JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 18, No. 7, pp. 1313-1321, Nov. 2017

비전공자 SW 프로그래밍 교육과정 및 콘텐츠 개발 모형의 효과성 탐색: SW 해결안의 시각적 표현을 중심으로

이 민 정[1]
[1]중앙대학교 창의ICT공과대학 컴퓨터공학부

Exploring the Effect of SW Programming Curriculum and Content Development Model for Non-majors College Students : focusing on Visual Representation of SW Solutions

Minjeong Lee[1]
[1]School of Computer Science and Engineering, Chung-Ang University, Seoul 05974, Korea

[요 약]

ICT 기반 디지털 융합이 가치를 창출하는 미래사회에는 다양한 분야의 전문가가 협업하며 SW적 문제 해결을 주도하는 역량이 더욱 강조된다. 비전공자에게는 전문 분야의 문제를 풀기 위해 SW 전문가와 효과적으로 협업할 수 있는 SW적 소통 능력이 필요하다. 따라서 비전공자를 위한 SW 교육은 높은 수준의 코딩 역량을 목표로 하는 기존의 SW 전공자 대상 프로그래밍 교육과는 달라야 한다. 또한 다이어그램 기반의 시각적 표현이 원활한 소통과 협업에 도움이 된다는 것은 이미 알려져 있다. 본 연구에서는 비전공자를 위한 SW 교육목표를 'SW적 문제해결을 위한 시각적 프로그래밍 역량 함양'이라 정의하고, 이를 달성하기 위한 비전공자 SW 프로그래밍 교육과정과 SW 해결안의 시각적 표현에 중점을 둔 SW 교육 콘텐츠 개발 모형을 탐색하였다. 본 논문의 결과는 비전공자를 위한 적절한 SW 학습 방안을 마련하고 실질적인 SW 역량을 함양하기 위한 방향을 설정하는 데 도움이 될 것이다.

[Abstract]

In the future society where ICT-based digital convergence creates new value, collaborative skills among experts in various fields and SW based problem solving ability is more emphasized. Non-SW specialists are required to have SW based communication skills to effectively collaborate with SW experts to solve their problems. Therefore, SW programming curriculum for non-major college students should be different from the existing programming education for SW-majors aiming at a high level of coding ability. It is also known that diagram-based visual representation is helpful for productive communication and collaboration. In this study, we defined the SW education objectives for the non-majors as cultivating the visual programming ability for SW based problem solving. In order to accomplish this, we explored SW programming curriculum and content development model for non-majors focusing on visual representation of SW solutions. The results of this paper will help to provide appropriate SW learning model for non-majors and to cultivate practical SW capabilities.

색인어 : SW 프로그래밍 교육과정, 콘텐츠 개발 모형, 비전공자, 문제해결, 시각적 표현
Key word : SW programming curriculum, Content Model, Non-major Students, Problem Solving, Visual Representation

http://dx.doi.org/10.9728/dcs.2017.18.7.1313

Received 03 November 2017;**Revised** 18 November 2017
Accepted 25 November 2017

***Corresponding Author; Minjeong Lee**

Tel: +82-10-23645-6486
E-mail: minjeonglee@cau.ac.kr

Ⅰ. 서 론

인공지능, IoT, 빅데이터, 모바일 통신 등의 정보통신기술(ICT: Information and Communication Technology)을 기반으로 가상현실, 드론, 자율주행자동차, 로봇산업 등 상상 속에 있던 새로운 산업이 등장하고 핀테크, 스마트팜, 신재생에너지, 스마트시티, 스마트공장, 스마트헬스 등 기존 산업과 ICT가 융복합하여 새로운 가치를 재창출하는 4차 산업혁명 시대를 맞이하고 있다. 이처럼 다양한 산업 분야의 요구에 ICT와 같은 첨단기술을 유연하게 적용하고 확장할 수 있게 하는 가장 중요한 요소가 소프트웨어이다[1],[2].

따라서 4차 산업혁명으로 도래할 미래의 디지털 사회에서는 모든 분야에서 ICT와 융합하는 소프트파워를 실현하기 위해 전공과 관계없이 SW적 문제해결 역량을 가진 인재가 필요하다. SW적 문제해결 역량이란 SW를 기반으로 창의적이고 효과적이며 효율적으로 문제를 해결하는 역량을 의미한다. SW적 문제해결 역량이 강조되면서 컴퓨터의 정보처리 관점에서 문제를 해석하고 해결안을 설계하고 구현할 수 있는 컴퓨팅 사고력(computational thinking)을 증진하기 위한 연구가 활발하게 이루어지고 있으며, SW 프로그래밍 교육을 통해 문제해결 역량이 향상된다는 다수의 연구 결과를 확인할 수 있다[3]-[5].

