

학교 급별 연계성 있는 소프트웨어 교육 체제 설계를 위한 연구

신승기* · 배영권**

조지아대학교 학습설계공학전공* · 대구교육대학교 컴퓨터교육과**

요 약

본 연구에서는 소프트웨어 교육의 전면 시행을 앞두고 학교 급별 연계성 있는 교육과정 운영을 위한 가이드 라인을 제시하고자 하였다. 이를 위하여 학생들의 수준에 맞는 교육용 프로그래밍 언어의 형태를 제시하기 위해 해외의 사례를 살펴보았다. 해외의 사례는 현재 필수교육과정으로써 프로그래밍 과정을 운영하고 있는 나라의 교육과정을 분석하여 초-중-고등학교에 제시된 프로그래밍 언어의 유형을 살펴보고, 이를 통해 우리나라의 소프트웨어 교육 체제 구성을 위한 제언을 하고자 하였다. 해외의 사례를 살펴본 결과, 초등학교에서는 블록기반의 프로그래밍 언어를 활용하고 있고, 중학교부터는 텍스트 기반의 프로그래밍 언어를 중점적으로 사용하고 있는 것으로 나타났다.

키워드 : 소프트웨어 교육, 코딩 교육, 교육체제, 교육과정, 해외사례

A Study on the Hierarchical Instructional System Design of Software Education by School System

Seungki Shin* · Youngkwon Bae**

Learning, Design, and Technology, The University of Georgia, USA* ·
Dept. of Computer Education, Daegu National University of Education**

ABSTRACT

In this study, the direction for hierarchical curriculum organization about software education in Korea was suggested in terms of overall execution of software education. The international case studies especially was conducted in order to suggest the propel educational programming language for level of students in the programming activity. In terms of the international case studies, the type of programming language was examined, which is suggested to each school level as a part of required regular curriculum. Then, the direction was supposed to suggest the instructional system organization of software education for Korea through the result of case studies. The results of case studies indicated that elementary school use the block based programming language, and text based programming languages are used from middle school.

Keywords : Software Education, Coding Education, Instructional System, Curriculum, Case Studies

교신저자 : 배영권(대구교육대학교 컴퓨터교육과)

논문투고 : 2015-12-03

논문심사 : 2015-12-04

심사완료 : 2015-12-19

1. 서론

2015개정교육과정이 발표되면서 나타난 가장 큰 변화 중의 하나는 소프트웨어 교육이 필수정규교육과정으로 반영된다는 것이다[3]. 소프트웨어교육의 내용이 중학교부터는 기존의 선택교과였던 ‘정보’교과가 필수교과로써 체제가 변환되어 소프트웨어교육의 내용을 반영하여 2018년부터 실시되며, 초등학교에서는 2019년부터 ‘실과’교과에 내용이 반영됨으로써 필수정규교육과정으로 반영되어 실시된다[3][12]. 2015개정교육과정에 나타난 정보교과와 내용체계는 ‘정보문화’, ‘자료와 정보’, ‘문제해결과 프로그래밍’, ‘컴퓨팅 시스템’으로 제시되어 있으며, 2009개정교육과정과 비교하여 새롭게 나타난 부분은 컴퓨팅 사고력(Computational Thinking)을 기반으로 문제해결력을 신장하는 데 초점을 둔다는 것이다[12][13]. 2015개정교육과정의 정보교과에서 제시된 컴퓨팅 사고력과 관련된 학습목표를 살펴보면 ‘컴퓨팅 원리에 따라 문제를 추상화하여 해법을 설계하고 프로그래밍 과정을 통해 소프트웨어로 구현하여 자동화할 수 있는 능력을 기른다’라고 제시되어 있다[12]. 즉, 기존의 교육과정에서도 프로그래밍 과정이 제시되어 있었으나, 2015개정교육과정에서 필수정규교육과정으로 반영되면서 컴퓨팅 사고력(Computational Thinking)을 사고전략으로 반영하여 소프트웨어교육에 대한 내실 있는 교육과정을 운영하고자 하는 것이다[13]. 이는 프로그래밍과정을 통한 소프트웨어교육역량을 기르는 것을 2015개정교육과정의 핵심내용 중 하나로 설정하였음을 의미한다. 교육과정에서는 프로그래밍 과정에서 컴퓨팅 사고력을 활용해야 하며, 컴퓨팅사고력의 핵심 내용은 추상화(Abstraction)와 자동화(Automation)라고 제시하고 있다[12].

그러나, 2015년에 고시된 소프트웨어교육운영지침[13]과 2015개정교육과정에 제시된 정보교육과정[12]에서는 프로그래밍 교육을 위한 프로그래밍 언어 선정기준에 대해 세부적인 지침이 다소 마련되어 있지 않은 편이다. 2015개정교육과정의 정보교과해설서 각론의 ‘문제해결과 프로그래밍’ 단원을 살펴보면, ‘학습자 수준에 적절한 교육용 프로그래밍 언어를 선택하도록 명시하고 있으며, 특정 프로그래밍 언어의 기능 습득에

치중하지 않도록 유의하라고 제시하고 있고, 문제해결을 위한 프로그램 설계 및 개발과정을 통해 컴퓨팅 사고력을 신장하는데 초점을 두어야 한다’라고 기술하고 있다[12].

반면, 해외의 소프트웨어 교육과정을 살펴보면 학교급별 학습자의 수준에 따른 적절한 프로그래밍 언어의 형태 및 종류들을 국가교육과정의 차원에서 구체적으로 제시하고 있다. 이는 지역단위교육청 및 학교현장에서 관련 교육과정을 운영하면서 혼란을 최소화하고, 프로그래밍 언어 선정에 필요한 시간과 비용을 최소화하여 실제적인 교수학습방법 자체에 대한 연구가 이루어질 수 있는 여건과 환경을 제공하는 것이다.

