

코딩 교육에서 사회적 구성의 의미에 관한 고찰

김홍래

춘천교육대학교 컴퓨터교육과

요약

본 연구는 학교에서 SW와 인공지능 교육에 대한 관심이 확대되는 시점에서 코딩을 가르치는 방법의 이론적 배경을 탐구하고자 하였다. 학교 현장의 코딩 교육은 매우 다양한 형태로 나타난다. 학교의 코딩 교육의 실제적 장면에서 어떻게 가르칠 것인가는 매우 중요한 과제이다. 이를 위하여 코딩 교육을 위한 사회적 구성주의의 개념과 페퍼트의 구성주의 접근 방식을 탐구하였다. 페퍼트의 '생각하게 하는 사물'의 비유를 통하여 사회적 지식의 구성의 의미를 분석하였다. 코딩 교육을 위한 사회적 구성주의적 접근 모형 5가지를 분석하였으며, 이를 바탕으로 코딩 교육을 위한 학습 전략을 제시하였다. 그것은 만들기, 텅커링하기, 협력과 공유하기, 스캐폴딩과 학습의 주도성, 실제적 객체의 활용 등이다. 코딩 교육에 관한 사회적 구성주의에 대한 탐구와 관련 연구가 학교 현장의 교육에 도움이 되기를 기대한다.

키워드 : 코딩, 컴퓨터과학교육, 구성주의, 사회적 구성주의, 시모어 페퍼트

A Study on the meaning of Social Constructionist Approaches to Coding Education

Kim, Hongrae

Chuncheon National University of Education

Abstract

This study attempted to explore the theoretical background of the method of teaching coding at a time when interest in software and artificial intelligence education in schools is expanding. Coding education in the school field appears in a wide variety of forms. How to teach in the practical scene of coding education in schools is a very important task. For this, the concept of social constructivism for coding education and Papert's approach to constructionism were explored. The meaning of the composition of social knowledge was analyzed through Papert's analogy of 'objects to think with'. Five social constructionist approaches to coding education were analyzed, and based on this, a learning strategy for coding education was suggested. It includes making, tinkering, cooperation and sharing, scaffolding and agency in learning, and the use of tangible media. It is hoped that the exploration of social constructionism in coding education and related research will be helpful to the educational method in the school field.

Keywords : Coding, Computer Science Education, Constructivism, Social Constructionism, Seymour Papert

1. 서론

컴퓨터는 30여 년 전 진보적인 사상을 가진 학자들에 의하여 교육적 목적으로 도입되었다. 그들은 어린이들이 컴퓨터를 이용하여 기성세대와는 달리 보다 새로운 것을 발견할 수 있을 것이라는 믿음이 있었다. 그러나 컴퓨팅 기술의 발전은 교육의 본질적 목적에 기여하기보다 오히려 교육적 수단을 강화하였다. 학교는 컴퓨터를 활용하여 어린이들의 창의성과 상상력을 기르기보다 ICT교육이라는 이름으로 워드프로세서와 같은 도구의 학습에 몰입하였다. 학교에서 코딩 교육의 부활은 교육에서의 도구적 활용의 한계를 인정하고 교육의 본질적인 가치를 회복하고자 하는 진보적 학자들의 생각을 다시 받아들인 것이라고 볼 수 있다.

1980년대에 시모어 페퍼트(Seymour Papert)는 컴퓨터를 사교의 도구로 인식하고 프로그래밍 언어인 'LOGO'를 학습의 중심에 두고자 하였다. 단순히 LOGO 언어 자체를 배우는 것이 목적이 아니라 LOGO를 통하여 사고력을 증진하고자 하였다. LOGO는 프로그래밍을 통하여 다양한 교육과정에 융합될 가능성이 많았음에도 불구하고 사교의 도구로서 확장되기보다는 워드프로세서, 스포레드시트와 같은 도구에 그 자리를 내어주고 말았다. 학교에서의 컴퓨터는 전통적인 교육 환경에서 학습을 보다 더 효율적으로 도와주는 도구 이상의 가치를 가질 수 없었다[1][17].

실비아 마르티네즈(2019)는 페퍼트의 말을 빌려 듀이의 '경험에 의한 학습'은 실패하지 않았지만, 시대를 너무 앞서갔다고 지적하고 현재는 그 아이디어를 구현할 수 있는 환경이 갖추어졌음을 3가지 관점에서 제시하였다[1]. 첫째, 학생들이 변화의 주체가 되었다. 전통적인 학교는 학생들에게 목표에 도달할 것을 요구했지만, 지금의 학생들은 학교가 자신들의 요구에 부합하도록 변화할 것을 요구한다. 둘째, 컴퓨터가 교사를 위한 기술이 아니라 학습자를 위한 기술이다. 보급형 컴퓨터, 센서, 프로그래밍 도구들은 학습자들이 학습의 주인이 되도록 하였다. 셋째, 과거에는 대학 수업에서나 가능했던 복잡한 아이디어들이 어린이들도 이해할 수 있는 수준으로 제공된다. 전문가들의 경험을 학교에서도 접할 수 있는 환경이 도래했다고 보는 것이다.

최근 코딩 교육에 대한 다양한 연구와 사례들이 발표

되고 있으나 대부분 교육학적 배경에 대한 탐색은 충분하지 않은 것으로 판단된다. 코딩을 가르치는 교사들의 접근 방식에 있어서 구성주의적 접근에 대한 논의는 빈약하다. 코딩을 지도하는 교사들을 위하여 교육부 소속 기관이나 유관기관에서 제공하는 많은 자료들은 교육학적 배경에 대한 충분한 정보가 생략되어 있다. 학교에서의 코딩 교육은 언플러그드, 블록 코딩, 피지컬 컴퓨팅, 로봇을 활용한 코딩 등 다양한 도구와 사례들이 제시되고 있으나 그 교육학적 배경을 충분히 드러내지 않았다. 대부분의 수업들은 교사들이 학습 내용과 방법을 결정하고 학생들은 따라하는 형태의 수업으로 학생들의 자유도가 매우 제한적인 수업이 이루어지기도 한다.

