

초등학교 영어교과를 적용한 프로그래밍 교육 모델 개발

허미연* · 김갑수**

서울상도초등학교* · 서울교육대학교**

요 약

그동안의 소프트웨어교육과 타 교과와의 연계·융합에 대한 연구는 주로 수학과 과학교과에 편중되어왔다. 이는 학생의 교과에 대한 다양한 선호와 학습 성격 유형 등을 만족시키지 못하여 학습 격차를 유발할 수 있다. 뿐만 아니라, 컴퓨팅 사고를 적용할 수 있는 다양한 융·복합적 문제의 해결과정을 다루어야 함을 감안할 때 바람직하지 않다. 그리하여 기존의 수학과학적 접근에서 벗어난 언어적인 접근인 영어교과와의 연계를 통해 학생들의 다양한 성향과 선호를 포용하고, 영어교과와 소프트웨어교육의 새로운 언어를 배우는 과정과 방법상의 유사점을 접목시켜 교육 효과의 향상을 도모하고자 하였다. 이를 위하여 초등 영어교과와 소프트웨어교육의 교수학습모델 분석을 토대로 연계에 적합하도록 기존의 영어교과와 소프트웨어 교수학습모델을 변형하여 수업모형을 개발하였다. 이후 초등학교 영어교과내용 중 소프트웨어교육에 적용 가능한 학습 요소를 추출하여 개발된 수업모형에 적용한 프로그램을 설계하여 실제적인 학습 활용 방안을 모색하였다.

키워드 : 스크래치, 알고리즘, 영어교과, 프로그래밍, 컴퓨팅 사고력

A Study on the Development of Programming Education Model Applying English Subject in Elementary School

Miyun Heo* · Kapsu Kim**

Seoul Sangdo Elementary School* · Seoul National University of Education**

ABSTRACT

Research on software education and linking and convergence of other subjects has been mainly focused on mathematics and science subjects. The dissatisfaction of various preferences and types of learning personality cause to learning gap. In addition, it is not desirable considering the solution of various fusion problems that can apply the computational thinking. In this way, it is possible to embrace the diverse tendencies and preferences of students through the linkage with the English subject, which is a linguistic approach that deviates from the existing mathematical and scientific approach. By combining similarities in the process of learning a new language of English education and software education. For this purpose, based on the analysis of teaching - learning model of elementary English subject and software education, we developed a class model by modifying existing English subject and software teaching - learning model to be suitable for linkage. Then, the learning elements applicable to software education were extracted from the contents of elementary school English curriculum, and a program applied to the developed classroom model was designed and the practical application method of learning was searched.

Keywords : Scratch, Algorithm, English Subject, Programing, Computational Thinking

본 논문은 제 1저자의 석사 학위 논문을 수정 재구성하였음.

교신저자 : 김갑수(서울교육대학교)

논문투고 : 2017-07-30

논문심사 : 2017-08-07

심사완료 : 2017-10-18

1. 서론

지식정보사회의 흐름에서 살아남기 위해서 전 세계적으로 장기적인 거시적 안목을 지닌 창의적 컴퓨팅적 사고를 지닌 인재를 양성하는 교육의 중요성을 강조하고 있다[12]. 미국 등 다른 나라들은 이미 정부주도하에 아동 소프트웨어 교육을 필수적으로 실시하고 있다[1][2]. 우리나라도 중요성을 인지, 2015개정교육과정부터 중·고등학교에서는 정규교육과정에 편성·실시하고 있으며, 초등학교에서는 2019년부터 실시 예정이다. 그러나 편성 내용이 초등 5,6학년을 대상으로 1년에 17차시만 편성되어 있어 편성 시간만으로 실질적인 컴퓨팅 사고력의 신장을 기대하기는 어렵다[6][12][13]. 특히, 초등학교 시기는 고등 사고력의 기초를 형성하는 시기이므로 이 시기에 적절한 교육을 실시하여야 소프트웨어교육에 대한 이해도가 높아짐은 물론, 소프트웨어에 대한 관심과 흥미를 유발하여 이후 진로 선택에 많은 영향을 미칠 수 있다. 이는 곧 미래의 국가경쟁력과 직결될 것이다[7][11]. 이런 이유로 영국, 인도, 핀란드, 미국 등에서는 초등학교 저학년부터 소프트웨어교육을 강조하여 실시 중이거나 권장하는 반면[3][15], 우리나라는 초등학교 소프트웨어교육 편성 비중이 상대적으로 낮아 수업시수가 충분히 확보되어 있지 않다. 이 문제의 해결을 위해 소프트웨어교육을 타 교과와 연계·융합하여 실시하여 교과시간을 확보하고, 컴퓨터 외 타 교과에 흥미를 느끼고 익숙한 학생들을 학습에 자연스럽게 유인하고 각 교과의 학습간의 전이효과도 기대할 수 있을 것이다. 이런 기대효과로 최근까지 다양한 연계방안이 연구되어 왔는데, 연구의 내용이 주로 수학과 과학교과와의 연계에 편중되어왔다[9][10][16]. 이는 초등학생의 교과에 대한 다양한 선호와 학습 성격 유형 등을 만족시키지 못하여 학습의 격차를 야기할 수 있다. 또한, 소프트웨어교육이 실생활의 다양한 융·복합적 문제를 컴퓨팅적 사고를 적용해 해결하는 것이 목적임을 감안할 때 바람직하지 않다. 따라서 본 연구를 통해 프로그래밍을 기존의 수·과학적인 측면이 아닌 언어적인 측면을 통해 접근한 초등학교 영어교과 기반의 소프트웨어수업모형을 개발하고, 실제 교육프로그램을 설계하여 효과를 검증하여 초등학교 소프트웨어교육의 활성화에 도움이 되고자 한다.