따라서 본 연구에서는 소프트웨어교육을 필수정규교육과정으로 운영하고 있는 외국의 사례를 분석하여 공통적으로 나타나는 프로그래밍 언어 선정 기준 및 사례를 살펴보고, 우리나라의 소프트웨어 교육에 대한 시사점을 분석하고 제언하고자 한다.

2. 연구 목적 및 연구 방법

본 연구의 목적은 소프트웨어교육과정에서 제시된 ‘문제해결과 프로그래밍’ 단원의 교수학습방법에 대한 제언을 통해 학습자의 수준에 따른 구체적인 프로그래밍언어 선정 기준을 제공하는 데 있다.

이를 위한 연구방법으로는 현재 소프트웨어교육을 필수정규교육과정으로 반영하고 있는 외국의 교육과정을 분석하여, 국가수준교육과정에서 제시되어 있는 각론의 내용을 살펴봄으로써 학습자의 수준에 따라 어떤 프로그래밍 언어를 선정하여 제시해야 하는지에 대한 지침을 사례조사 하도록 한다. 또한, 구체적인 프로그래밍 언어가 제시되어 있다면 학교급별로 활용하기 위한 프로그래밍 언어가 어떻게 나타나 있는지를 살펴봄으로써, 학년단위 또는 학교급별 학습자의 수준에 따라 제시할 수 있는 적절한 프로그래밍 언어의 사례를 분석한다. 또한 사례조사를 통해 나타난 공통점을 토대로 우리나라 소프트웨어교육과정에서 프로그래밍 언어 선정과 제시를 위한 기준과 방안을 제언하고자 한다.

3. 해외의 소프트웨어 교육과정 분석

3.1 교육과정 분석 기준

현재 소프트웨어 교육을 필수정규교육과정으로 반영하여 추진하고 있는 나라들이 어떤 나라들이 있는지를 살펴보고, 국가수준교육과정으로 소프트웨어교육과정이 마련되어 있는 나라들을 대상으로 교육과정을 분석하였다. 국가수준교육과정으로서 필수정규교육과정으로 소프트웨어교육을 운영하고 있는 국가들은 다음 <Table 1>과 같이 나타났다.

<Table 1> Countries that operate the software education as a national required regular curriculum

Continent	Countries
Europe	U. K, Finland, Estonia
Oceania	Australia
Asia	India, Korea, Israel

<Table 1>에 제시된 국가 이외에도 미국 등 여러 나라가 언급되고 있지만, 예를 들어 미국의 경우 소프트웨어교육이 활성화되어 있지만, 국가수준의 필수정규교육과정으로 제시되어있지 않은 부분이기 때문에 표와 본 연구의 연구대상에서 제외하였다.

3.2 영국의 소프트웨어 교육과정 분석

영국은 소프트웨어 교육을 국가수준의 필수정규교육과정으로 운영하기 위하여 2013년 9월 교육과정을 공시하고 2014년 9월부터 소프트웨어 교육이 전면 실시되었다[8][16].

2013년 공시된 영국의 소프트웨어 국가수준 교육과정에 의하면, 학제를 다음 <Table 2>와 같이 제시하여 설명하고 있다. <Table 2>에 제시된 학제는 우리나라의 초등학교에서 고등학교까지에 해당하는 과정이며, 표에는 나타나있지 않지만 Key Stage 5과정으로 12-13학년의 Secondary/College 단계도 영국의 교육과정에서는 포함되어 있다[2].

<Table 2> School system in U.K.

Stage	Age	Grade	School
Key Stage 1	5-7	1-2	Primary
Key Stage 2	7-11	3-6	
Key Stage 3	11-14	7-9	Secondary
Key Stage 4	14-16	10-11	

영국의 국가수준 교육과정에 제시된 소프트웨어 교육관련 교육과정 해설에서 우리나라의 중학교 단계에 해당하는 Key Stage 3의 학생들에게 프로그래밍 언어 성취 기준에 대해서 다음과 같이 설명하고 있다.

Pupils should be taught to use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems.

학생(중학생)들은 다양한 컴퓨팅 문제를 해결하기 위하여 반드시 2가지 이상의 프로그래밍 언어를 학습해야 하며, 그중 최소한 한 가지 언어는 텍스트 기반의 프로그래밍 언어를 학습해야 한다.

미국의 CSTA(Computer Science Teachers Association)와 같은 컴퓨터교육을 위한 비영리기구가 영국에도 있으며 CAS(Computing At School)라고 불린다. 컴퓨터교육관련 교사 및 교수들로 구성된 비영리기구로 국가수준교육과정에 대한 해설 및 보완 등의 역할을 수행하고 있다[1][5]. CAS에서 발간한 초등학교 소프트웨어교육 수업을 위한 해설서를 살펴보면 Key Stage 1과 Key Stage 2를 위한 프로그래밍 언어들과 활용 기준들이 제시되어 있으며, 세부 내용은 아래 <Table 3>와 같다[5].

<Table 3> Programming Tools in Curriculum of U.K.

Stage	Programming Tool
Key Stage 1	Screen-based Programming Tools such as Scratch
Key Stage 2	Physical systems based on Screen-based Programming Tool

<Table 3>에서 살펴볼 수 있는 바와 같이, 영국 초등학교과정에서는 Screen-based Programming Tools 즉, 비주얼 또는 블록 기반의 프로그래밍 언어를 활용

하여 지도할 것을 기준으로 제시하고 있다.

또한, 초등학교의 일반학급에서 우수 학생 및 영재학생들에 대한 프로그래밍 언어 선정 기준이 제시되어 있으며 세부 내용은 아래 <Table 4>와 같이 나타났다[5].