초등학생 대상 코딩 교육은 단순히 프로그래밍 도구를 이용하여 블록을 조립하는 것을 목적으로 하지 않는다. 문제의 이해, 목표 설정, 해결 방법을 탐구하는 창의적이고 열린 문제 해결 능력을 기르기 위한 것이다. 블록 프로그래밍을 통한 문제 해결이나 피지컬 컴퓨팅을 통한 문제 해결을 하는 수업에서 학습 환경에 대한 이해, 도달해야 하는 목표와 교사의 역할과 학생들의 활동에 대한 교육학적 이해와 가이드라인이 매우 필요하다. 대부분의 코딩 교육 프로그램이 주로 엔트리와 스크래치 같은 블록 프로그래밍 도구를 이용한 절차와 방법을 설명하는 내용으로 구성되어 있다는 점에서 이론적 탐구는 더욱 중요하다.

본 논문은 교육 분야에서 코딩 교육에 대한 중요성이 사회적으로 확대되고 있고 학교의 정보기술과 관련된 물리적 환경이 개선되고 있는 상황에서 학습자 중심의 수업을 위한 접근 방법으로서 사회적 구성주의적 관점을 분석하고 학교에서의 실천 방안을 제시하고자 하였다.

2. 사회적 구성주의의 개념과 의미

2.1 Constructivism과 Constructionism

2.1.1 Piaget의 구성주의(Constructivism)

Piaget는 인간이 어떻게 보편적이고 타당한 수학적 지식을 알게 되는가라는 질문에 대하여 인간이 지식을 '구성'하는 것이라고 주장하였다. 이것은 그 이전까지의

객관적 세계에 대한 이해라는 관점을 개인이 인식의 주체로 전환하는 것이었다. 사람은 자신이 이미 알고 있는 지식에 새로운 경험을 연결함으로써 새로운 지식을 능동적으로 구성한다고 보았다[8]. 구성주의는 지식을 학습자에게 전달하는 것이 아니라, 학습자의 두뇌에서 구성되는 것으로 인식한다. 새로운 지식은 문제 상황에서 학습자가 이미 알고 있는 지식과 결합하여 새로운 상황을 이해하는 과정에서 발생한다. 피아제의 구성주의는 학습자가 세상의 문제 상황과의 상호작용을 통한 개인적인 학습 과정이라고 본다.

2.1.2 Papert의 구성주의(Constructionism)

일반적으로 사회적 구성주의에서 인간의 인식 작용은 개인과 객체 간에만 이루어지는 것이 아니라 ‘사회’ 속에서 이루어진다는 점이다. 학습은 다른 사람들과 대화를 나누거나 함께 일하는 과정에서 구성될 수 있다. 지식의 구성은 개인적 과정과 사회적 과정이 긴밀한 상호작용을 통하여 이루어진다. 즉, 학습은 사회적으로 의미 있는 활동에 참여함으로써 가장 효과적으로 이루어진다.

페퍼트는 1958년부터 19663년까지 피아제와 함께 제네바 대학에서 연구하였다. 그는 구성주의 이론을 바탕으로 MIT로 옮겨 ‘인식론과 학습 연구’ 그룹을 만들었다. 페퍼트는 피아제의 발생론적 인식론을 바탕으로 인지를 강조한 피아제의 인식론을 정의적 영역까지 확대하려고 하였다[8]. 페퍼트에 의하면,

“피아제의 구성주의 이론들로부터 우리는 학습을 지식의 전달이 아니라 지식의 재구성으로 바라보게 되었다. 우리는 이 관점을 확대하여, 학습자가 의미 있는 물건을 만들어보는 것처럼 직접적인 경험이 학습에 가장 효과적이라고 생각한다[17].”

Kafai와 Resnick(1996)도 “지식은 교사로부터 학생에게 단순히 전달되는 것이 아니라, 능동적으로 학습자의 마음에 구성된다. 어린이들은 아이디어를 배우지 않는다. 그들은 아이디어를 창조한다”고 그의 주장을 뒷받침한다[1][20].

페퍼트는 어린이들이 스스로 지식을 구성한다는 인지적 구조 모델을 피아제로부터 이어받았다. 어린이는 학교에 가기 전부터 ‘피아제식 학습(Piagetian learning)’ 또는 ‘가르침 없는 학습(learning without being taught)’을 통하여 매우 많은 양의 지식을 습득하지만 특정 유

형의 학습은 가르침을 통해서도 잘 이루어지지 않는다는 것에 의문을 가졌다. 이에 대하여 페퍼트는 어린이를 ‘만드는 자로써의 어린이(child as builder)’[17]로 인식하고 무엇인가를 만들고자 하는 사람들은 반드시 재료가 필요한 것처럼 재료가 지식 구성의 핵심적인 역할을 한다고 생각하였다. 따라서 어린이들에게 문제 해결을 위한 풍부한 재료가 제공된다면 구성주의적인 피아제식 학습이 쉽게 일어날 수 있다고 하였다. 특히, 피아제의 발생론적 인식론과의 차이에 대하여 다음과 같이 설명한다[17].

“많은 경우에 있어서, 피아제는 특정 개념의 발달이 늦어지는 것은 그것이 매우 복잡하거나 형식적이기 때문이며 반면 내 생각에 가장 중요한 요인은 개념을 단순화하고 구체화하는 재료가 빈약한 문화적 배경 때문이라고 생각한다.