2. 관련 연구

2.1 소프트웨어 수업 모델

소프트웨어교육의 수업모델은 전통적인 교수학습모델을 학생들의 컴퓨팅 사고력의 신장에 학습 초점을 맞춰 변형·적용하여 개발되었다. 김진숙 외 연구진이 2015년 소프트웨어교육 교수학습 모형 개발 연구를 통해 5가지 중점 소프트웨어교육 교수학습 모델을 정하고, 각 모델에 대한 특징을 다음과 같이 정리하였다[9].

2.1.1 시연중심모델

직접교수법을 응용하여 가르치려는 컴퓨팅적 사고 과정을 교사가 설명을 하거나 시범을 통해 소개하고, 이 내용을 학생들이 그대로 따라하며 이 과정에서 학습자는 다양한 질문과 답변을 통해, 과정을 이해하게 된다. 이후 배운 내용을 토대로 학생들이 직접 만들어 보는 과정을 거치며 소프트웨어개발 제작 역량을 키우게 된다. 교수학습 과정에서 교사는 적합한 시연자료의 제작에서부터, 적절한 발문을 통해 학습자가 모방과정에서 자연스럽게 컴퓨팅 사고력을 이해하고 습득하도록 이끌어주는 전체적인 조율을 해야 한다.

2.1.2 재구성중심모델

발견학습법에서 응용된 모델로 다양한 사례를 중심으로 개념과 원리를 발견하고, 제시된 사례의 수정과 재구성을 통하여 컴퓨팅적 사고를 이끄는 모델이다. 컴퓨팅 사고력이 담긴 프로젝트를 경험하는 과정에서 자연스럽게 내포된 알고리즘을 발견하고, 교사는 이 모듈 및 패키지들을 의도적으로 변형하여 제시하여 학생으로 하여금 수정하도록 하여 이 과정에서 프로그래밍의 문법이나 알고리즘의 지식과 기능을 이해하게 된다[9].

2.1.3 개발중심모델

학생 스스로 탐구 과정을 설계하고 실행하며 새로운 지식을 얻거나 문제를 해결하는 활동을 통해 학습하는

탐구학습방법을 바탕으로 소프트웨어개발의 전 과정을 이해하는 모델이다. 탐구 단계에서 실제 프로그램을 분석, 탐구하여 개발하고자 하는 모듈이나 소프트웨어의 패턴을 스스로 파악하고, 구현할 알고리즘을 세부적으로 생각하여 계획하는 설계 단계를 거쳐 배운 것을 토대로 새로운 산출물을 개발하여 생산한다. 자신의 작품은 학습자들이 공유하거나, 디버깅 과정을 거치며 피드백을 실시하며, 이 과정을 통해 학습자의 컴퓨팅적 사고를 신장시킨다.

2.1.4 디자인 중심모델

디자인 중심모델은 단지 프로그램이나 소프트웨어의 기계적이거나 자동화된 부분에 초점을 두는 것이 아니라 소프트웨어의 개발이 삶을 개선하고 사람들의 요구에 맞게 하기 위함을 목적으로 하여 이에 부합되는 활동을 진행하는 학습법이다. 인간의 요구에 부합되도록 창의적으로 프로그래밍을 계획·설계·구현하여 산출해 봄으로써, 학습자가 컴퓨팅 사고력을 신장할 수 있게 된다.

2.1.5 CT요소 중심모델

학습자 스스로 문제를 파악하고 분석, 실행하는 문제해결학습법이 기반이 된 CT요소 중심모델은 문제를 해결가능한 단위로 분할하는 분해하고 분해된 문제에서 패턴을 파악하여 이 패턴의 복잡도를 줄여 빠르고 효율적으로 문제를 해결하도록 단순화, 추상화하여 공식을 설정하여 이를 토대로 공식이나 원리를 순서에 맞게 절차적인 구성을 한 후, 컴퓨터 용어로 구현하여 실행하는 프로그래밍 과정을 거치며 컴퓨팅 사고를 신장시키게 된다.

2.2 영어교과 교수학습모형

언어 학습은 오래전부터 효과적인 학습방법에 대한 논의가 활발히 진행되어 다양한 관점의 학습방법이 발표되었는데, 이전에는 문법적인 면이 강조되었다면, 오늘날에는 실생활에서 사용할 수 있는 실질적인 언어 기 능 습득이 강조되고 있다. 이런 흐름에 맞춰 초등 영어 교육에서도 초등학생에게 적합한 의사소통 능력 향상을 위한 6가지 교수학습방법을 다음과 같이 설명하였다[4].