<Table 4> Programming Tools for Gifted and Talented in U.K.

Pupils	Programming Tools
Leading Practitioners	Introducing the text-based programming using Python
Gifted	Logo or Python by particularly advanced pupils

이와 더불어, 단계별로 활용할 수 있는 프로그래밍 언어와 관련 도구들에 대해서도 언급하고 있다[5]. 초등학교 단계에서는 대표적으로 스크래치 프로그래밍 언어 사용을 권장하고 있으며, 이에 대한 대안으로 Kodu와 Logo를 제시하고 있다. 또한 KS1(Key Stage 1)과 KS2(Key Stage 2)에서 활용할 수 있는 프로그래밍 도구들을 아래 <Table 5>와 같이 소개하고 있다. KS1에서는 프로그래밍 장난감으로서의 교구를 소개하고 있으며, KS2에서는 프로그래밍언어와 연동할 수 있는 외부 장치로써의 교구들을 소개하고 있다는 특징이 있다[5].

<Table 5> Physical Tools for Programming in U.K.

Stage	Physical Tools
Key Stage 1	As a programmable toys, Bee-Bots, Roamers, Probots and Big Traks As a physical components,
Key Stage 2	FlowGo with Windows PC, LEGO WeDo with Scratch 1.4, Arduino or Raspberry Pi

3.3 핀란드의 소프트웨어 교육과정 분석

신승기, 배영권(2015, March)은 우리나라 소프트웨어 교육의 시사점을 분석하기 위하여 핀란드의 소프트웨어 교육과정에 대해 분석하고, 초-중-고등학교가 연계된 소프트웨어 교육이 실시되어야 한다고 하였으며, 초등학교부터 필수교육과정으로 소프트웨어교육이 이루어져야 한다고 하였다[19].

또한, 핀란드의 소프트웨어 교육과정인 KOODI2016을 인용하여[10] 학습자수준에 따른 프로그래밍 언어

선정 기준에 대해서 아래의 <Table 6>과 같이 제시하였다[9]. 세부 내용을 살펴보면, 초등학교 저학년인 1-2학년에서는 프로그래밍 개념을 익히기 위해 놀이를 활용하고, 초등학교 3-6학년에서는 비주얼 교육용 프로그래밍 언어를 활용할 것을 제시하고 있다. 또한, 중학교 단계인 7-9학년에서는 실제 프로그래밍 언어 즉 텍스트 기반의 프로그래밍 언어를 활용할 것을 기준으로 삼고 있다.

<Table 6> Programming Tools in Curriculum of Finland

Grades	Programming Tools
1-2 Grade	Play for learning the strategies and solving the problems
3-6 Grade	Visual programming language as an educational programming language
7-9 Grade	Actual programming language

이를 위하여, 각 학년에 적합한 프로그래밍 언어가 KOODI2016에 제시되어 있으며, 세부 내용은 아래의 <Table 7>과 같이 나타났다[10][19].

<Table 7> Recommended Resources in Finland

Grades	Programming Resources
1-2 Grade	CS Unplugged
3-6 Grade	Scratch
7-9 Grade	Khan Academy
Mobile	Tynker, Hopscotch

3.4 에스토니아의 소프트웨어 교육과정 분석

신승기, 배영권(2015, September)은 에스토니아에서 운영되고 있는 국가수준 교육과정에서의 에스토니아 소프트웨어 교육과정을 분석하여, 초-중-고의 학년 및 학교급간의 특성을 고려한 연계성 있는 교육과정의 필요성과 세부 교육과정 운영 지침을 제시하였다[20]. 또한, 에스토니아의 소프트웨어 교육을 위한 ProgeTiger 프로젝트를 운영하고 있는 HITSA라고 불리는 정부의 연구기관에서는 학습자의 수준을 고려하여 학년별 프로그래밍 언어 선정 기준을 아래 <Table 8>과 같이 제시하고 있다[8][20].

<Table 8> Programming Tools in Curriculum of Estonia

Grade	Programming Tools
1-4	1) First steps in programming - learning how to use the mouse and keyboard
	2) Graphic programming language - Kodu, Logo, and Scratch
5-9	1) Lego Robots - From NXT-G to NXC
	2) Web Programming - Website and web application
10-12	1) Web Programming - Website and web application

<Table 8>에서 살펴볼 수 있는 바와 같이 초등학교 저학년과 중학년에 해당하는 1-4학년에서는 프로그래밍이 친숙해지기 위한 흥미 있는 활동과 함께, 그래픽 기반의 프로그래밍 언어를 기준으로 제시하고 있다 [8][20]. 또한 초등학교 고학년과 중학교에 해당하는 단계인 5-9학년에서는 로봇을 활용하여 그래픽 프로그래밍 언어에서 텍스트기반 프로그래밍 언어로 넘어가는 중간 단계의 과정을 제시하고, 이후에 웹 프로그래밍 언어로서 텍스트기반 프로그래밍 언어를 활용할 수 있도록 하였다. 고등학교에 해당하는 10-12학년에서는 웹 프로그래밍 언어를 기반으로 실제 문제해결과정을 통해 프로그래밍 활동을 할 수 있도록 제시하고 있다[8][20].

3.5 인도의 소프트웨어 교육과정 분석

인도의 소프트웨어교육 관련 교육과정은 CMC(Computer Masti Curriculum: 즐거운 컴퓨터 교육 과정)이라는 이름의 프로젝트로 추진되었다[18][21]. 인도의 CMC에서는 대단원으로 ‘컴퓨터 친숙해지기’, ‘컴퓨터 응용프로그램 활용하기’, ‘사고과정 기르기’, ‘컴퓨터 프로그래밍’, ‘정보윤리’ 등 다섯 가지 분류로 구성되어 있으며, 프로그래밍 언어 선정에 대한 기준 및 활용 예시 등은 ‘컴퓨터 프로그래밍’ 단원에 제시되어 있다 [18][21].