페퍼트는 컴퓨터를 사람들이 생각하거나 만들고자 하는 것을 구현하기 위해 사용하는 재료라고 보았다. 마찬가지로 그의 생각을 가장 잘 드러낸 것은 ‘생각하게 하는 사물(objects to think with)’인 ‘로고 거북이’이다[17]. 현재 시점에서 개념을 습득하기 위한 재료는 스크래치, 엔트리, 레고 EV3, 피지컬 컴퓨팅 도구들이다. 이들이 ‘생각하게 하는 사물’이며 개념을 연결하는 다리이다.

2.2 ‘사회적 구성’의 의미

코딩 관련 지식이 사회적으로 구성되는 맥락을 다음과 같이 두 가지의 형태로 생각할 수 있다. 하나는 ‘개인이 구성한 주관적 지식이 사회의 객관적인 지식이 되는 과정’과 그 반대로 ‘사회의 객관적인 지식이 개인의 주관적인 지식이 되는 과정’이다[2]. 전자의 경우, 사회적으로 학자나 전문가, 창의적인 사람에 의하여 새로운 이론이나 주장이 공식적으로 공표되면 관련 분야에서 동일한 지식 구조에 대한 조사와 실험이 이루어지고 평가를 통하여 공인된다. 그러나 학교 교실과 같은 공간에서 이와 같은 전문적인 경험을 하는 것은 어려운 일이다.

그렇다면 코딩 학습에서 개인의 주관적인 지식은 어떻게 객관적인 지식이 될 수 있는가? 임계훈(2001)은 ‘소집단 협력 학습’을 통한 동료 학습자들 간의 상호작용을 객관적 지식 창조의 사회적 과정의 준거로 분석하였다[3]. 협력 학습 과정에서 학생들 간의 단방향적인

지식 전달 과정이 교사의 지식 전달과 동일하게 이루어진다는 점, 개인의 주관적 지식을 공개적으로 평가할 수 없다는 점 등을 들어 사회적 상호작용의 근거라고 볼 수 없다고 하였다. 그러나 현재의 정보통신기술은 학생들도 전문가와 같은 방법과 절차로 지식을 생산하고 평가할 수 있는 온라인 학습 환경을 제공한다. 학생들은 교실이나 온라인 환경에서 교사, 동료 학습자, 전문가 등과 풍부한 상호작용을 통하여 지식의 생성 과정에 참여할 수 있다.

한편, 김영전(2020)은 의학 연구에서 임상실습 중의 학생들의 대화와 담론을 통하여 사회적 상호작용의 과정을 분석하였다. 학생들은 상호작용을 통한 학습에 대하여 ‘의미 있는 사건’, ‘공유의 이유’, ‘학습 논점’, ‘학습의 참여’의 활동을 하고 있다고 밝혔다[4]. 마찬가지로 코딩 과정은 학생들 간의 다양한 대화와 담론을 형성한다. 가장 대표적인 사례로는 프로젝트와 리믹스(Remix), 디버깅(Debugging) 등과 같은 활동이다[9]. 스크래치 웹 사이트에는 학생들이 온라인을 통하여 다른 사람들과 프로젝트 팀을 구성할 수 있으며 자신이 발견하거나 생산한 지식에 대한 사회적 공유와 평가에 참여할 수 있게 되었다.

두 번째로 ‘사회의 객관적인 지식이 개인의 주관적인 지식이 되는 과정’은 학교에서 일어나는 대부분의 교육 활동이 해당될 것이다. 페퍼트는 사회의 객관적 지식이 개인의 주관적인 지식으로 어떻게 치환되는지를 설명하지는 않았다. 다만, 그는 컴퓨터가 어린이들을 프로그래밍한다고 비판하면서 “내 비전에서는 어린이가 컴퓨터를 프로그래밍한다. 어린이는 컴퓨터를 프로그래밍하면서 가장 현대적이고 가장 강력한 기술에 대한 감각을 익힐 뿐 아니라 과학과 수학, 지적인 모형 구성을 통해 얻은 심도 있는 개념과 친밀해진다”고 말한다. 그는 ‘생각하게 하는 사물(objects to think with)’[1][17]을 통하여 객관적인 지식이 더욱 쉽게 주관적인 지식이 된다는 점을 강조하였다. 현재의 시점에서 ‘생각하게 하는 사물’은 교육용 로봇이나 피지컬 컴퓨팅 도구, EPL 언어 등이 이에 해당된다.

2010년 미국 교육부는 Webb’s DOK모델[5]을 참고하여 SBAC(Smarter Balanced Assessment Consortium)를 만들고 각 주별로 공통핵심기준에 대한 기대치를 내고 있는지 평가 및 검증할 수 있도록 하였다[16].

Webb’s DOK는 2018 국제학업성취도 평가 과학 프레임 워크에도 반영되어 학습자가 고차원적 인지수준의 능력을 발휘하는지를 평가할 수 있도록 한 것이다. Heidi Williams(2017)는 4수준의 단계들이 서로 협력적으로 학습을 돕기 때문에 컴퓨팅 사고와 컴퓨터 과학 분야에 적합한 모델이라고 하였다.

<Table 1> Webb’s Depth of Knowledge(DOK) Levels

Level	Summary	Thinking Involved
DOK 1	Recall & Reproduction	- What is the answer? - What is the outcome/result? - How did this outcome happen?
DOK 2	Basic Application of Skills and Concepts	- How can we reach the answer? - How do you produce this result? - How is this concept used? - How should we find a solution and why?
DOK 3	Strategic Thinking	- Which approach leads to a better outcome? - Why do things work the way they do?
DOK 4	Extended Thinking	- What can you design/create? - What is the connection between different concepts? - What are the possibilities for using this?