이에 따라 대학에서도 미래사회에 대비한 실질적인 SW적 문제해결 역량을 갖춘 인재를 양성하기 위해 전체 재학생을 대상으로 컴퓨팅 사고력을 함양할 수 있는 SW 교육을 진행하는 추세이다. 그런데, 컴퓨터가 생소한 비이공계열의 SW 비전공자를 대상으로 SW 전공자와 동일한 방식으로 SW 프로그래밍 교육을 진행한다면 학습 효과나 성취도 면에서 좋은 결과를 도출하기 어렵다[6].

비전공자들은 고등학교 교육내용과 전공 교육과정에서 상대적으로 수학적 사고 학습 비중이 낮고 절차적 문제해결 과정보다는 통합적 문제분석 훈련에 집중해 왔기 때문에 문제해결을 위한 사고 체계와 전개 과정이 컴퓨팅 사고와는 차이가 있을 수 있다[7]. 여기에 심리적으로 컴퓨터에 대한 막연한 두려움을 갖기도 한다. 이들 중에는 사회 변화에 대비하기 위해 자발적으로 SW 학습을 시작하는 경우도 있으나 비전공 학생 대다수는 학교에서 주도하는 커리큘럼에 수동적으로 참여한다. 이들은 학습 도중 어려움에 봉착했을 때 쉽게 포기하는 경향을 보이기도 하지만 적정한 수준에서 한계를 극복했을 때 학습 몰입도가 상승하고 SW 분야에 흥미와 자신감을 얻는 양상을 보인다는 점에서 SW 프로그래밍 교육과정을 검토할 필요가 있다.

사실상 SW 중심 사회의 모든 구성원이 SW 전문 개발자일 필요는 없다. 오히려 다양한 분야의 종사자들이 각자의 관점에서 SW의 구조와 역할을 이해하고 최신 ICT 기술을 습득할 수 있다면 이들 간의 협업을 통해 ICT 융복합 활동이 더욱 시너지를 낼 수 있다. 이처럼 전공 분야에 따른 학습자 특징과 사회적 역할에 따라 요구되는 SW 역량의 차이를 고려하여, 높은 수준의 코딩 역량을 요구하는 기존의 프로그래밍 교육과는 차별되는 비전공자를 위한 SW 프로그래밍 교육과정이 필요하다.

이에 본 연구에서는 비전공자를 위한 SW 프로그래밍 교육과정과 문제해결안의 시각적 표현을 기반으로 한 콘텐츠 모형을 제안하고 그 효과성을 탐색하였다

비전공자 SW 프로그래밍 교육과정의 목표를 'SW적 문제해결을 위한 시각적 프로그래밍 역량 함양'으로 설정하고 이를 달성하기 위한 교육내용으로 '① 일상의 문제를 컴퓨팅 환경의 문제로 재구성', '② SW 해결안의 시각적 표현', '③ 시각화한 SW 해결안 기반 소통과 협업'을 구성하였다. 비전공자들에게 가장 필요한 학습으로 정보처리 관점에서의 컴퓨팅 환경에 대한 이해와 SW적 해결안의 설계 과정에 집중하기 위해 순서도, NS차트, 동작흐름도와 같은 다이어그램을 이용한 비주얼 프로그래밍을 활용하였으며 SW 프로그래밍 교육방법으로서 주차별 학습 모델과 콘텐츠를 제시한다.

본 연구의 비전공자 SW 프로그래밍 교육과정과 콘텐츠 개발 모형은 비전공자에 대한 적절한 SW 프로그래밍 학습 방안을 제시하고 실질적으로 미래 사회를 대비하는 교육 효과를 제고하는 데 기여할 것이다.

Ⅱ. 관련기술 연구

2-1 비전공자를 위한 SW 교육에 관한 고찰

대학의 SW 교육이 전체 재학생으로 확대되는 추세에 따라 SW 전공자와 학습 배경과 특성이 다른 비전공자를 대상으로 교육을 진행하게 된다. 우리나라의 인문·사회·경영·예체능 등 인문계열 학생들은 입시 준비 과정에서 수학이나 과학보다는 사회, 언어, 역사 과목을 중심으로 다양한 사회적 주제를 토의 및 토론하며 논술적 사고력을 강화하는 과정에 집중한다[7]. 이는 비전공자들이 절차적 알고리즘에 의한 문제해결 과정을 다루는 SW 교육을 처음 대할 때 초보 학습자로서 생소함을 느끼는 요인이라 할 수 있다.