신승기, 배영권(2014, December)은 인도의 소프트웨어 관련 교육과정을 분석하여[21], ‘컴퓨터 프로그래밍’ 단원에서 학년별 프로그래밍 활용 기준 및 예시에 대하여 아래 <Table 9>과 같이 설명하였다[18]. 9-12학년의 내용은 CMC 공통교육과정과는 별도로 <Table 9>의

내용과 같이 제시되었다[21].

<Table 9> Programming Tools in Curriculum of India

Grades	Programming Resources
3-5	GUI(Graphic User Interface) based Programming language: Scratch
6-8	Syntax based programming language: BASIC
9-12	Programming languages: machine language, assembly language, high level language(BASIC or C++)

초등학교 고학년에 해당하는 3-5학년에서는 그래픽 기반의 프로그래밍 언어인 스크래치를 활용하여 프로그래밍에 대한 기초개념을 익히며, 중학교에 해당하는 6-8학년에서는 문법(Syntax)기반의 프로그래밍 언어 즉, 텍스트기반 프로그래밍 언어인 베이직을 활용하여 프로그래밍에 대한 교수학습을 실시하도록 하였다. 고등학교에 해당하는 9-12학년에서는 실제 프로그래밍에 활용되는 언어들을 중심으로 활용함으로써, 고급프로그래밍언어를 학습할 수 있도록 기준을 제시하였다 [18][21].

또한, 6학년과정에서는 비주얼기반프로그래밍 언어에서 텍스트기반프로그래밍 언어로 넘어가는 과도기적의 과정을 갖기 위해, 순서도(Flow-chart)와 유사코드(Pseudo-code)를 활용하여 학습의 전이가 일어날 수 있도록 교육과정이 편성되어 있다[21].

3.6 호주의 소프트웨어 교육과정 분석

호주는 소프트웨어 관련 교육으로 코딩교육을 국가 수준교육과정에 필수정규교육과정으로 초등학교에서부터 실시하기로 최근 2015년 9월 결정하였다[16]. 호주의 소프트웨어 관련 교육은 테크놀로지(Technologies)교과군의 디지털 테크놀로지(Digital Technologies)교과에서 실시되며, 코딩교육은 초등학교 5-6학년(만 10-11세)에서 실시되고, 프로그래밍 교육은 중학교과정인 7학년(만 12세)부터 고등학교과정까지 실시된다고 밝히고 있다[4][16]. 호주에서는 코딩교육과 프로그래밍 교육에 대한 용어와 목적을 달리 사용하고 있다. 이는 호주의 교육과정의 디지털 테크놀로지교과에서 학년별로 적용하는 프로그래밍 언어가 무엇인지를 살펴보면 명확하게

구분이 된다. 호주의 교육과정에 나타난 학년별 프로그래밍 언어는 다음 <Table 10>과 같이 나타난다.

<Table 10> Programming Tools in Curriculum of Australia

Grades	Programming Resources
F-2 Grade ¹⁾	Robotic Devices
3-4 Grade	Visual Programming Languages
5-6 Grade	Visual Programming Languages
7-8 Grade	General-purpose Programming Language
9-10 Grade	Object-oriented Programming Language

<Table 10>에 나타난 General-purpose Programming Language는 텍스트기반의 범용 프로그래밍 언어(C, Java 등)를 의미하며[22], Object-oriented Programming Language는 객체지향 프로그래밍 언어로써, General-purpose Programming Language의 기능을 포함한 상위 개념의 프로그래밍 언어(Python, Delphi, C++, C# 등)를 의미한다[23].

<Table 10>에서 살펴볼 수 있는 바와 같이, 초등학교 과정에서는 비주얼기반의 프로그래밍언어를 활용하며, 중학교단계부터는 텍스트기반의 프로그래밍 언어를 활용하는 것을 살펴볼 수 있다.

즉, 호주의 소프트웨어교육 관련 교육과정에서 초등학교과정에서 사용하는 ‘코딩교육’이라는 용어는 교육용으로 주로 활용되고 있는 비주얼프로그래밍 언어를 의미하며, 중학교 이상의 과정에서 제시하고 있는 ‘프로그래밍교육’의 의미는 텍스트기반의 프로그래밍 언어를 의미함을 살펴볼 수 있다.

호주의 소프트웨어 관련 교육에서 코딩교육이 5-6학년년부터 시작됨에도 불구하고, F-2학년에서 로봇 교구를 활용하는 까닭은 실제 코딩교육이 실시되는 5-6학년 이전에 문제해결과정에 대한 개념을 이해하기 위한 초기단계로써 절차적 접근을 위해 학습자의 수준을 고려하여 친숙한 소재로 제시된 것이며, 3-4학년에서 비주얼 프로그래밍 언어가 제시된 것은, 문제해결과정의 설계 및 알고리즘의 이해 등을 위한 도구로 제시된 것으로 코딩교육을 위한 사전단계로 제시된 것으로, 코딩교육으로 진행되기 전의 과도기적 단계가 마련된 것이다 [4].

1) F(Foundation)학년은 유치원 단계를 의미한다[11].