페퍼트의 사회적 구성주의 관점을 바탕으로 Webb’s DOK 모델을 바라보면, 학교는 사회의 객관적 지식을 ‘생각하게 하는 사물’을 통하여 더욱 쉽게 구성할 수 있는지에 더욱 관심을 가져야 한다. 학교는 컴퓨팅이나 컴퓨터과학의 객관적 지식이 개인의 주관적인 지식으로 되는 과정에서 ‘생각하게 하는 사물’을 적극 활용하고 상호 협력하는 문화를 만들어야 한다. 코딩 관련 지식은 학교 교실에서 교사와 학생간의 상호작용을 통하여 인 증받기도 하지만, 엔트리, 스크래치와 같은 플랫폼은 온라인에서의 상호작용을 통하여 불특정 다수로부터 인증 받는 과정을 거친다. 그러므로 코딩 교육에서 ‘사회적 구성’은 새로운 아이디어를 생성할 때, 반대로 사회적 지식을 개인적으로 내면화하는 과정에서도 ‘사회적 대화와 피드백’을 통하여 이루어진다[4]고 이해할 수 있다.

3. 사회적 구성주의 전략의 분석

3.1 사회적 구성주의 접근 모형

사회적 구성주의는 코딩 교육뿐만 아니라 컴퓨터과학의 커다란 담론이기도 하다. 스크래치의 개발자 Mitchel Resnick(2017)의 저술 ‘평생 유치원’의 한국어판 서문에서 그는 다음과 같이 말하였다[6].

“아침에 일어나 스크래치 웹사이트에 접속했을 때, 내가 본 것은 30개의 동일한 프로젝트가 거의 동시에 등록되어 있는 화면이었다. (중략) 자세히 보니 한국의 특정 지역에서 아이들 30명이 따로따로 등록한 프로젝트였다. 그 프로젝트는 한국의 한 교실에서 선생님이 학생들에게 특정 프로젝트를 어떻게 만들어내는지 단계별로 가르친 결과물이었다.”

이와 같은 코딩 사례는 엔트리와 스크래치 웹사이트에서 쉽게 발견할 수 있다. 대부분 선생님의 시범을 따라한 결과로서 나타난 것이다. 코딩 교육에서의 이와 같은 접근 방식은 시모어 페퍼트와 같은 진보주의 학자들이 구상했던 개인의 지식 구성과는 거리가 먼 활동이라고 생각한다.

사회적 구성주의 접근에서 학습 경험을 어떻게 설계할 것인가에 대하여 MIT 미디어 랩의 평생유치원 그룹은 4가지의 핵심 요소를 제안하였으며 ‘창의적 학습을 위한 4P’라고 하였다[8]. 그것은 프로젝트(Project), 동료(Peer), 열정(Passion), 놀이(Play)이다. 첫째, 사람들은 개인적으로 의미 있는 프로젝트에 능동적으로 참여하는 활동을 통하여 가장 잘 배운다. 새로운 아이디어를 도출하기, 프로토타입을 설계하기, 정교화하기 등이 이에 해당한다. 둘째, 사회적 활동으로서 학습을 확산하는 것으로 사람들과 아이디어를 공유하고 프로젝트를 함께하고, 서로의 일을 돕는 것 등이다. 셋째, 사람들은 자신이 관심이 있는 프로젝트에 열정을 가지고 참여할 때 도전적인 과제에 오랫동안 열심히 지속하게 되며 그 과정에서 더 많은 것을 배우게 된다. 넷째, 학습은 거의 대부분 놀이로 구성된다. 새로운 것에 도전하고, 재료를 개조하고 위험을 감수하며 끊임없이 반복한다.

코딩을 어떻게 가르칠 것인지에 대하여 국가, 기관, 학자들의 접근 방식 5가지 모델을 분석하였다. 영국의 CAS[21]와 Code.org[22], HGSE[23]등과 같은 기관의 가이드라인과 학자들의 구성주의적 접근 방법 모델[9[24]를 선정하였다.

<Table 2> Computational Thinking Approaches

Models	Barefoot	Connected Code	A Framework for K-12 CSE	Positive Technological Development	creative computing
Author	Computing at school	Kafai &Burke	acm & code.org csteaxher.org	Marina Umaschi Bers	HGSE
year	2014	2014	2016	2017	2017
approaches	Tinkering experimenting &Playing	From Code to Application	Fostering an Inclusive culture	Communication	Creating
	Creating designing &making	From Tools to communities	Creating computational artifacts	Collaboration	Personalizing
	Debugging finding &fixing errors	From Scratch to Remix	Collaborating around computing	Community Building	Sharing
	Persevering keeping going	From Screen to Tangible	Testing and Refining Computational Artifacts	Content Creation	Reflecting
	Collaborating working together		Recognizing and defining Computational Problems	Creativity	
			Communicating about computing	Choice of Conduct	
			Developing and Using Abstractions		

3.2 사회적 구성주의 학습 전략

3.2.1 만들기(Making)

사회적 구성주의 접근은 학생들에게 코딩 교육에서 응용 프로그램을 개발하거나 사물(artifacts)을 만드는 학습으로의 전환을 강조한다. 위의 <표 2>에서 제시한 사회적 구성주의 관점의 접근 방법에는 ‘Creating designing & making’, ‘From Code to Application’, ‘Creating computational artifacts’, ‘Content Creation’,

‘Creating’등이 있다.

전통적으로 프로그래밍을 학습하는 방법은 순차, 반복문, 조건문, 변수 등에 대한 강의와 연습 등이다. 이와 같은 형태의 학습은 모든 학생들이 동일한 결과물을 생산하게 되는 획일적인 수업이 되기 쉽다. 반면 사회적 구성주의 학습은 자신만의 응용 프로그램이나 사물을 설계하고 프로그래밍 하는 과정에서 지식이 구성된다고 본다. 이 과정에서 학습자는 컴퓨터를 프로그래밍 하는 튜터(Tutee)[1]로서의 역할을 수행하고 자신의 작품에 대한 동일시와 정서적인 애착을 갖게 된다[9][17].