2.2.1 자연 교수법

목표 언어를 모국어 습득과 같은 절차로 교수하는 방법으로 문법 해설과 번역 등을 하지 않고, 목표언어만을 사용하여 진행되는 수업모형이다. 이 때, 교사는 학생들에게 이 해나 추측 가능한 언어를 입력하여 학습 활동에 동기를 유발하도록 한다. 이 방법은 자연스럽게 이상적인 방법이지만, 목표언어에 언어 장벽이 있거나, 친근하지 않은 학생에게는 적합하지 않으며, 학습 시간이 오래 걸린다.

2.2.2 전신 반응 교수법

신체 행동을 통하여 언어를 학습하는 방법으로 교사의 행동이나 실물 제시를 통해 학생들이 의미를 추측 및 파악하고 직접 수행까지 하는 과정을 통해 자연스럽게 학습이 이루어지는 교수학습방법이다. 사고와 행동을 함께 하여 내용의 파악 및 기억의 지속 효과가 높으나, 초기이상의 학습에는 적용에 한계가 있다.

2.2.3 총체적 언어 교수법

언어를 요소별로 분해하지 않고 전체적, 종합적으로 가르치는 방법으로 학습자들이 실제 의사소통을 하며 자연스럽게 언어를 배우는 학습법이다. 학습자들의 생활과 요구에 맞는 상황에서의 표현을 행해봄으로써 좀 더 목표 언어에 능동적으로 자기화하여 참여할 수 있는 반면, 이런 표현이 가능한 상태까지 도달하기가 힘들거나 아동의 상황의 다양성이 보장되지 않을 수 있다는 점은 유의해야 한다.

2.2.4 의사소통 중심 교수법

학습자의 동기를 불러일으키는 상황 및 활동을 제시하여 학습자의 수준에서 의사소통을 할 수 있는 재료로서 언어를 제시하여 학습 후 직접 의사소통을 하도록 하는 교수법이다. 학습 동기가 강화되어 학습에 대한 참여는 증대되는 등 학습효과가 좋지만, 적절한 동기유발 상황 제시 등의 교사의 부담감과 학습자들의 수준차 등은 해결과제이다.

2.2.5 내용 중심 교수법

다양한 활동을 위한 도구로서의 언어를 사용함으로써 풍부한 상황에 대한 언어 표현을 무의식적으로 사용하며 익히는 교수학습방법이다. 다양한 활동을 통해 학습하므로 언어에 대한 거부감 없이 다양한 흥미와 관심사를 흡수할 수 있으나, 활동 및 학습의 목표가 언어습득이 아닌 활동수행에 치중될 수 있고, 활동에 적합한 언어로 학습이 제한될 수 있다.

2.2.6 과제 수행 교수법

목표언어를 사용하는 과제를 제시하여 과제를 수행하는 과정에서 언어를 익히는 교수학습법이다. 학생들이 교사중심이 아닌 학습자 중심으로 정보차 활동 등의 과제를 해결하는 과정에 능동적으로 참여하게 되므로 개인의 발화기회 및 실제 언어 사용 시간이 길어지는 반면, 적절하고 의미 있는 목표언어 연습을 위한 과제설정의 어려움과 수준차 등은 해결과제이다.

3. 모형 개발

3.1 개발 배경

영어교육이 다른 나라의 언어를 새롭게 배워 사용할 수 있게 하는 과정이라는 점에서, 컴퓨터의 언어를 처음 배워 사용하는 과정에서 컴퓨터의 사고구조를 이해하여 습득하는 것이 목적인 소프트웨어교육과 닮아있다. 특히 소프트웨어교육이 소프트웨어언어를 처음 학습하는 과정이므로 모국어룰 체계적으로 배우는 국어보다 모국어 아닌 영어를 배우는 과정과 유사하므로 영어교육과의 연계가 적합하다고 사료된다. 또한, 다양한 교육용 프로그래밍 언어(EPL) 중에서 스크래치는 상황 및 무대를 직접 구성하는 스토리 기반의 프로그램으로 상황에 따른 의사소통 기능을 익히는 것이 주 학습목표인 영어 교과활동의 흐름에 적용하기 적절하다고 사료된다. 스크래치 프로그램의 다양한 멀티미디어 기능과 공유하기 및 리믹스(편집) 등의 기능이 영어교과의 다양한 의사소

통 활동에 도움이 될 것이다. 소프트웨어교육의 관점에서, 스크래치 프로그램이 컴퓨팅 사고력의 신장을 목표로 하는 소프트웨어교육에 효과가 크다는 것이 많은 연구를 통해 검증되었고, 프로그램 수업모형을 프로그래밍의 기능적인 측면만이 아닌 무대 구성과 의사소통을 위한 도구의 측면으로 자연스럽게 학습에 적용할 수 있어 연계에 적용하고자 한다.

3.2 수업모형 개발 방향 및 구조

3.2.1 개발 방향 및 준거

본 연구를 통해 개발하는 수업모형은 4학년 학생을 대상으로 영어교과의 내용을 기반으로 한 프로그래밍 교육을 실시하는 것을 목적으로 한다. 소프트웨어교육이 2019년부터 초등학교 5학년 학생들부터 편성이 예정되어 있으므로, 본 수업모형은 4학년 학생들에게 소프트웨어교육을 처음 경험하는 기회를 제공할 것이다. 이를 고려하여 본 수업모형은 학생이 프로그래밍에 대한 긍정적인 경험을 가질 수 있도록 유도하며, 연계되는 영어 교과의 학습 목표 성취에도 도움이 되고, 교사가 현장에서 쉽게 적용할 수 있는 교수학습 방법과 자료로 구성하는 등의 방향을 가지고 개발토록 한다.