4. 프로그래밍 언어 선정 기준 및 사례

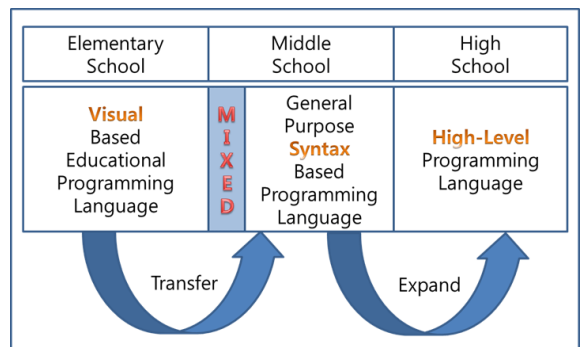
4.1 프로그래밍 언어 선정 기준

앞서 살펴본 내용을 토대로 필수정규교육과정으로써 소프트웨어 교육관련 코딩교육 혹은 프로그래밍 교육을 국가수준교육과정으로 운영하고 있는 사례들을 통해 학교 급별 운영되고 있는 프로그래밍 언어들을 다음 <Table 11>과 같이 정리할 수 있다.

<Table 11> Programming Tools from Case Studies

Countries	Grade		
	Primary	Middle	High
U.K.		VPL ↓	Thinking Process
Finland		TPL	x
Estonia	VPL (Visual Programming Language)	&	Web Programming
India		TPL (Textual Programming Language)	High level Programming
Australia			Object Oriented Programming

이를 토대로 소프트웨어 관련 교육을 국가수준의 필수정규교육과정으로 적용한 국가들의 프로그래밍 언어 선정기준에 대한 공통점을 정리한 것은 다음 (Fig. 1)과 같이 나타난다.



(Fig. 1) Common Standard for Programming Language Selecting from Case Studies

첫째, 초등학교에서는 비주얼기반의 프로그래밍 언어를 활용하며, 프로그래밍 언어와 연동되는 구체물을 이용한다.

둘째, 중학교부터는 텍스트기반의 프로그래밍 언어를 활용한다.

셋째, 비주얼 프로그래밍 언어에서 텍스트 기반 프로그래밍언어로 넘어가는 시기를 과도기로 설정하여 관련성에 대한 집중 지도가 이루어진다.

넷째, 고등학교에서는 응용프로그래밍개발 등의 실제적 프로그래밍 활동을 위한 프로그래밍 언어를 활용하여 학생들의 경험을 확산시킨다.

4.2 프로그래밍 언어 선정 사례

소프트웨어 교육 관련 외국의 국가수준교육과정을 살펴보면, 교육과정 내에서 프로그래밍 언어 선정기준에 따라 프로그래밍 언어를 예시로 직접 제시하는 국가들이 있다. 본 연구에서 살펴본 다섯 개의 국가들 중에서, 현재 교육과정을 개정 중인 호주를 제외한 나머지 국가(영국, 핀란드, 에스토니아, 인도)에서는 학년 및 학교 급을 고려하여 적절한 프로그래밍 언어를 제시하고 있다[5][8][10][18][19][20][21]. 그중에서 프로그래밍 언어 학습 및 구체물을 활용한 프로그래밍을 위한 교구를 제시한 내용을 제외하고 프로그래밍 언어 자체에 대한 내용이 제시된 부분을 살펴보면 아래의 <Table 12>와 같이 나타났다.

<Table 12> Suggested Programming Languages from Case Studies

Countries	Grade		
	Primary	Middle	High
U.K.	Scratch	.	.
Finland	Scratch	Khan Academy ²⁾	.
Estonia	Scratch Kodu Logo	NXT-G NXC	.
India	Scratch	BASIC	BASIC C++

2) Javascript, HTML&CSS 등 텍스트기반 언어를 학습[9]

<Table 12>에서 살펴볼 수 있는 바와 같이 현재 우리나라의 소프트웨어 교육의 교육용 프로그래밍 언어로써 많이 활용되고 있는 스크래치(Scratch) 프로그래밍 언어는 해외의 경우 초등학교에서 집중적으로 교수학습이 이루어지는 것을 살펴볼 수 있다.

4.3 영재교육 및 수준별 학습

우리나라의 소프트웨어 교육에 대한 전면 도입과 함께 언급되는 우려 중의 하나는 수준별 학습에 대한 내용과 영재교육에 대한 내용일 것이다. 수준별 학습과 영재교육의 교수학습 내용 선정 및 방법에 대한 내용은 과거 컴퓨터교과가 도입된 이후로 지속적으로 연구되고 있고 있는 부분이기도 하다.

본 연구에서 살펴본 다섯 국가의 교육과정에서도 대부분 이에 대한 내용이 나타나 있지 않지만, 영국의 교육과정에서는 이에 대해서 가이드라인을 제시하고 있다 [5].

초등학교의 경우 비주얼프로그래밍 언어사용을 원칙으로 하지만, 일반학급에서 우수한 학생들을 대상으로 텍스트기반 프로그래밍 언어인 Python을 소개하는 정도로 수준별 학습을 진행할 수 있다고 제시하고 있다 [5].

또한 초등학교의 영재학생들에 대해서는 알고리즘과 프로그램 개발에 초점을 맞추어 교수학습을 진행하되 컴퓨팅 사고력(Computational Thinking)이 적용되도록 수업이 구성되어야 하며, 텍스트기반 프로그래밍 언어인 Logo나 Python을 활용하는 것이 적절하다고 밝히고 있다[5].