페퍼트는 피아제와 차별화하기 위하여 ‘생각하게 하는 사물’이라는 개념을 제시하였다. 이것은 프로그램과 로봇, 게임 등과 같이 물리적인 세계와 디지털 세계에 존재하면서 기존의 지식과 새로운 지식을 연결하고 검토하며 수정하는데 사용된다. 이것은 과거에 컴퓨터와 LOGO 프로그램이었으며 현재는 게임이나 로봇, 스토리텔링 등이 될 수 있다. 박주연(2019)은 학습자의 수준에 따라 흥미가 높은 애니메이션, 미술과 음악 프로젝트, 이야기, 그리고 게임 유형의 프로젝트 순의 접근이 학생들의 사고력에 차이가 나타남을 확인하였다[18]. 더 나아가서, 학습자는 스스로 게임을 개발하거나 프로그램을 설계함으로써 개인적으로 지식을 생성하고 이를 다른 사람과 공유하는 사회적 구성의 경험을 할 수 있다[9][17]. 인터넷을 통한 공유는 자신의 ‘생각하게 하는 사물’ 개발 과정에 사용했던 컴퓨팅 사고를 온라인에서의 참여로 확대될 수 있다.

3.2.2 텅커링(Tinkering)하기

텅커링은 학습자의 개인적인 관심과 경험 그리고 다양한 학습 성과 사이에 연결 지점을 형성한다. 텅커링은 학생들이 무엇을 어떻게 해야 할 지 정확히 모르는 상태에서 직관, 상상, 호기심 등에 따라 학습할 때 일어난다. 이는 놀이와 학습을 통합한 활동으로 질문, 탐색, 반복을 통한 발견의 즐거움을 통한 학습을 말한다[7]. 위의 <표 2>에서 텅커링과 관련된 접근 방법으로 ‘Tinkering experimenting & Playing’, ‘From Scratch to Remix’, ‘Fostering an Inclusive culture’, ‘Testing and Refining Computational Artifacts’, ‘Choice of Conduct’, ‘Personalizing’ 등이 있다.

터클과 페퍼트는 학습 환경에서 다양한 ‘앎의 방식’에 대하여 플래너(Planner)와 브리콜레르(Bricoleurs)로 구분하였다[8]. 플래너는 코딩할 때 전통적이고 논리적인 하향식(Top-down) 모델을 사용하는 사람을 말한다. 반면 브리콜레르는 구체적이고 개인적인 접근을 사용하며 자신만의 프로젝트를 상향식(bottom up)으로 만드는 사람을 말한다. 도구로서의 컴퓨터에 대한 접근은 사고와 일의 방식에 있어서 플래너의 방식에 중점을 둔다. 반면 텅커링할 수 있는 학습 환경은 브리콜레르를 위한 작업 공간을 만드는 것뿐만 아니라 소외되는 학생이 없도록 정의적인 영역을 포함한다[1].

학교에서의 코딩 활동이 교사가 주도하는 방식으로 이루어지면 학생들은 ‘플래너’의 역할에 집중하게 되어 매 시간마다 동일한 프로젝트가 만들어질 것이다. 반면 브리콜레르와 같이 되기 위해서는 교사가 학생들에게 실험을 하고 위기를 감수하고, 자신들의 아이디어를 가지고 놀 수 있도록 허용해 주어야 한다. 터클은 텅커링 접근 방식을 ‘소프트 마스터리(soft mastery)’와 ‘하드 마스터리(hard mastery)’로 비유하였다[1][25]. 전자는 학생들의 다양한 방식과 생각 대하여 동등한 가치를 부여하는 것이며 후자는 단계별 문제해결, 플로우차트에 의한 관리, 분석적 접근법 등을 의미한다.

그러므로 코딩 교육에서 학생들은 학습의 주체로서 자신들의 스타일대로 학습에 참여하는 브리콜레르가 되어야 하고 텅커링에 있어서 소프트 마스터리의 접근을 하여야 한다.

3.2.3 협력과 공유하기

사회적 구성주의에서 협력과 공유는 필수적이다. 코딩은 협력과 공유를 반드시 필요로 하는 탐구 중심의 학문이다. 컴퓨터과학은 서로 알고 있는 것을 공유하고 다른 사람에게 해결방법을 설명하면서 학습하는 것이다[1]. 위의 <표 2>의 접근 방법에서 공유와 협력에 대한 아이디어는 ‘Collaborating working together’, ‘From Tools to communities’, ‘From scratch to Remix’, ‘Collaboration around computing’, ‘Collaboration’, ‘Sharing’ 등이다.

온라인을 통한 협력과 공유는 웹 2.0의 철학을 반영한다. 스크래치와 엔트리, Hour of Code 등과 같은 교육

플랫폼은 학습 과정 및 결과의 공개, 공유와 참여가 가능한 웹 2.0 플랫폼이자 학생들의 놀이터이다. 학생들은 코딩을 배우기 위하여 플랫폼에 접속하는 것이 아니라 놀기 위하여 접속한다. 플랫폼은 프로그래밍 할 수 있는 도구를 제공하고 공동 관심사를 가진 학생들이 함께 프로젝트를 수행할 수 있도록 지원한다. 전통적인 교실에서 교사가 학생에게 교과를 가르치듯이 코딩을 가르치는 것을 지양하여야 한다. 개인적인 문제 해결을 위한 프로그래밍은 학교와 온라인 커뮤니티로 확대되어야 한다. 또한 학생들의 실제적 경험이 프로그래밍 도구와 연결되어야 한다.