김수환[8]은 비전공자를 대상으로 진행한 컴퓨팅 사고력 교육 과정에서 나타난 학습자들의 어려움을 분석하였는데, 가장 어려워하는 부분은 변수와 리스트의 학습, 아이디어를 생각하고 구현하는 과정, 적합한 명령어의 사용의 순으로 나타났다. 또한 교육 내용이 학습자의 흥미와 재미를 유발했을 때 컴퓨팅 사고력 향상에 긍정적인 영향을 준다는 점을 확인하였다. 또한, 성정숙[9]은 SW 교육에서의 초보 학습자의 특성을 분석하였는데 프로그래밍 수업이 진행되면서 난이도가 높아질수록 흥미가 떨어지다가 개별 프로젝트를 끝낸 후 흥미가 급상승하는 것을 확인하였다. 따라서 비전공 학생들을 위한 SW 교육은 프로그래밍 언어의 문법을 차례로 학습하는 단선적인 구성보다 다양한 주제를 중첩하며 학습자가 직접 문제를 풀어내는 과정

에서 흥미를 유발하고 성취감을 느낄 수 있는 나선형의 교육 내용으로 구성할 필요가 있다.

전수진[10]은 SW 교육 콘텐츠를 스토리텔링, 게임, 미디어 아트, 교육학습 콘텐츠, 시뮬레이션, 실생활 중심 콘텐츠로 6가지 주제 영역으로 나누고 비전공자들의 SW 교육 콘텐츠에 대한 선호도를 분석하였다. 그 결과 주로 개인의 흥미와 교수자의 영향을 받지만 게임과 스토리텔링 주제에 선호가 가장 높다는 것을 확인하였다. 이를 기반으로 SW 교육내용을 설계할 때 학습자의 선호도와 특성을 반영한다면 학습 참여도와 컴퓨팅 사고력 향상에 도움이 될 것이다.

2-2 컴퓨팅 사고력과 SW 프로그래밍 교육

Wing[11]은 일상의 문제를 해결함에 있어 문제의 핵심 요소를 추출하고 해법을 컴퓨팅 기기로 자동화할 수 있는 능력을 컴퓨팅 사고라고 정의하였다. 즉, 컴퓨팅 사고가 컴퓨터 과학이나 정보통신 지식의 학습 수준이나 SW 프로그래밍 능력 자체를 의미하는 것이 아니라, 문제를 체계적으로 인식하여 추상화하고, 필요한 자료를 수집하고 분석할 수 있으며, 최적화된 해법을 찾아 정보과학적으로 구현하는 방법을 설계하는 전체 과정에 필요한 사고 체계로 간주하였다.

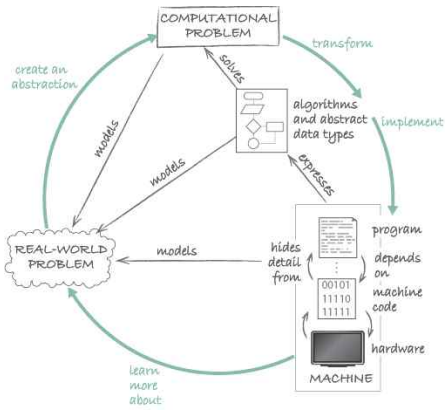

그림 1. Wing이 제안한 컴퓨팅 사고력의 개요 [3]
Fig. 1. Computational thinking overview diagram in [3]

그림1은 Wing의 컴퓨팅 사고에 의한 문제해결 과정을 표현한 것이다. 실세계의 문제를 컴퓨팅 환경의 문제로 전환한 후, 데이터 구조와 알고리즘 절차로 표현하고, 코딩을 통해 자동화하는 순환 구조를 나타내고 있다. 그는 또한 컴퓨팅 사고력이 컴퓨터 전공자는 물론, 언어, 수학, 의학, 경영, 법학, 정치, 예술 등 모든 분야에도 적용되는 효율적이고 체계적인 문제해결 역량이라고 강조하였는데, 이러한 면에서 컴퓨팅 사고력은 산업간 융복합이 활발하게 일어나고 있는 현시대의 인재상에 매우 적합하고 필수적인 역량이라 할 수 있다[3],[11].

컴퓨팅 사고력 기반의 문제해결 역량에 대한 관심이 높아지면서 이를 향상할 수 있는 교육과정에 관한 연구와 적용이 활발하게 진행되고 있다. 한편, SW 프로그래밍은 목표한 기능을 구현하기 위해 절차적이고 논리적인 과정으로 이루어지므로 논리적 사고력과 창의적 문제해결 능력을 향상시킨다고 알려져 있다[4],[5].