5. 우리나라 소프트웨어 교육과정의 프로그래밍 언어 선정을 위한 시사점 분석

5.1 2015개정교육과정에서 나타난 프로그래밍 언어 선정 기준

우리나라의 소프트웨어교육과정에 대한 내용은 크게 두 가지의 고시된 문서로 나타난다. 첫 번째는 2015년 2월 고시된 ‘소프트웨어 교육 운영 지침’으로, 2014년 7월

발표된 ‘소프트웨어 중심 사회 실현 전략 발표’에 따라 [14], ‘소프트웨어 중심사회’로 나아가기 위한 방안으로 교육 분야에서 소프트웨어 부문에 대한 추진을 위해 교육과정 편성 기준 및 운영 방법 등에 대한 내용을 제시하고 있다[13]. 또한 소프트웨어 교육에 대해서 ‘단순한 프로그래밍 교육이 아니라 실생활에서 주어지는 복잡한 문제를 컴퓨팅 사고력을 기반으로 해결하는 것이 주된 활동이며... (해설서 p.56.)’라고 정의하고 있으며, ‘소프트웨어 교육은 컴퓨팅 사고력을 가진 창의융합 인재의 양성을 목표로 하기 때문에 프로그래밍 언어의 사용법과 같은 도구의 사용 방법을 교육하는 것이 아니라 사고력을 발현할 수 있는 학습 경험을 제공하는 것이 중요하다(해설서 p.56)’라고 밝히고 있다[13]. 즉, 단순한 프로그래밍 교육이 아니라 일상생활 속에서 나타나는 문제들을 컴퓨팅 사고력을 기반으로 프로그래밍 활동을 통해 해결해 나가는 과정임을 의미하고 있다.

프로그래밍 언어는 컴퓨팅 사고를 구현하기 위한 문제해결 도구이기 때문에 소프트웨어교육 관련 교과에서 활용하기 위한 중심 도구이며, 이는 ‘소프트웨어 교육 운영 지침’의 해설서에 제시된 교수학습 방법에서 ‘컴퓨팅 사고력을 기반으로 문제 해결 과정을 설계하고 이를 프로그래밍을 통해 실현할 수 있도록 이수기간 및 시간을 편성한다(p.55)’라는 부분에서 명확히 확인할 수 있다.

즉, 프로그래밍 언어 자체에 대한 학습을 지양하고 컴퓨팅 사고력기반의 문제해결력을 기르는 것이 목적이라는 의미이며, 프로그래밍 언어는 핵심 도구으로써 활용이 이루어져야 한다는 의미이다. 그러나, ‘소프트웨어 교육 운영 지침’에서 프로그래밍 언어 선정에 대한 방법 또는 기준에 대한 언급이 없어서 학교 현장에서 학교 급에 따라 어떤 프로그래밍 언어를 활용해야 할지에 대한 의문을 남긴다. 예를 들어, 비주얼프로그래밍 언어를 초-중-고 모두 활용하는 것이 적절한 것인지, 또는 초등학교에서만 지도해야 하는 것인지 등에 대한 기준이 없어서 국가주도의 집중적인 소프트웨어 교육 추진에 대해 어려움을 겪을 수 있는 부분 중 하나로 예측된다.

소프트웨어교육과정에 대한 두 번째 고시문은 2015년 9월 발표된 ‘2015개정교육과정에서의 정보교과의 교육과정’에 대한 내용이다[12]. 급별 개정교육과정에 편

성된 다섯 단원 중에서 프로그래밍과 관련된 단원은 ‘문제 해결과 프로그래밍’이다. 이와 관련된 교과 목표로서 ‘컴퓨팅 원리에 따라 문제를 추상화하여 해법을 설계하고 프로그래밍 과정을 통해 소프트웨어로 구현하여 자동화할 수 있는 능력을 기른다(p.223)’라고 제시하고 있다[13].

이에 대한 성취기준으로 [9정04-01]에 ‘사용할 프로그래밍 언어의 개발 환경 및 특성을 이해한다(p.229)’라고 제시하고 있으며, 교수학습방법 및 유의 사항으로 ‘학습자 수준에 적절한 교육용 프로그래밍 언어를 선택한다(p.230)’와 ‘특정 프로그래밍 언어의 기능 습득에 치중하지 않도록 유의하고 문제 해결을 위한 프로그램 설계 및 개발 과정을 통해 컴퓨팅 사고력을 신장하는데 초점을 둔다(p.230)’라고 명시되어 있다. 즉, 컴퓨팅 사고력을 기반으로 문제해결능력을 기르기 위한 도구로써 프로그래밍 언어를 사용하되, 지나치게 언어 자체에 대한 기능 습득은 지양하라는 의미로 해석된다.

문제해결과정에 초점을 두더라도 프로그래밍 언어는 도구로써 활용해야 하기 때문에, 프로그래밍 언어가 선정이 우선적으로 실시되어야 할 필요가 있다. 소프트웨어 교육을 국가수준교육과정에서 필수정규교육과정으로 제시하고 있는 외국의 사례에서 어떤 형태의 프로그래밍 언어가 학교 급별 또는 학년 군별로 적절한지에 대한 국가적인 검토를 통해 국가수준 교육과정에서 제시하고 있는 반면, 우리나라의 소프트웨어 교육 관련 교육과정에 제시된 프로그래밍 언어 선정 기준은 다소 모호하게 수립이 되었다고 볼 수 있다.

5.2 내실 있는 소프트웨어 교육과정운영을 위한 프로그래밍 언어 선정 기준 제언

본 연구에서 분석한 해외의 국가수준교육과정에 대한 내용을 토대로 우리나라의 소프트웨어 교육 관련 교과(정보)의 내실 있는 운영과, 소프트웨어 교육의 안착 및 이를 토대로 미래 인재양성과 국가경쟁력 확보를 위해서 다음 (Fig. 2)와 같이 학년 및 학교 급에 따른 프로그래밍 언어 선정 기준을 제시하고자 한다. 이는 신승기, 배영권(2015, September)의 에스토니아의 소프트웨어 교육과정 분석을 통해 제시한 제언과 맥락을 같이 한다[20].