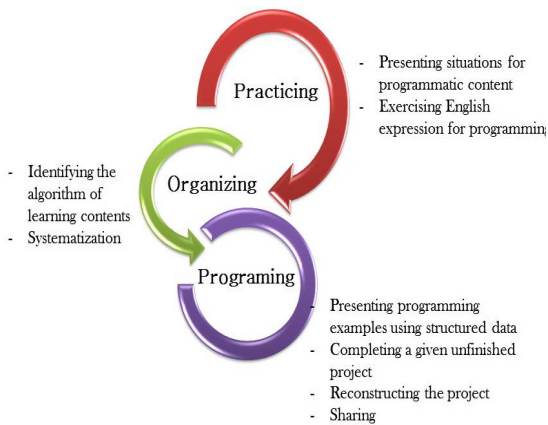
또한, 수업모형은 블록형 프로그래밍 언어인 스크래치의 다양한 기능을 경험하여 코딩에 대한 흥미와 학습의욕이 고취되고, 프로그래밍에 친숙해지도록 구성되어야 하며, 주어진 예시에 대한 재구성 과정을 통해 창의적 문제 해결력과 알고리즘적인 사고를 기를 수 있어야 한다.

3.2.2 수업모형의 구조

P.O.P.수업모형은 영어교과 내용과 활동이 자연스럽게 소프트웨어교육 활동으로 이어지는 흐름을 가진다. 앞서 이론적 배경에서 제시한 바와 같이 영어 교과 교육과정은 각 단원의 해당 언어 내용을 4차시에 걸쳐 듣기, 쓰기, 말하기, 읽기 등의 기능을 고르게 연습하며 진행하며, 교육과정의 내용은 의사소통 상황에 필요한 내용으로 구성되었다. 이 중 프로그래밍교육과 연계할 수 있고, 연계교육을 실시했을 때, 학습의 효과가 증대될 것으로 예상되는 내용을 선정하여 첫 번째 익히기 단계

에서는 평소의 영어교과 활동과 동일하게 상황과 목표 언어를 제시하고, 목표언어를 연습하는 흐름으로 진행한다. 다만, 제시하는 상황이나 활동이 3단계의 프로그래밍하기 단계에서 구현할 상황과 연계되도록 구성한다. 두 번째 체계화하기 단계에서는 익히기 단계를 통해 연습한 활동을 체계화, 절차화하는 언플러그드 활동으로 구성하여 다음 단계로 이어질 수 있는 시간을 가진다. 체계화하기 단계 활동 내용을 기반으로 세 번째 프로그래밍하기 단계에서는 완성형 스크래치 프로그램 한 편을 제시한다. 완성형 프로그램에 주요 목표언어를 비워놓은 미완성형 프로그램을 학생에게 배포하여 빈 공간을 채워 프로그램을 완성하도록 한다. 학생들이 완성한 프로그램을 자유롭게 변형하여 재구성하는 리믹스 활동을 실시하여 창의적인 프로젝트를 구현·공유하도록 한다.

P.O.P.수업모형은 2차시에 걸쳐 진행되며, 교육과정상 소프트웨어교육 시간이 확보되지 않았으므로, 익히기 단계와 체계화하기 단계는 영어교과시간에 실시하고, 프로그래밍 단계는 창의적 체험활동 시간 등을 활용토록 한다. 수업모형의 전체적인 흐름과 구조는 (Fig. 1) 과 같다.



(Fig. 1) Structure of P.O.P. Teaching Model

3.3 수업모형의 단계별 설명과 예시

3.3.1 1단계: 익히기(Practicing)

익히기 단계는 프로그래밍에서 구현될 내용과 관련된 의사소통 상황의 제시와, 프로그래밍 교육에 사용될 영

어 표현을 연습하는 데 주안점을 둔다. 본 단계는 정규 영어교과시간에 실시하도록 하며, 그런 만큼 학습이 목표언어를 익히는 의사소통 기능 향상에 초점을 맞추어 활동하게 된다. 수업 모형도 영어 교과의 의사소통중심 교수법을 중점 활용한다. 표면적으로는 영어교과시간과 같은 흐름을 가지지만, 학습 초기의 상황과 활동을 프로그래밍과 연계될 수 있는 내용으로 구성한다는 점이 차별화된다.

본 단계에서 교사는 프로그래밍으로 구현할 내용을 추출하여 적절한 스토리나 상황을 제시한다. 이는 학습자의 프로그래밍 학습에 대한 동기유발과 프로그래밍을 기능적인 학습이 아닌 맥락 속에서 이해할 수 있는 학습이 되도록 하는 과정이므로 신중하게 제시한다. 상황이나 스토리를 통해 제시된 스크래치의 애니메이션으로 완성할 영어표현인 목표언어를 다양한 자료와 활동을 통해 익히도록 한다.

