondary teachers",

- http://www.computingschool.org.uk/data/uploads/cas_secondary.pdf(October 2, 2015)
- [4] CSTA & ISTE, "Operational definition of computational thinking for K-12 education", <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>(November 30, 2015)
- [5] A. Dubrow, "Increasing accessibility to computer science education across the U.S. National Science Foundation", http://www.nsf.gov/news/news_summ.jsp?cntn_id=129882(October 3, 2015)
- [6] J. Johnson, "France to offer programming in elementary school". <http://www.itworld.com/article/2696639/application-management/france-to-offer-programming-in-elementary-school.html>(November 2, 2015)
- [7] S. Shin, W. Bae, "Review of software education based on the coding in Finland", *The Journal of The Korean Association of Information Education*, Vol. 19, No. 1. pp.127-138, 2015.
- [8] MOE & MSIP, "Human Resource Development Plan for software-intensive society", <http://www.msip.go.kr/web/msipContents/contentsView.do?catelId=mssw311&artId=1270998>(October 2, 2015)
- [9] H. Jung, "An Empirical study on information liberal education in university based on IT fluency and computational thinking concept", *Journal of the Korea Society of Computer and Information*, Vol. 19, No. 2, pp.263-274, 2014.
- [10] J. M. Wing, "Computational thinking. Computations of the ACM", Vol. 49, No. 3, pp.33-35, 2006.
- [11] H. Choi, "Developing lessons and rubrics to promote computational thinking". *Journal of The Korean Association of Information Education*, Vol. 18. No. 1, pp.57-64, 2014.
- [12] S. Kim, "Effects of teaching and learning strategies of learner-centered learning for improving computational thinking", *Journal of The Korean Association of Information Education*, Vol. 19. No.3, pp.323-332, 2015.
- [13] M. Guzdial, "Paving the way for computational thinking, *Communications of the ACM*", Vol. 51, No. 8. pp.25-27, 2009.
- [14] A. Perils, "The computer in the university", In M. Greenberger, Ed. *Computers and the World of the Future*(pp.180-219). Cambridge, MA:MIT press. 1962.
- [15] S. Papert, "Mindstorms: Children, computers, and powerful ideas", NewYork, NY:Basic Books, 1980.
- [16] A. V. Aho. "Computation and computational thinking", *Computer Journal*, Vol. 55, pp.832-835, 2012.
- [17] B. C. Czerkawski, E. W. LymanIII, "Exploring issues about computational thinking in higher education", *Tech Trends*, Vol. 59, No. 2, pp.57-65. 2015.
- [18] CSTA & ISTE, "Computational thjinking in K-12 education teacher resource second edition", http://www.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvsn=2(November 11, 2015)
- [19] J. H. Joo, "Understanding korean college students' social commerce behavior through an Integrated model of technology readiness, technology acceptance model, and theory of planned behavior", *Journal of Digital Convergence*, Vol. 13, No. 7, pp.99-107, 2015.
- [20] S. Pi. "Educational utilization of smart device in the convergence education era", *Journal of Digital Convergence*, Vol. 13, No. 6, pp.29-37, 2015.
- [21] D. Kim, E. Ha, "The future direction of information education in university according to computerization", *Journal of Digital Convergence*, Vol. 13. No.10, pp.33-40, 2015.
- [22] Hankyung, "Resurrection of Department of Computer Science", <http://www.hankyung.com/news/app/newsview.php?aid=2015052630831>(September 16, 2015)
- [23] Gachon University, "Proposal of SW intensive University 2015", Gachon University, September 1, 2015
- [24] H. Kim, "Current trends of computational thinking education in university", *KERIS Symposium in ICT education*, pp.237-250, 2015.
- [25] J. Na, "Computational thinking education, curriculum development, its status", *Computational Thinking Forum*, pp.45-51, 2015.
- [26] Yonsei University, "Computational thinking course syllabus", <http://www4.yonsei.ac.kr/fresh/ct/>(October 10, 2015)

[27] Harvard University, "CS50 syllabus", <http://cdn.cs50.net/2015/fall/lectures/0/w/syllabus/cs50/cs50.html>(November 10, 2015)

[28] UC Berkeley University, "Beauty and joy of computing", http://bjc.berkeley.edu/bjc-r/course/berkeley_bjc.html,(October 20, 2015)

[29] S. Choi. "An analysis of 'Infomatics' curriculum from the perspective of 21st century skills and computational thinking", The Journal of Korean Association of Computer Education. Vol. 14. No. 6, pp.19-30, 2011.

[30] Hart Research Associates, "Trends and emerging practices in general education-based on a survey among members of the association of american colleges and universities", Research Report. Washington DC: AAC & U. 2009.

[31] C. Dede, P. Mishra, J. Voogt, "Working group 6: Advancing computational thinking in 21st century learning", In: EDUsummit 2013, International summit on ICT in education, Washington, 1-2 October, 2013.

[32] College Board. "AP computer science principles including the curriculum framework", College Board, 2016

[33] J. Park, "Effects of storytelling based software education on computational thinking", Vol. 19, No. 1, pp.57-68, 2015.

[34] S. Kim, S. Han, "Design-based learning for computational thinking", Vol. 16, No. 3, pp.319-326, 2012.

[35] Y. Jeon, T. Kim, "The development of the CT-CPS framework for creative and integrative software education", Korean Journal of Teacher Education, Vol. 31, No. 3, pp.67-88, 2015.

[36] E. Baker, B. Trygg, P. Otto, M. Tudor, "Project-based learning model, relevant learning for the 21st Century", Pacific Education Institute, 2011.

[37] S. H. Park, "The effectiveness of learning community for the development of convergence of university students", Journal of Digital Convergence, Vol. 13, No.9, pp.29-37, 2015.

[38] S. Yu, J. Kang, "Correlation between critical thinking ability and disposition of university nursing

students for the development of convergence", Journal of Digital Convergence, Vol, 13, No. 6, pp.197-203, 2015.

[39] S. Yoon, E. Jang. "A development of creative capstone design Education", Journal of the Korea Convergence Society, Vol, 5, No. 4, pp.15-20, 2014.

[40] J. Lee, "r-learning and educational information policies", Journal of the Korea Convergence Society, Vol, 1, No. 1, pp.1-15, 2010.

박 성 희(Park, Sung Hee)



- 1996년 2월 : 이화여자대학교 교육 공학과(문학사)
- 1998년 8월 : 이화여자대학교 교육 공학과(문학석사)
- 2006년 8월 : 미국 퍼듀대학교 교육 공학(철학박사)
- 2010년 3월 ~ 현재 : 가천대학교 교육대학원 조교수

- 관심분야 : 테크놀로지 활용 교수-학습, 고등교육
- E-Mail : sungheepark@gachon.ac.kr