Level	Grade	Type of Programming Language
Elementary School	1	Unplugged
	2	
	3	Visual Programming Language (VPL)
	4	
	5	
	6	
Middle School	1	Visual + Textual
	2	Textual Programming Language (TPL)
	3	
High School	1	Textual Programming Language (TPL)
	2	
	3	

(Fig. 2) Suggestion for Software Education Curriculum of Korea

초등학교는 학습자의 발달단계를 고려하여 저학년과 중·고학년으로 나누고, 저학년에서는 컴퓨터에 친숙해질 수 있는 활동과 컴퓨터과학에 대한 기본개념을 언플러그드활동을 통해 놀이로써 친숙하게 접하여 흥미를 갖는데 목표를 둔다. 중·고학년에서는 비주얼 기반의 프로그래밍 언어를 활용하여, 제시된 프로그램을 따라서 만들거나 간단히 수정하는 활동을 통해 문제해결을 위한 프로그래밍 활동의 개념을 익히도록 한다.

중학교 1학년에서는 전이(Transfer)과정을 갖도록 한다. 다시 말해, 비주얼 프로그래밍 언어와 텍스트 기반의 프로그래밍 언어를 함께 제시하여 상관관계를 살펴보고 유사점을 알아보는 활동 등을 통해 학습의 경험이 전이되도록 한다.

중학교 2-3학년에서는 초등학교과정에서 학습한 문제해결절차 및 프로그래밍과정과 방법에 대한 개념을 토대로 텍스트기반의 프로그래밍 언어를 활용하여 간단한 문제 해결을 위한 설계 및 프로그래밍을 통한 해결과정을 경험하도록 한다.

고등학교 과정에서는 하나의 TPL에서 확산(Expand)하여 여러 가지 상위 프로그래밍 언어들에 대한 개념(객체, 클래스 개념 등)을 익히고, 이를 토대로 우리 주변에서 발생하는 실제 문제를 해결하기 위해 문제해결 설계과정을 토대로 프로그래밍을 활용한 문제해결과정을 경험하도록 한다.

6. 결론 및 제언

소프트웨어교육에 대한 요구는 더 이상 고려할 대상이 아니라, 21세기를 살아가는 모든 사람들에게 반드시 필요한 필수 요소이다[15][24]. 정보화 사회 이전에 반드시 갖추어야 할 소양으로 3Rs(Reading, wRiting, aRithmetic)가 언급되었다면, 이제는 3Rs + Computational Thinking으로[24], 컴퓨팅 사고력을 기반으로 한 코딩 능력, 즉 소프트웨어 관련 소양이 반드시 필요한 시대를 살아가고 있다. 이를 위해 세계 각국에서는 국가경쟁력확보를 위해 소프트웨어 관련 교과를 국가수준교육과정에서 필수정규교육과정으로 반영하여 초등학교에서 고등학교로 이어지는 체계적인 교육과정을 수립하여, 미래인재를 양성하고 나아가 미래 국가 경쟁력을 확보하기 위해 추진되고 있다. 이를 위해서, 소프트웨어교육 관련 교과의 핵심내용을 코딩 또는 프로그래밍으로 간주하여 학교 급에 따라 어떤 프로그래밍언어를 활용하는 것이 교육적 효과를 높일 수 있는지에 대해 국가적 차원에서 고민하고, 이를 국가수준 교육과정에 명시하였다.

소프트웨어 교육에 대한 용어와 프로그래밍 교육에 대한 용어가 현재 혼용되는 경우가 많으나, 2015년에 발표된 ‘소프트웨어 교육 운영 지침[13]’과 소프트웨어 교육 정책이 반영된 2015 개정교육과정의 ‘정보’ 교육과정 해설서[12]에서 프로그래밍 관련 교육 내용이 하나의 세부단원 및 내용요소로써 제시되어 있으므로, 소프트웨어 교육은 프로그래밍 교육을 포함한 상위단계의 소프트웨어 산업까지를 아우를 수 있는 교수학습이라고 할 수 있다.

본 연구에서 살펴본 해외 다섯 국가의 국가수준 교육과정을 살펴보면, 초등학교에서는 비주얼기반의 프로그래밍 언어를 선정하고, 중학교에서는 텍스트기반의 프

로그래밍 언어를 선정하되, 비주얼기반의 프로그래밍 언어에서 텍스트기반 프로그래밍 언어로 넘어가는 과도기적 단계를 모두 갖고 있었다. 이후 고등학교에서는 응용프로그램 개발 등의 고급 프로그래밍 언어를 교육 과정에 제시하고 있었다. 또한 현재 교육과정을 지속 개발하고 있는 호주를 제외한 나머지 네 개 국가에서는 국가수준교육과정에서 학교급과 학년을 고려하여 어떤 프로그래밍 언어를 어느 시기에 활용하는 것이 효과적인지 실제 프로그래밍 언어 명칭을 함께 기입하여 실제적인 도움이 되도록 구성되어 있다.

국가별로 교육환경 및 문화적 차이 등으로 일률적인 교육과정 편성이 이루어질 수 없으며, 교육과정에 대한 상호 비교 및 평가는 가능하지 않은 부분이다. 그러나 이미 소프트웨어 교육 관련 교육과정을 시행하고 있는 국가들의 사례를 통해서, 우리나라의 개정교육과정의 전면 시행을 앞두고 참고가 될 수 있는 사항들을 고려해 볼 수 있는 모델로 삼을 수 있다. 또한 향후 연구에서는 학교 급에 따른 프로그래밍 언어 선정 기준에 대한 전문가 의견 인터뷰 또는 설문 분석을 추가로 실시하여 본 연구에서 제시하고 있는 기준에 대한 타당도와 신뢰도를 뒷받침하고자 한다.