3.3.2 2단계: 체계화하기(Organizing)

본 단계에서는 1단계 익히기 활동 내용에 대한 알고리즘을 파악하여 체계화하는 것에 중점을 둔다. 본 단계는 1단계의 익히기 활동 중에서 프로그래밍을 위해 필요한 내용을 추출하여 자료를 분석하고, 추상화 하는 등의 언플러그드 활동을 기본으로 한다. 그 외에도 이전 단계의 활동 중 스크래치 프로그램으로 구현될 수 있는 요소를 추출하는 활동도 포함한다. 체계화하기 단계는 내용 및 학습 흐름에 맞게 1차시 또는 2차시에 실시한다.

2단계에서는 연습한 활동 내용에 대한 알고리즘을 알고, 순서도를 완성하는 활동이 주가 되지만, 이 외에도 상황에 맞게 프로그램에 구현될 배경이나 스프라이트 디자인하기, 활용되는 게임이나 활동의 복잡한 로직(logic)에 대한 알고리즘을 파악하는 활동, 또는 직접 부른 노래를 프로그램에 적용하는 활동 등 다양한 활동이 가능하다.

3.3.3 3단계: 프로그래밍하기(Programing)

3단계는 선행 단계의 학습을 토대로 실제 프로그램을 구현해 보는 단계이다. 1단계에 제시된 상황이나 스토리를 배경으로 활동한 내용을 2단계 체계화를 거쳐서 프

로그래밍을 완성한다. 4학년 학생들이 프로그래밍 지식이 없는 상태이므로 학생이 바로 프로그래밍을 실시하지 않고 먼저 교사가 시연한 완성형 프로젝트를 보여주도록 한다. 이 과정은 학생에게 프로그래밍 욕구를 불러일으키고, 프로그램을 어떻게 제작하는 지에 대한 제작 방향을 제시하는 역할을 한다. 시연한 프로젝트에 목표 언어의 낱말이나 대화를 비운 미완성형 프로젝트를 교사가 스튜디오에 공유하여 배포하면, 학생은 이 프로젝트에 스크립트를 채워 완성하도록 한다. 이 과정에서 학생들은 간단한 활동이지만, 프로그래밍의 성취감을 경험할 수 있게 되고, 프로그램에 사용된 블록이나 기능의 종류 등을 파악할 수 있다. 완성한 프로젝트에서 기능, 대화 등을 재구성, 편집하여 나만의 스크래치 프로젝트를 창의적으로 구현한다. 이 때, 학생들의 편집 과정에 최대한의 자율성을 부여해 프로그램을 여러 방향에서 탐험해보는 시간을 가질 수 있다. 또한, 3단계에서 생성된 생성 프로젝트는 학급 스튜디오에 올려 다른 학생들과 공유하며, 피드백이 자연스럽게 이루어지도록 한다.

이와 같이 단계별로 살펴본 P.O.P.수업모형은 영어교과 시간과 창의적 체험활동 시간에 걸쳐 2차시 수업으로 진행되며, 각 단계의 중점 학습 기능에 따라 주 활용 수업모형도 변화한다. 1단계 익히기 활동에서는 영어 의사소통 기능 향상 중심의 학습의 중점 목표이므로 의사소통 중심 교수법으로, 2~3단계는 컴퓨팅 사고력 신장이 학습의 중점 목표로 진행되므로 소프트웨어수업모형을 활용하여 진행된다. 각 단계별로 다양한 활동이 이루어

어지지만 활동이 유기적인 흐름을 가지며, 그 내용을 정리하면 <Table 1>과 같다.

4. 검증 및 효과

본 연구에서는 초등영어교과 내용을 기반으로 한 프로그래밍 교육수업모형인 P.O.P.수업모형의 효과에 대해 검증하기 위해 전문가 검토 조사를 실시하여, 수업모형의 목표, 내용 등에 대한 콘텐츠에 대한 검증을 실시하고, 수업모형을 적용한 학생들에게 소프트웨어교육의 인식 및 태도 등에 대한 설문 조사를 실시하여 학습의 효과를 알아보았다.

4.1 검증방법 및 결과

4.1.1 검증방법 및 내용

본 수업모형에 대한 전문가 집단에게 검토 설문 조사를 실시하여 타당성을 검증하였다. 조사는 온라인과 오프라인 설문지를 동시에 진행하였다.

개발된 수업모형의 타당성과 학습의 효과를 검토하기 위해 조사 대상으로 초등교사 20명을 전문가 집단으로 선정하였다. 수업모형이 영어교육내용 기반이므로 영어교육을 실시한 경험이 있거나, 실시할 예정이어서 수업모형에 대하여 잘 이해할 수 있는 교사를 우선적으로 선정하였다.

검증에 대한 전문가 조사 항목은 신재훈(2017)의 예비 정보교사를 위한 정보보안 교육과정의 타당성 검증한 문항과 이현주(2012)의 수학교육과정에서 초등정보영재 프로그래밍 프로그램의 전문가 검토지 등을 바탕으로 본 연구의 개발 수업모형의 내용에 적합하게 수정 보완하여 모형의 목표, 구성 및 내용과 방법, 기대 효과 등을 포함하는 설문 문항을 구성하였다[10][14].