미국은 전산학 교사협회(CSTA : Computer Science Teachers Association)에서 컴퓨터 과학을 STEM(Science, Technology, Engineering, Mathematics) 교육의 핵심 영역으로 정하고 단순히 프로그래밍 스킬을 넘어 소프트웨어 설계를 포함한 개발 능력을 키울 수 있는 컴퓨팅 사고력 기반의 SW 교육을 K-12학년 과정과 대학 과정에 적용하고 있다[12]-[14].

영국에서는 "Computing" 과목을 필수로 이수하도록 하고 있으며, 이스라엘, 중국, 핀란드 등 해외 많은 나라에서 컴퓨팅 사고력의 중요성을 인식하고 SW 교육을 초등 정규 과정에 도입하고 있다[15].

2-3 초보 개발자를 위한 프로그래밍 교육

1) 언플러그드 프로그래밍(Unplugged Programming)

언플러그드 프로그래밍은 컴퓨터나 SW 없이 활동이나 놀이를 통해 컴퓨터과학 개념 및 알고리즘을 학습할 수 있는 교수·학습 방법으로 팀 벨(Tim Bell) 교수가 제안한 방법이다. 언플러그드 활동은 학생들 스스로가 문제를 해결하게 도전하는 활동이 필요하고, 이 활동을 통해 발견학습이 가능하며, 이를 위한 설계가 면밀하게 이루어져야 한다. 언플러그드에 필요한 교수학습 자료로는 비디오 시연, 실제 실연, 외부 활동 등 다양할 수 있다. 언플러그드 프로그램에 적합한 주제와 활동으로 잘 설계된 20개의 주제가 개발되어 실생활 관련 놀이 형태로 컴퓨터 과학을 쉽고 재미있게 배울 수 있다[16].

2) 다이어그램(순서도, NS차트, 동작흐름도)을 이용한 시각적 프로그래밍

프로그래밍 작업은 문제 해결에 필요한 알고리즘만에 집중할 수 없고 프로그래밍 언어 문법에 따라 코드를 작성해야 하므로 문법을 공부하고 문법에 맞는지를 계속 확인해야 한다. 그러나 순서도, NS 챠트, UML과 같은 다이어그램을 이용한 시각적 프로그래밍(visual programming)은 코딩과정 없이도 문제에 집중할 수 있게 해준다[17]. 다이어그램 작성을 위해서는 간단한 기호들만 익히면 되기 때문에 프로그래밍 언어 문법을 익히고 기억하는 것에 비해 학습 시간과 노력이 들지 않는다. 최근에는 다이어그램 자체를 해석하여 실행시켜주는 다양한 도구들을 사용할 수 있기 때문에 다이어그램을 이용해 문제를 해결한 후 실행을 통한 검증이 가능하다[17]. 이로써 프로그래밍 언어를 이용한 코딩 과정을 생략하고 알고리즘에 집중한 프로그래밍 학습과 문제해결 학습이 가능하다.

3) 고급 언어 프로그래밍

프로그래밍 교육에 가장 많이 활용되는 언어로는 C, Java 등이 고급 프로그래밍 언어들이다. 중고등학교 정보·컴퓨터 과목

에서도 C언어를 기반으로 프로그래밍 교육이 이루어지고 있다. 그러나 고급 프로그래밍 언어 수업 설계를 살펴보면 문법 위주의 내용이 대부분을 차지하고 있어 실질적 알고리즘 교육이 되기 어렵다. 또한 디버깅에 많은 시간을 투자하고 있어 프로그래밍 학습의 흥미를 잃게 하는 요인이 되기도 한다.

4) 교육용 프로그래밍 언어

학생들을 대상으로 하는 프로그램 교육은 실제적인 문제해결에 주력하기 보다는 학습을 통해 논리적 사고력, 문제해결력 등의 고등 사고력 신장이 목적이라는 점에서 고급 프로그래밍 언어를 통한 교육의 한계점을 극복하기 위해 교육용 프로그래밍 언어가 등장하였다. 교육용 프로그래밍 언어는 문법 암기에서 벗어나, 자신의 생각을 블록이나 간단한 스크립트로 표현하여 프로그래밍할 수 있게 하는 언어이다. 국내에서는 스크래치(Scratch), 두리틀(DoLittle), 로고(Logo) 등을 활용하고 있으며 다양한 교과와 연계한 융합 교육에도 많이 활용하고 있다[18].