작 프로그래밍은 코딩을 위한 매우 훌륭한 방법이다. 두 학생이 한 대의 컴퓨터를 이용하여 운전자(driver)와 네비게이터(navigator)의 역할을 수행한다. 운전자는 키보드와 마우스를 이용하여 타이핑을 하거나 코드를 작성한다. 네비게이터는 프로젝트의 큰 그림에 집중하고 작성된 코드가 논리적인지를 확인한다. 학생들은 자신의 생각을 소리내어 말하고 상호작용하며 함께 문제를 해결한다. 이 방법은 혼자서 코딩하는 방식에 비하여 동일 작업시간에 훨씬 더 좋은 코드를 작성할 수 있다[10]. 최형신 외(2020)는 초등학교 4학년 대상 로봇 기반 SW융합교육에서 4인 1팀 학생들의 협업능력 및 의사소통능력에 향상을 가져온다고 하였다[13].

미첼 레스닉과 브라인언 실버맨(2017)은 학습자들의 공동체 참여를 위한 비유를 제시하였다[6]. 낮은 문턱(low floor), 코딩 도구는 새로운 사용자의 자신감에 따라 서서히 도구를 익힐 수 있을 만큼 쉬워야 한다. 높은 천장(high ceiling), 코딩 도구는 경험이 풍부한 사용자가 도구를 능숙하게 사용할수록 더욱 어렵고 복잡한 애플리케이션을 제작할 수 있도록 해야 한다. 넓은 벽(wide wall), 코딩 도구는 개인의 관심을 반영할 수 있는 다양한 프로젝트를 제작할 수 있는 환경을 제공해야 한다.

3.2.4 실제적 객체의 활용

구성주의의 혁신적인 접근 방법으로서 손으로 만질 수 있는 장난감, 로봇, 피지컬 컴퓨팅 재료 등을 실제적 객체라고 한다. 스크래치와 엔트리 등과 같은 블록 프로그래밍은 화면 속에 있는 추상화된 객체(고양이나 엔트

리 붓)를 구체화하며 코딩은 추상화된 객체를 구체화한다. 로봇이나 피지컬 컴퓨팅을 위한 도구들은 학생들이 손으로 만질 수 있는 것으로서 직접적인 체험을 제공한다. 로봇은 화면 속의 객체가 실제의 공간에서 움직이는 구체적 실체로 나타난 것이다. Kafai와 Burk(2014)는 페퍼트의 신체적 동조성의 교육적 요소를 다음과 같이 설명하였다[9].

“이는 (바닥에서 혹은 화면에서 거북이가 되어서 하는) 표현과 (회로를 조작하는) 손길과 (물건의 기능을 이해하고 물건과 개인적으로 연결 고리를 만드는) 교육에 관한 문제였다”.

위의 <표 2>의 접근 방식에서 이와 같은 관점은 ‘Creating designing & making’, ‘From Screen to Tangible’, ‘Testing and Refining Computational Artifacts’, ‘Creativity’, ‘Creating’ 등이다.

코딩 교육을 위한 실제적 객체를 활용한 교육을 피지컬 컴퓨팅이라고 한다. 피지컬 컴퓨팅은 학교와 사회에서 매우 빠르게 확산되고 있다. 교사들의 테크놀로지 활용에 대한 긍정적 태도와 교육적 신념은 로봇활용교육에 있어 구성주의적인 학습자 주도의 교수·학습 활동을 강화한다[11][12]. 특히, 학교의 분위기와 테크놀로지 활용과 관련된 연수는 로봇활용교육 활성화에 영향을 미치는 중요한 요인으로 나타났다[12]. 서영민과 이영준(2010)은 로봇을 활용한 교육이 상호작용적이고 놀이중심의 체험적이고 협력중심의 학습 환경을 제공한다고 하였다[14].

피지컬 컴퓨팅은 두뇌와 신체적 활동을 디지털 및 물리적 재료를 통합함으로써 어린이들이 단순히 지식을 소비하는 사람이 아니라 적극적으로 지식을 만드는 사람으로 성장시킨다. 홍지연 외(2019)는 로봇을 활용한 언플러그드 활동을 통하여 컴퓨팅 환경 없이도 센서를 활용해 물리적 세계와 상호작용할 수 있는 코딩 교육이 가능함을 확인할 수 있었으며 학교의 환경적 요인에 구애받지 않고 로봇 활용 코딩 교육을 일반화할 수 있을 것이라고 하였다[19]. 즉, 컴퓨터를 사용하는 새로운 방식은 미래의 디지털 세계에서 학습이 어떻게 변화하게 될 것인지에 대한 창조적 경험을 제공하게 될 것이다.

3.2.5 스캐폴딩과 학습의 주도성

스캐폴딩은 학습자들이 스스로 도달하기 어려운 목표

를 타인의 도움을 통하여 성취하도록 도와주는 학습 전략이다. 이것은 학습자가 점차 독립적으로 학습을 진행하는 것이 가능해짐에 따라 점진적으로 감소하여 소거된다. 스캐폴딩은 Vigotsky의 ZPD(Zone of Personal Development)개념을 설명하는 것으로 코딩 교육에서도 학생들의 목표를 성취하기 위하여 교수자 또는 동료와 상호작용을 통해 학습자에게 제공되는 것으로 다양한 도구, 전략, 안내, 자원 등을 포함한다[15]. Moran Tsur(2017)는 지식의 사회적 구성을 위한 전략으로 스캐폴딩과 학습자의 주도성을 제안하였다[8]. 스캐폴딩은 ‘채널링과 집중(Channeling and focusing)’, ‘모델링(modeling)의 접근 방식을 취한다[8]. 채널링은 학생들의 과제에 대한 자유도를 낮추는 것을 말하고 집중은 특정 기능에 대한 학습자의 관심을 집중시키는 것이다. 모델링은 과제에 대한 더욱 발전된 해결책을 얻기 위하여 전문가의 능력을 배우는 것이다.