4.1.2 검증결과

수업모형의 학습 목표가 성취 가능한 수준으로 제시되었고, 학습자가 이해 가능하도록 제시되었다는 의견이

<Table 1> Flow of P.O.P. Teaching Model

Step	Classes	Period	Main use class model	Key learning function
Practicing	English	1	Presentation-Practice-Production	Improvement of communication skills in English
Organizing	English+Creative activity	1~2	Decomposition-Pattern Recognition-Abstraction-Algorithm-Programming	↓
Programing	Creative activity	2	Demonstration-Modeling-reCreating	Enhancement of Computing Thinking Power

<Table 2> Contents of the Teaching Model

	questionnaire	ratio (%)				
		Strongly Positive	Positive	neutral	negative	Strongly negative
1.	Understanding of Learning Objectives	40	60			
2.	The achievement potential of learning objectives	30	65	5		
3.	Contents of the English-language linkage	50	45	5		
4.	Appropriate level of learning content	30	60	10		
5.	Appropriateness of learning content	70	25	5		
6.	Appropriateness of Learning volume	20	65	10	5	
7.	Interest of Learning Contents	60	40			
8.	The appropriateness of the flow of learning contents	35	60	5		
9.	Appropriateness of teaching and learning methods	25	75			
10.	Appropriateness of presentation of learning materials	25	60	10		
11.	The appropriateness of the instructional model for self-directed learning	35	55	10		

높았고, 영어교과내용과 연계성이 고려되었으며, 학습 목표에 도달하는 데 필요한 내용이라는 의견이 높았다. 또한 학생의 흥미를 유발하였지만, 학습자의 수준과 학습 양의 고려가 다소 미흡하다는 의견도 있었다. 전체적

<Table 3> Application and the Expected Effects

	questionnaire	ratio (%)				
		Strongly Positive	Positive	neutral	negative	Strongly negative
12.	Applicability to school environment and parish use	55	40	5		
13.	Suitability of application within learning time	15	55	25	5	
14.	Application intent When securing environment	30	70			
15.	Cooperation in reaching English learning objectives	35	65			
16.	Usefulness as Programming Education Reconstruction Material	65	35			
17.	Enable for procedural thinking about problem solving.	20	65	15		
18.	Applicability of Computing Thinking in Problem Solving Process	25	60	15		
19.	Contribution for Software Education	45	45	10		

인 수업모형의 내용은 학습 흐름에 적절하게 구성되었다는 의견이 높았다.

수업모형의 교수·학습 방법은 적절하며 학습에 필요한 자료가 알맞게 제시되었다는 의견이 높았다. 또한, 수업모형의 방법과 자료는 학습자가 자기 스스로 학습할 수 있도록 제시되어 있다는 의견도 높았다.

수업모형이 실제 교실 환경 내에서 학교 현장의 교구의 사용으로 적용이 가능하다는 의견이 많았으나, 교과 수업 시간 내에 적용이 힘들다는 의견도 있었다. 그러나

이런 시간적 공간적 환경들이 확보된다면, 수업모형을 학습에 적용할 의향이 있다는 의견은 높아 수업모형의 활용 의지가 높았다.

또한, 수업모형은 영어교육과정의 학습 목표의 달성에도 도움을 주고, 차후 프로그래밍 교육을 위한 재구성 자료에 도움이 된다는 의견이 높았다. 수업모형의 내용이 컴퓨팅 시스템의 원리를 활용하고 학생들이 절차적으로 사고하여 컴퓨팅 사고력을 신장하도록 하여 소프트웨어교육에 도움을 준다는 의견이 높았다.

4.2 적용 및 효과

4.2.1 수업모형 적용 방법

본 수업모형을 서울시의 한 초등학교에서 한 학급 23명을 대상으로 실시하였다. 해당 학교에서 영어 교과 수업을 교과 전담교사가 담당하고 있어 1차시인 익히기(Practicing) 단계는 영어교과 전담교사가 실시하고, 2차시 체계화하기(Organizing)와 프로그래밍하기(Programing) 단계를 담임교사가 실시하였다. 2단계에는 내용 파악 또는 순서도 등의 언플러그드 활동을, 3단계에서는 활동한 내용을 스크래치 프로그램에서 구현하는 활동을 실시하였다.

본 수업모형 적용 후에 어떤 효과가 있었는지 학생들에 대한 설문 조사를 통해 검증하였다. 설문 항목은 김지현(2013)의 초등 정보 영재학생들을 위한 암호화 교육 프로그램에 대한 학생 설문 문항과 홍정미(2015)의 스크래치를 활용한 소프트웨어융합교육 프로그램에 대한 학생의 ICT소양 및 소프트웨어인식의 변화에 대한 설문 문항 등을 참고하여 구성하였다[5][8].