Ⅲ. 비전공자 SW 프로그래밍 교육과정 개발

3-1 비전공자 SW 프로그래밍 교육과정의 설계

강성원[19]은 프로그램과 SW를 혼돈하는 것을 경계하며 프로그램은 특정 프로그래밍 언어로 작성된 코드이고 SW는 프로그램을 제작하기 위한 전체 활동 과정을 포함한다고 구분하였다. 이에 따라 SW 역량은 소프트 콘텐츠 역량과 SW개발 역량으로 구성되며 소프트 콘텐츠 역량이 SW의 콘텐츠를 창작해 내는 역량이라면 SW개발 역량은 소프트 콘텐츠를 프로그램으로 제작하는 능력이라는 것이다. 미래사회가 요구하는 이종 분야 간 융복합 활동을 주도하기 위해서는 설계측면으로는 소프트 콘텐츠 역량이 필요하고 구현측면으로는 SW개발 역량이 필요하므로, SW 전공자는 SW개발 역량 향상에만 집중할 것이 아니라 창의적 소프트 콘텐츠 설계를 위한 인문적 학습을 병행해야 한다고 주장하였다.

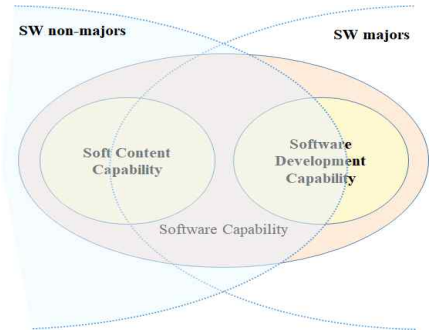

그림 2. SW 역량과 구성 [19]
Fig. 2. SW capability and components

한편 SW 비전공자의 관점에서는 자신의 전문 분야에 SW적 특성을 녹여내는 소프트 콘텐츠 설계에 상대적으로 강점이 있다. 그러므로 비전공자는 기본적인 SW개발 역량을 갖추되 소프트 콘텐츠 설계 역량 강화에 집중하는 것이 유리할 것이다. 또한 소프트 콘텐츠의 구현을 위해 높은 수준의 SW개발 역량이 필요한 경우를 대비하여 SW 전문가와 원활히 협업할 수 있도록 SW적 소통 기술을 배양하는 것이 효과적인 SW 역량 강화 활동이라 할 수 있다.

본 연구에서는 미래사회의 SW비전공 분야의 인재상을 자신의 분야에서 소프트 콘텐츠 역량을 기반으로 한 SW 융복합을 선도하고 SW적 소통과 협업이 가능한 인재라고 보고 이에 적합한 비전공자 SW 프로그래밍 교육과정과 콘텐츠 개발 모형을 탐색하였다.

비전공자 대상 SW 프로그래밍 교육과정의 목표를 'SW적 문제해결을 위한 시각적 프로그래밍 역량 함양으로 정하고 이를 달성하기 위한 3개의 수행 목표를 함께 제시하였다.

(1) 일상의 문제를 해석하여 컴퓨팅(디지털) 환경의 문제로 재구성할 수 있다.
- 실세계에서 발견한 문제나 새로운 아이디어를 컴퓨터와 SW로 해결하려면 컴퓨터에서 실제로 동작할 수 있는 수준으로 문제의 규모를 제한하거나 동작 조건을 명확하게 확정해야 한다. 이처럼 실세계와 컴퓨팅 환경 사이의 차이를 이해하기 위해 기본적인 컴퓨터의 구조와 SW 동작과 제어 방법을 학습한다.

(2) 컴퓨팅 환경 내 변환된 문제의 SW 해결안을 다이어그램 기반으로 명확하고 시각적으로 표현할 수 있다.
- 컴퓨팅 환경으로 변환된 문제의 해결안을 도출해야 하는데, 이 때 해결안을 반드시 코딩하여 SW로 만드는 것을 최종 목표로 삼지 않아도 된다. 대신 해결안의 주요 동작을 순서도, NS차트, 동작흐름도 등의 다이어그램을 활용하여 명시적이고 시각적으로 표현하는 연습을 반드시 진행한다. 이 과정은 초보 개발자가 SW 개발 작업에 매몰되지 않고 애초에 만들고자 했던 소프트 콘텐츠에 집중하여 해결안을 구성하고 검증할 수 있는 역량을 키울 수 있게 해준다[17].

(3) 시각화한 SW 해결안을 기반으로 소통, 협업하여 복잡한 문제를 해결할 수 있다.
- 복잡하거나 다양한 요구 사항이 포함된 문제를 여러 사람이 함께 작업할 경우 효율적인 업무 분장과 원활한 소통 환경이 결과물의 품질을 좌우한다[20]. 일단 복잡한 문제를 여러 개의 하위 문제로 분해한 후 각 하위 문제의 독립적 해법을 찾아 다이어그램으로 시각화할 수 있다면, 전체 해결안은 하위 문제의 다이어그램들을 서로 연결하여 체계적으로 구성할 수 있다. 이를 여러 사람이 함께 해결해야 한다면, 각 하위 문제를 기준으로 업무를 배분한 후 각

팀원이 도출한 하위 문제의 SW 해결안을 다이어그램으로 서로에게 설명하게 한다. 최종적으로 모든 하위 문제의 시각화한 SW 해결안을 서로 연결함으로써 전체 문제의 해법을 완성하는 과정을 학습한다. 이 과정을 통해 문제해결을 위한 협업의 소통 매체로서 다이어그램으로 시각화한 SW 해결안을 활용하는 역량을 키울 수 있다.