교육과정의 실천가는 교사이며, 수혜자는 학생이다. 교육과정이 개정되면 가장 고민하고 노력해야 할 사람도 현장교사이다. 만약 국가수준에서 모든 현장교사들이 겪어야 할 고민들을 사전에 국가적 차원에서 합리적인 근거와 연구를 통해 제시해 줄 수 있다면, 분명 학교현장의 부담을 덜어 줄 수 있는 핵심적인 부분이 될 수 있을 것이다. 물론 학교단위 교육과정이 편성되는 상황에서 모든 상황과 여건을 고려할 수는 없겠지만, 소프트웨어 교육의 핵심인 컴퓨팅 사고력 기반의 프로그래밍 교육을 위한 도구 선정의 가이드라인이 제시된다면, 우리나라의 소프트웨어 교육이 더욱 안정감 있게 적용되고 추진될 수 있을 것이다. 아울러 세계적 수준의 하드웨어 인프라를 통해 세계 일류 소프트웨어 교육 강국 및 소프트웨어 산업 강국으로 나아갈 수 있는 기반이 마련될 수 있을 것이다.

참고문헌

- [1] CAS (Computing At School). Who we are. Retrieved from <http://www.computingatschool.org.uk/about>
- [2] Department for Education in U.K (2014, June). Interim KS5 minimum standards. Retrieved from <https://www.gov.uk/government/publications/interim-ks5-minimum-standards>
- [3] Ministry of Education, Korea (2015). The Elementary and secondary school curriculum. #2015-74 (Annex 1).
- [4] Australian Curriculum (2015). Curriculum of Digital Technologies. Retrieved from <http://www.australiancurriculum.edu.au/technologies/digital-technologies/curriculum/f-10?layout=1>
- [5] CAS (2013). Computing in the national curriculum; A guide for primary teachers. Retrieved from <http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf>
- [6] Department for Education in U.K. (2013, September). National curriculum in England: computing programmes of study. Retrieved from <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>
- [7] Dredge, S. (2014, September). Coding at school: a parent's guide to England's new computing curriculum. *theguardian*. Retrieved from <http://www.theguardian.com/technology/2014/sep/04/coding-school-computing-children-programming>
- [8] Innovation Centre in HITSA. Programming at Schools and Hobby Clubs. Retrieved from <http://www.innovatsioonikeskus.ee/en/programming-schools-and-hobby-clubs>
- [9] Khan Academy (2015). Computer programming. Retrieved from <https://www.khanacademy.org/computing/computer-programming>
- [10] Koodi2016 (2014). Koodi2016-ensiapua ohjelmoinnin opettamiseen peruskoulussa. Retrieved from

- https://s3-eu-west-1.amazonaws.com/koodi2016/Koodi2016_LR.pdf
- [11] Lau-Kee, D., Billyard, A., Faichney, R., Kozato, Y., Otto, P., Smith, M., & Wilkinson, I. (1991, October). VPL: an active, declarative visual programming system. In *Visual Languages, 1991., Proceedings. 1991 IEEE Workshop on* (pp. 40-46). IEEE.
- [12] Ministry of Education, Korea (2015). *Informatics Curriculum. #2015-74 (Annex 10)*.
- [13] Ministry of Education, Korea (2015). *Software Education Instructional Guidance*.
- [14] Ministry of Science, ICT and Future Planning (2014, July). *Software-driven society realization strategy briefing*.
- [15] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- [16] Sadler, D. (2015, September). Tech sector cheers on new PM Malcolm Turnbull. startup smart. Retrieved from <http://www.startupsmart.com.au/planning/tech-sector-cheers-on-new-pm-malcolm-turnbull/2015091515517.html>
- [17] School education (2015, December). Department of Education and Training, Australian Government. Retrieved from <https://www.education.gov.au/school-education>
- [18] Shin, S., Bae, Y. (2014, December). Analysis and Implication about Elementary Computer Education in India. *Journal of The Korean Association of Information Education*, 18(4), 585-594. doi: <http://dx.doi.org/10.14352/jkaie.2014.18.4.585>
- [19] Shin, S., Bae, Y. (2015, March). Review of Software Education based on the Coding in Finland. *Journal of The Korean Association of Information Education*, 19(1), 127-138. doi: <http://dx.doi.org/10.14352/jkaie.2015.19.1.127>
- [20] Shin, S., Bae, Y. (2015, September). Study on the Implications about Curriculum Design through the Analysis of Software Education Policy in Estonia. *Journal of The Korean Association of Information Education*, 19(3), 361-372. doi: <http://dx.doi.org/10.14352/jkaie.2015.19.3.361>
- [21] Sridhar Iyer, Farida Khan, Sahana Murthy (2013). *CMC: A Model Computer Science Curriculum for K-12 Schools*. 3rd Edition. Indian Institute of Technology Bombay, Mumbai. Retrieved from <http://www.cse.iitb.ac.in/internal/techreports/reports/TR-CSE-2013-52.pdf>
- [22] Wikipedia. General-purpose programming language. Retrieved from https://en.wikipedia.org/wiki/General-purpose_programming_language
- [23] Wikipedia. Object-oriented programming language. Retrieved from https://en.wikipedia.org/wiki/Object-oriented_programming
- [24] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

저자소개

신 승 기



2007 한국교원대학교 컴퓨터교육과(교육학사)
 2009 아주대학교 정보통신대학원(공학석사)
 2012 대구교육대학교 컴퓨터교육과(교육학석사)
 2014~현재 The University of Georgia Ph.D Student
 관심분야: 소프트웨어교육, Computational Thinking
 e-mail: shin@uga.edu



배 영 권

2006 한국교원대학교 컴퓨터교육
과(교육학박사)

2006~2007 Indiana University
VisitingScholar

2007~2009 목원대학교 컴퓨터교
육과 교수

2013~2014 The University of
Georgia VisitingScholar

2009~현재 대구교육대학교 컴퓨
터교육과 교수

관심분야: 소프트웨어교육, STEAM
교육, 영재교육

e-mail: bae@dnue.ac.kr