4.2.2 수업모형 적용 결과

<Table 4>의 내용과 같이 학생들은 P.O.P.수업모형을 적용한 수업에 적극적으로 참여한 것으로 나타났으며, 컴퓨터를 사용한 수업에 대한 만족도가 높았고, 특히 스크래치 프로그램을 사용한 수업에 더 흥미를 느끼고, 학생들 대부분이 P.O.P.수업모형을 적용한 수업에서 학습 내용이 이해되었고, 컴퓨터와 스크래치 프로그램을 사용한 수업이 학습에 도움이 되었다고 답하였다. 또한, 수업한 내용도 기억에 남을 것이라는 의견이 많아

<Table 4> Effectiveness after Application

No	questionnaire	ratio (%)				
		Strongly Positive	Positive	neutral	negative	Strongly negative
1.	Interest in computer-based instruction	74	13	13		
2.	Interest in class using scratch programs	83	13	4		
3.	Activeness in class participation	74	17	7		
4.	Understanding the content to learn	87	4	7		
5.	Remain in head about the content	70	22	7		
6.	Efficacy of learning computer use lessons	65	22	13		
7.	Efficacy of learning in class using scratch programs	74	26	4		
8.	Willingness to create works using computers	87	13			
9.	Awareness of the utility of programming	70	13	17		
10.	Information exchange and creation using computer	70	22	7		

P.O.P.수업모형의 내용에 대해 학생들이 이해하였으며, 학습에 유의미한 효과가 있는 것으로 판단된다. 또한, 학생들이 컴퓨터를 이용한 창작물을 만드는 것에 긍정적인 사고를 가지는 응답 비율이 높았고, 프로그래밍이 생활에 도움이 된다는 인식을 가지는 학생이 많고, 프로그래밍에 대한 창작 의지도 높은 것으로 나타났다.

기타 의견으로 수업의 가장 흥미롭고 기억에 남는 부분을 묻는 질문에 스크래치를 배운 것과, 스프라이트를 직접 만들고 움직이게 한 것이라는 등의 의견이 많아, 학생들이 스크래치를 이용한 프로그래밍 경험이 긍정적이었고, 프로그램에서 창작하는 활동에 대한 만족도가 높았음을 알 수 있다.

5. 논의 및 결론

본 연구의 결과는 다음과 같다.

첫째, 초등학교 4학년에 적용할 수 있는 영어교과연계 프로그래밍 교육과정을 개발하였다. 초등 4학년 영어교육 과정 내용 중 프로그래밍의 다양한 기능을 경험할 수 있고, 프로그래밍으로 구현할 수 있는 요소를 추출해 학습 내용을 선정하고, 이를 바탕으로 영어 교과 시간과 창의적 체험 활동 시간의 2차시에 걸쳐 ‘익히기(Practicing)-체계화하기(Organizing)-프로그래밍하기(Programming)’의 P.O.P. 학습 단계를 구성하고, 본 단계에 따라 구성된 교육과정의 교수·학습 과정안과 학습 자료를 개발하였다.

둘째, 개발한 P.O.P.수업모형과 이에 따른 교육과정은 전문가 집단의 검토를 통해 교육과정의 목표, 구성, 내용, 방법, 기대효과 등의 타당성을 검증받았다. 그 결과, 개발된 수업 모형의 내용과 방법, 기대효과 등이 전반적으로 타당하다고 검증되었다. 다만, 연계교육을 실시할 시간과 교구 및 컴퓨터실 확보 등의 충분한 제반 환경의 미흡과 영어교과 전담교사와의 연계 등과 같은 우려가 있어 이 점은 추후 해결과제이다.

셋째, 학생들이 본 수업 모형을 통해, 소프트웨어 프로그램 제작에 대한 긍정적인 인식을 가지게 되었고, 프로그래밍 교육에 흥미를 가지고 적극적이고 능동적으로 참여한 것으로 드러났다. 초등 4학년 한 학급 학생들에게 개발된 수업모형에 따른 교육과정을 적용하여 수업을 진행한 후 실시한 설문에서, 학생들은 흥미를 가지고 적극적으로 학습에 참여했으며, 프로그래밍 창작에 대해 긍정적인 인식을 가지게 되었고, 수업의 이해도도 높았다는 결과가 나타났다. 이는 본 수업 모형의 중점 개발 방향이 학생들이 프로그래밍에 대한 긍정적인 경험을 가지도록 유도하는 것임을 감안하면 본 수업 모형은 유의미한 학습의 효과를 거둔 것으로 판단된다.

이상의 연구결과를 토대로 제언을 하자면, 다음과 같다.

첫째, 프로그래밍 교육을 실시할 때, 프로그래밍의 기초와 원리부터 단계별로 학습해야 한다는 인식을 전환해야 한다. 현대 사회에는 이미 수준 높은 프로그램들이 일상화되어 있다. 이러한 상황에서 기본·기초적인 원리 위주의 교육을 실시하면, 학생들의 학습동기를 유발하지 못하고, 학습의욕이 저하될 수 있을 것이다. 그러므로 학생들에게 새로운 방식으로 동기를 유발하고, 실제 구

현을 경험해 볼 수 있도록 한다면 학생들이 성취감을 느끼고, 지속적으로 프로그래밍에 대한 흥미와 학습의지를 고취시킬 수 있을 것이다.