3-2 비전공자 SW 프로그래밍 교육내용 및 활동

본 연구에서 제안한 비전공자를 위한 SW 프로그래밍 교육과정을 구현할 콘텐츠 개발 모형을 표 1과 같이 설계하였다.

표 1. 비전공자 SW 프로그래밍 교육과정의 단계별 콘텐츠 모형
Table 1. Content Model of SW programming education for non-Majors

Goal	Content	Activity
① transforming real problems into computational problems	- computer architecture and SW operation - data structure - procedural description of mathematical problems - procedural description of real world situation	procedural description of solutions (writing)
② visualization of SW solution	- diagramming tools - repeating patterns - animation/simulation - role playing	diagramming and coding of solutions (individual project)
③ communication and collaboration based on visualized SW solution	- daily living problem - team project - R&R, collaboration	problem analysis, solutions design, implementation, presentation, evaluation (team project)

첫 번째 단계에는 실세계를 컴퓨팅 환경에 반영하기 위해서 컴퓨터 구조와 SW의 동작에 관한 학습을 가장 먼저 진행한다. 학생들의 흥미를 끌 수 있도록 직접 HW 보드를 살펴보거나 SW의 동작을 절차에 따라 문장쓰기를 해보는 과정을 병행할 수 있다.

두 번째 단계로 일상의 문제를 컴퓨팅 환경에서의 문제로 전환하여 해법을 도출한다. 이를 위해서는 무엇보다 다양한 유형의 일상의 문제를 실세계로부터 컴퓨팅 환경으로 투사하는 작업을 반복적으로 수행하는 것이 효과적이라고 알려져 있다[21]. 전수진[15]은 SW 교육 콘텐츠 중 스토리텔링, 학습콘텐츠, 실생활 중심의 콘텐츠가 가장 선호되고 있음을 분석하였다. 이를 토대로 본 연구에서는 인문 계열의 수학 학습 과정에서 다루는 잘 알려진 문제, 반복 패턴에 기반한 동작 시뮬레이션, 여러 객체 간 상호 동작과 동기화를 설정하는 애니메이션과 역할 게임 등의 장르를 활용하였다[10],[22]. 일상의 문제를 컴퓨팅 환경으로 전환하여 표현할 때에는 순서도, NS차트, 동작흐름도 등 다이어그램을 활용하여 SW 해결안의 내용과 흐름을 시각화하여 명시적으로 표현하는 다이어그래밍 활동을 진행한다.

최종 단계로 일상 속 문제를 스스로 발굴하고 해석하여 컴퓨팅 환경에서 해결하는 프로젝트를 완성한다.

위에서 기술한 각 단계에 맞춰 실제 SW 프로그래밍 교육과정과 콘텐츠 모형을 적용할 수 있는 주차별 SW 교수학습 모델을 표2와 같이 구성하였다.

표 2. 비전공자 SW 프로그래밍 교육의 주차별 교수학습 모델
Table 2. Teaching-learning model of SW programming education for non-Majors

week	Goal	Content	method
1	① transforming real problems into computational problems	computer architecture software operation	theory lecture
2		information presentation data structure	
3		concept of procedural solution (algorithm) expression of algorithm - flowchart/NSchart - summing of (1~10) - exchanging 2 numbers - introducing myself	
4	② visualization of SW solution	solution design(flowchart) - odd/even count - multiplication table	theory lecture programming
5		solution design(flowchart) - Factorial, Fibonacci	
6		solution design(flowchart) - maximum/minimum	
7		programming tool - Scratch (EPL)	
8		application - Animation (object interaction)	individual project
9		application - Fractal patterns - role playing (simulation)	
10	③ communication and collaboration based on visualized SW solution	application - Shortest Path	
11		application - Ordering Coffee	
12		project (design)	team project
13		project (implementation)	
14		project (presentation/evaluation)	

1~3주차에는 실세계와 컴퓨팅 환경의 차이를 이해하기 위한 기간으로 컴퓨터의 기본 구조와 소프트웨어의 동작을 학습하고 특히 컴퓨터와 소프트웨어에서 정보를 다루는 방식을 학습한다.