스캐폴딩 과정에서 교사는 촉진자로서 학생들에게 문제 해결을 위한 조언, 도구 및 재료 등을 제공하여야 한다. 예를 들면, 스크래치의 리믹스(remix)는 스캐폴딩의 좋은 접근 방식이다. 이것은 채널링과 집중 전략을 포함한다. 학습자는 코드의 일부를 수정함으로써 목표에 달성할 수도 있고 전문가를 모델링함으로써 새로운 창작으로 나아갈 수도 있다.

학습자 주도성을 지원하는 것은 사회적 구성주의의 중요한 가치이다. 코딩 교육에서 누가 학습의 주도권을 가질 것인가의 문제는 매우 중요하다. 코딩 교육에서 ‘Hour of Code’와 같이 퀴즈 형태의 프로그램은 여전히 주도권이 컴퓨터에 있다. 반면 스크래치와 엔트리 등과 같은 학습 환경은 학생들이 스스로 학습 과정을 결정할 수 있는 권한을 부여한다. 페퍼트는 어린이들의 생각 표현 능력 향상을 위하여 학습자의 주도성을 ‘놀이터(Playground)’와 ‘울타리(Playpen)’로 비유하였다[6]. 프로그래밍 언어는 학생들의 생각을 표현하기 위한 ‘놀이터’가 될 수 있음에도, 교사들이 전통적인 방식으로 프로그래밍을 가르칠 경우 담장에 둘러싸인 ‘울타리’안에 있게 될 것이라고 하였다.

‘울타리’는 주로 주어진 문제를 해결하는 학습 환경을 제시한다. Hour of Code의 과제, 엔트리의 학습 단계(따라하기 → 도전하기 → 퀴즈풀기 → 자유롭게 만들기), CodeMonkey, codeSpark, Academy 등과 같은 서비스

는 퀴즈 중심의 환경을 제공한다. 반면, ‘놀이터’는 프로젝트가 가능한 개방된 학습 환경을 제공한다. 스크래치 플랫폼, 엔트리의 ‘작품 만들기’, 마인크래프트, ‘코두’ 게임 등과 같은 환경은 학생들이 제약 없이 자신의 프로젝트를 만들 수 있도록 개방되어 있다. 구성주의적 관점에서 ‘운동장’의 비유는 학생들이 코딩을 통하여 자신의 생각을 표현하고 학습하며 자신들이 하고 있는 일에 대한 메타학습 환경을 의미한다.

4. 코딩 교육에의 시사점

본 논문은 사회적 구성주의 관점에서 코딩 교육의 접근 방식을 분석하고 다음과 같은 시사점을 얻을 수 있었다.

첫째, 코딩 교육에 있어 사회적 구성주의적 접근의 가치를 확인할 수 있었다. 지식 구성자로서 어린이의 학습에 대한 신념, 페퍼트의 ‘생각하게 하는 사물’을 활용한 프로젝트, 브리콜레르로서의 문제 해결자, 짝프로그래밍을 통한 상호작용, 실제적 객체를 통한 실제적 문제 해결 등은 구성주의적 접근 방법을 가장 잘 드러낸다.

둘째, 만들기로 시작하는 코딩 교육이어야 한다. 코딩과 관련한 지식이나 방법에 대한 교육보다 컴퓨팅 사고력을 증진하는 교육이어야 한다. 학생들은 코딩을 배울 때, 창의적인 무엇인가를 만드는 것으로 시작하여야 한다. 블록 프로그래밍의 프로젝트, 로봇이나 피지컬 컴퓨팅 도구를 활용한 만들기가 중심이 되어야 한다. 이를 위해서는 프로젝트학습, 문제중심학습, 열린 과제 중심의 학습 전략을 이용한 텅커링, 만들기, 리믹스 등의 활동이 가능하다.

셋째, 학습자의 주도성과 교사의 지원 사이의 균형을 고려하여야 한다. 이를 위하여 학생들이 자신에게 의미 있는 선택을 할 수 있도록 스캐폴딩이 중요하다. 학생들은 코딩 과정에서 많은 선택을 하게 된다. 프로젝트의 주제와 해결 과정에 대한 의사 결정의 주체가 되어야 한다. 코딩 블록, 배경, 캐릭터, 움직임 뿐만 아니라 문제 해결 과정에 대한 선택권도 주어져야 한다. 이 과정에서 학생들은 자신의 작품에 대한 감정 이입과 애착을 가지게 된다.

넷째, 촉진자로서의 교사는 학생의 의사 결정을 존중하여야 한다. 플랫폼에서의 선택권을 주어야 한다. 문제

해결에 필요한 시간을 스스로 정하도록 한다. 텅커링 과정에 대한 신뢰를 주어야 한다. 학생들은 항상 창의적인 것을 만들 것이다. 답을 주기보다는 질문을 하여야 한다. 그리고 자신의 작품을 다른 학생들과 공유할 수 있도록 도와주어야 한다.

5. 결론

본 논문은 코딩 교육에서 사회적 구성주의의 접근 방식을 탐구하고 시사점을 얻고자 하였다. 학교에서의 코딩 교육에 대한 접근은 매우 다양한 방법으로 나타난다. 언플러그드 컴퓨팅, 엔트리와 스크래치를 이용한 블록 코딩, 마인크래프트와 같은 게임을 활용한 코딩, 로봇이나 피지컬 도구를 활용한 컴퓨팅 등 그 범위와 종류도 다양하다. 이와 같은 코딩 교육의 실제적 장면에서 어떻게 가르칠 것인가는 항상 중요한 과제이다. 본 연구는 코딩 교육을 위한 사회적 구성주의 관점에서의 접근 방법을 분석하고 시사점을 제시하였다.