둘째, 소프트웨어교육과 다양한 교과와의 연계가 필요하다. 그동안의 연계는 수학, 과학 교과 등에 치우쳐 논리적인 측면이 강조되어 시도되어 왔다. 그 결과, 소프트웨어교육의 선호도 또한 성향에 따라 편중되는 경향이 있어, 소프트웨어교육을 다양한 측면에서 접근하는 노력이 요구된다. 표현 수단의 측면에서 언어 교과와 연계하거나, 창의적이고 표현적인 측면을 강조하여 예체능 교과와 연계하는 등의 새로운 교과와 교수 학습 방법의 연계교육을 실시한다면, 소프트웨어교육 시간을 확보함과 함께, 다양한 성향과 기질을 가진 학생들을 학습에 유인하고 몰입하게 하는 유의미한 효과가 있을 것이다.

셋째, 소프트웨어교육에 대한 다양한 홍보 및 지원이 필요하다. 우리나라는 ‘소프트웨어 중심사회’의 정책적인 노력으로 방대한 자료들을 개발해 왔으나 실제 교육 현장에서 활용도가 높지 않았다. 실제로 초등학교 교사들이 소프트웨어의 중요성과 프로그래밍 교육에 대한 필요성을 인식하고 있지만, 소프트웨어교육에 대한 지식이 부족하고, 2019년부터 실시될 내용조차 인지하지 못하는 교사들도 많은 현실이다[1]. 단위교육청과 한국교육학술정보원(KERIS) 등의 기관들은 교사들이 소프트웨어교육에 대해 알 수 있는 다양한 홍보 및 지원방안을 마련하고, 개발된 자료들을 재구성·재배치하는 등의 노력을 통해 실제 현장에서의 활용도를 높여야 할 것이다. 또한, 현재 선도학교 위주로 실시되고 있는 소프트웨어교육을 단위 학교 급별로 선도요원을 선정하여 각 학교에 적합한 맞춤형 소프트웨어교육 방법을 연구하는 등의 적극적인 노력이 필요하다.

참고 문헌

- [1] Allen Tucker A (2003). Model Curriculum for K - 12 Computer Science: New York, CSTA.
- [2] CAS (2012). Computer science : A curriculum for schools, Computing At School. Available at: <http://www.computingschool.org.uk/data/uploads/ComputingCurric.pdf>

- [3] Deborah Seehorn, Stephen Carey, Daniel Moix, Brian Fuschetto, Irene Lee, Dianne O’Grady-Cuniff, Chris Stephenson, Anita Verno (2011). CSTA K-12 Computer Science Standards Revised 2011.
- [4] Ham Soonae (2009). Elementary school English Teacher’s guide. 4. Chun-jae.
- [5] Hong, Jungmi (2015). The Effect of Convergent Software Education Program using Scratch on Elementary Students’ ICT Literacy and their Awareness on Software Education. Master Degree paper. Gyeongin National University of Education.
- [6] KERIS (2015). Guidelines of Software Education Management.
- [7] Kim, Kapsu(2016). A Recognition Analysis of Elementary Teachers for Software Education of 2015 Revised Korea Curriculum. *Journal of The Korea Association of Information Education, Vol.20*. No. 1
- [8] Kim, Jihyun (2013). A Development and Application of Cryptography Learning Program for the Gifted Elementary Students of Information. Master Degree paper. Seoul National University of Education.
- [9] Kim, Jinsook (2015). A research on the Development of Teaching and Learning Models for SW Education. (62-65,78). Korean Educational Development Institute.
- [10] Lee, Hyunju (2012). A Study on Development of Programming Program for Elementary Gifted Children of Information in Math Curriculum. Master Degree paper. Seoul National University of Education.
- [11] Ministry of Education (2015). Guidelines for Software Education Operation. Ministry of Education. 2-3. MOE.
- [12] Ministry of Education (2015). The revised national curriculum 2015 for Primary and Secondary Schools, Vol.2015-75, No. 10.
- [13] Mustafa ILKHAN (2013). Faith—The Movement of increasing opportunities and improving Technology. Proceeding of Global Symposium ICT in Education. KERIS.
- [14] Shin, Jaehoon (2017). Design and Validity Verification of Information Security Curriculum for Informatics Pre service Teachers. Master Degree paper. Korea National University of Education.
- [15] Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C. and Verno, A. (2002). A Model Curriculum for K-12 Computer Science. Report of the ACM K-12 Education Task Force Computer Science Curriculum Committee.
- [16] Yoo, Jeongho (2015). Development of Coding Education Program Based on Mathematics Contents. Master Degree paper. Seoul National University of Education.

저자소개



허미연

2003 서울교육대학교 컴퓨터교육과 (학사)
 2017 서울교육대학교 컴퓨터교육과 (석사)
 2003~2010 서울성내초등학교 교사
 2010~2015 서울남사초등학교 교사
 2015~현재 서울상도초등학교 교사
 관심분야: SW교육, 프로그래밍교육, 영어교육, 수학정보 영재교육
 e-mail: versina11@naver.com



김 갑 수

- 1985 서울대학교 계산통계학과
(학사)
- 1987 서울대학교 계산통계학과
전산학전공(석사)
- 1996 서울대학교 계산통계학과
전산학전공(박사)
- 1987~1992 삼성전자 사원-과장
- 1995~1998 서경대학교 전임강사
-조교수
- 1998~현재 서울교육대학교 컴퓨
터교육과 조교수-교수
- 관심분야: 컴퓨터 교육, SW 공학,
정보 영재, 기능성 게임
- e-mail: kskim@snue.ac.kr