4~7주차에는 주어진 문제의 해결안을 순서도 혹은 NS차트를 활용하여 절차적으로 표현하는 방법을 학습한다. 이 과정은 컴퓨터의 동작을 시뮬레이션하는 훈련과 함께 평소에 사람이 직관적 혹은 통합적으로 풀어내던 문제해결 과정을 분해하고 상세하게 전개하는 과정을 익히게 된다. 이 단계까지는 이론적인 부분을 동영상이나 강의 자료로 미리 학습한

후 수업 시간에는 질의응답과 실습에 집중하도록 운영한다.

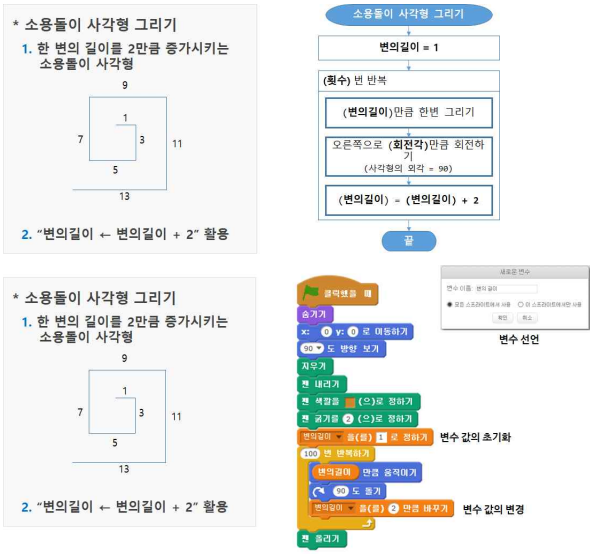

그림 4. 다이어그래밍과 비주얼 프로그래밍을 통한 문제해결
Fig. 4. Problem solving based on diagramming and visual programming

8주차 이후에는 다이어그래밍 방식의 해법 설계 작업과 스크래치를 활용한 비주얼 프로그래밍으로 개인 과제 혹은 팀 과제를 수행한다. 코딩을 시작하기 전에 반드시 다이어그램으로 해결안을 시각화하고 직접 팀원들에게 설명하도록 하는 과정이 매우 중요한데 이 정책은 학생들이 문제해결 과정이 아닌 코딩 자체에 지나치게 매몰되지 않도록 조정하기 위한 장치이다. 또한 이 과정을 통해 교수자가 학생들의 학습 수준을 확인함으로써 피드백을 제공하고 미션의 난이도를 조정할 수 있다. 12주차 이후에는 팀 단위로 자유 과제를 진행하며 각 팀에서 직접 아이디어를 조율하고 SW의 설계, 개발, 발표, 공감까지 SW 개발 과정 전체를 진행하게 한다. 이 과정에서 학생들은 스스로 문제 분해를 통해 역할을 분담하고 각자 담당한 미션의 해법을 시각적으로 표현한 다이어그램으로 소통하고 협업하는 방법을 습득할 수 있다. 이와 같은 자발적 문제 분석과 자유로운 의사소통, 협의가 가능한 환경에서의 도전 과제를 해결하는 활동이 창의성과 문제해결력 향상에 긍정적인 효과를 나타내는 것으로 알려져 있다[23].

Ⅳ. 적용 사례 및 결과

4-1 비전공자 SW 프로그래밍 교육과정과 콘텐츠 모형의 적용

C 대학은 비이공계열인 SW 비전공 신입생을 대상으로 비주얼 프로그래밍을 활용하여 SW적 문제해결 과정을 습득하고 컴퓨팅 사고력을 기를 수 있는 SW 기초 교과목을 운영한다. 시범적으로 경영학부 학생을 39명을 대상으로 본 연구에서 제안한 SW 프로그래밍 교육과정과 콘텐츠를 적용한 수업을 진행하였다.