일반적으로 사회적 구성주의는 학습자의 경험을 중시하며 학습자를 둘러싼 환경과의 상호작용을 통하여 스스로 지식을 구성한다는 입장을 취한다. 특히, 페퍼트의 사회적 구성주의 관점을 분석하고 학습을 위한 ‘생각하게 하는 사물’의 비유를 이해하고자 하였다. ‘사회적 구성’의 의미 분석을 통하여 개인의 객관적 지식이 사회적 지식으로 생성되거나 그 반대의 경우에 대한 의미를 탐구하였다. 대부분의 학교 지식은 기존의 객관적인 지식을 학생들이 내면화하는 과정이며 코딩을 위한 자료는 학습을 더욱 쉽게 하는 매개로서의 역할을 함을 설명하였다. 사회적 구성주의 학습 전략의 접근 모형으로 5가지의 프레임워크를 분석하고 이를 바탕으로 사회적 구성주의 학습 전략으로서 만들기, 텅커링하기, 협력과 공유하기, 스캐폴딩과 학습의 주도성, 실제적 객체의 활용 등을 탐구하였다. 이를 통하여 코딩 교육을 위한 사회적 구성주의 접근의 가치와 만들기로 시작하는 코딩, 학습자의 주도성을 지원하는 스캐폴딩, 촉진자로서의 교사의 역할을 제시하였다.

참고문헌

- [1] Sylvia Libow Martinez & Gary S. Stager (2019). *Invent to Learn: Making, Tinkering, and Engineering in the classroom* 2nd Edition. Modern Knowledge Press.
- [2] Hong, Jin-Kon (2002). A Study on the Meaning of ‘Social Construction’ in Mathematics Education. *The Mathematical Education*. 41(3), 6-34.
- [3] Jae Hoon, Lim (2001). A study on the epistemology of mathematics education. *Journal of Educational Research in Mathematics*, 11(2), 291-305.
- [4] Young Jon Kim (2020). Application of Social Constructivism in Medical Education. *Korean Medical Education Review*, 22(2), 85-92.
- [5] Webb, Norman L., Depth-of-Knowledge Levels for Four Content Areas(March 28, 2002).
- [6] Mitchel Resnick (2018). *Lifelong Kindergarten: Cultivating Creativity through projects. Passion, Peers, and Play*. Smilegate Foundation.
- [7] Kim, Hye Jeong, Suh, Hee Jeon (2020). A Study on the Principles of Makerspace Design for Early Childhood Education: Focusing on PTD and Affordance. *The Korea Society for Children’s Media*, 19(1). 203-230.
- [8] Moran Tsur (2017). *Scratch Microworlds: Introducing Novices to Scratch Using an Interest-Based, Open-Ended, Scaffolded Experience*. Master of Science at MIT.
- [9] Yasmin B. Kafai & Quinn Burke(2014). *Connected Code*. MIT.
- [10] Jane Krauss and Kiki Prottzman(2017). *Computational Thinking and Coding for Every Student*. Corwin.
- [11] Seongg-Won Kim, Youngjun Lee (2016), The Effects of Robot Programming on the Attitudes toward Robot of Pre-service Teachers. *The Journal of Korean Association of computer Education*, 19(6), pp.91-103
- [12] Won Sug Shin (2020). An Analysis of Teachers’

- TPACK on Robotics in Education. *Korea Academy Industrial cooperation Society*, 21(6), 196-203.
- [13] Hyungshin Choi, Jeongmin Lee (2020). The Effects of Educational Robot-based SW Convergence Education on Primary Students' Computational Thinking, Collaborative and Communication Skills. *Journal of The Korean Association of Information Education*, 24(2), 131-138
- [14] Seo, Y., & Lee, Y. (2010). A subject integration robot programming instruction model to enhance the creativity of information gifted students. *The Journal of Korean Association of Computer Education*, 13(1), 19-26
- [15] Joo Yeun Kim, Hyosun Park, Kyu Yon Lim (2015). The effects of scaffolding types on problem solving ability and achievement in problem solving learning with creative thinking method. *The Journal of Educational Information and Media*, 21(1), pp.111-136.
- [16] Heidi Williams (2017). No Fear Coding. International Society for Technology in Education.
- [17] Seymour Papert(1991). *Mindstorms: Children, Computers, And Powerful Ideas*. Basic Books. Inc. 2nd Edition.
- [18] Juyeon Park (2019). Evaluation of Computational Thinking through Code Analysis of Elementary School Students' Scratch Projects. *Journal of The Korean Association of Information Education* 23(3), 207-217.
- [19] Jiyeon Hong, Yungsik Kim(2019). Development and Application of Unplugged Activity-centered Robot for Improving Creative Problem Solving Ability. *Journal of The Korean Association of Information Education*, 23(5), 441-449.
- [20] Karen A. Brennan.(2013). *Best of Both Worlds: Issues of Structure and Agency in computational Creation, in and out of School*. MIT.
- [21] Barefoot, C. A. S. (2014). Computational Thinking. Retrieved from <http://barefootcas.org.uk/barefootprimary-comp-uting-resources/concepts/computational-thinking>
- [22] K - 12 Computer Science Framework. (2016). Retrieved from <http://www.k12cs.org>.
- [23] Harvard Graduate School of Education. Creative Compting curriculum. <http://creativecomputing.gse.harvard.edu/guide/curriculum.html>.
- [24] Marina Umaschi Bers(2017). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. Routledge.
- [25] Sherry Turkle(2005). *The Second Self: Computers and the Human Spirit*. MIT Press.

저자소개

김 홍 래



1989 춘천교육대학교

(교육학학사)

1995 한국교원대학교

컴퓨터교육과(교육학석사)

1999 한국교원대학교

컴퓨터교육과(교육학박사)

2021년 현재, 춘천교육대학교 컴퓨터교육과 교수
 관심분야 : 컴퓨터 교육과정, 교육 정책, 교과교육
 E-mail : saerom@cnu.ac.kr