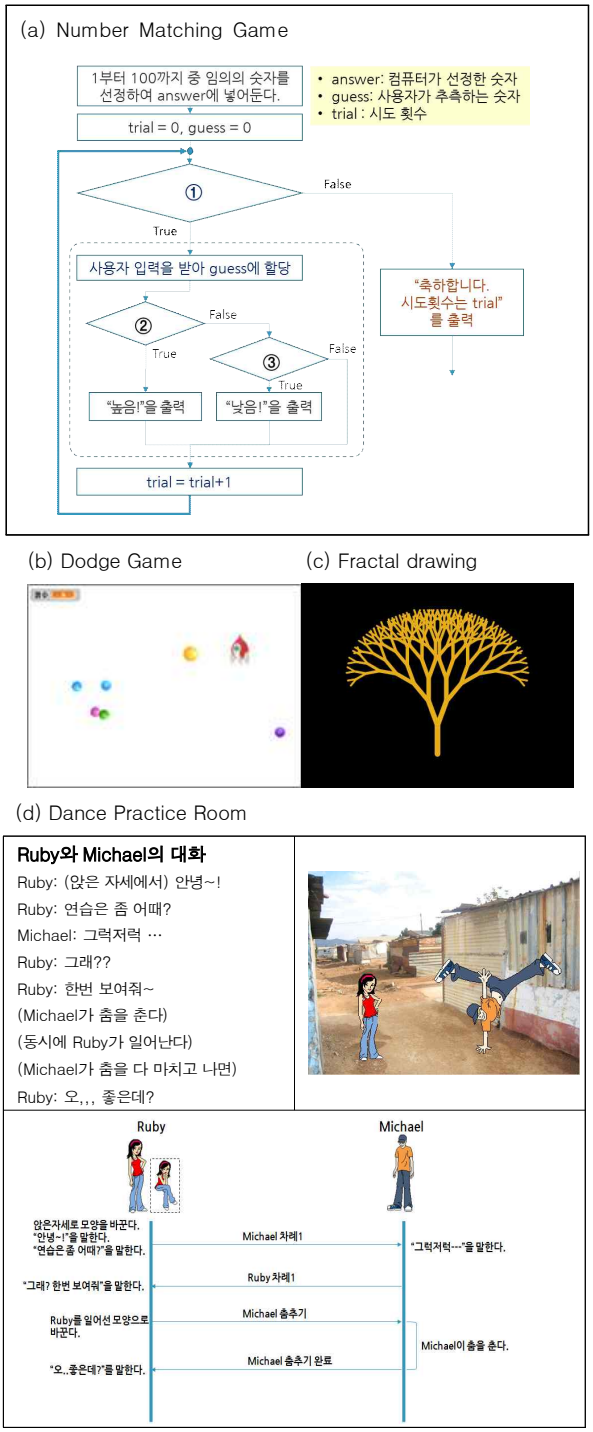

그림 5. 비전공자를 위한 SW 프로그래밍 교육 콘텐츠
Fig. 5. Contents for SW Programming education for non-majors

그림5의 (a)는 수업 중 상대방이 생각한 숫자를 스무고개를 통해 맞추는 게임을 순서도를 활용해 절차적으로 표현하

도록 실습한 내용이다. 그림5의 (b)와 (c)는 우주선 맞추기 게임과 프랙탈 나무 그림을 스크래치 프로그래밍으로 구현한 결과이다. 그림 5의 (d)는 여러 인물이 등장하는 시나리오를 제시하고 이를 절차적 다이어그램으로 설계한 결과이다.

4-2 해결안의 시각적 표현 기반 SW 프로그래밍 교육의 효과

SW 프로그래밍 수업에 앞서 학습자의 SW적 경험의 특이점을 확인하기 위해 SW 학습 경험을 설문하였다. 그림 6에 따르면 약 11%의 학생이 컴퓨팅 사고를 접한 적이 있으며 약 6%의 학생이 프로그래밍 언어를 학습한 적이 있는 것으로 나타났다. 프로그래밍 학습 경험과 컴퓨팅 사고에 대한 인지도에 나타나는 불일치는 4차 산업혁명 시대의 SW 중심 사회에 대한 홍보가 많이 되어 있으나 실제로 프로그래밍 언어를 활용하는 SW 교육은 충분히 확산되지 않았음을 보여 준다.

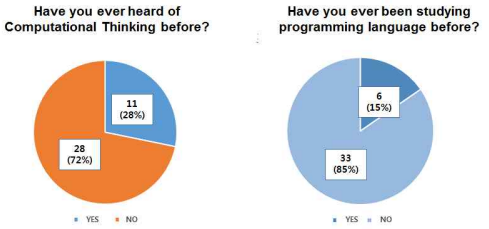

그림 6. SW 학습 경험에 관한 사전 조사
Fig. 6. Pre-survey on SW learning experience

SW 프로그래밍 교육과정을 마친 후 학습자를 대상으로 수업 활동 중 콘텐츠 선호도와 SW에 관한 인식 변화를 조사하였다. 그림 7은 수업 중 흥미로운 활동을 조사한 결과이며 '게임의 원리를 구현하는 작업'을 72%로 가장 많이 선택하였다. 또한 33%의 학생이 '순서도를 활용한 문제해법을 설계하는 과정'을 선택하여 생각에 머물던 아이디어와 해결방법을 체계화하고 도식화하는 과정에 흥미를 느낀 것을 확인할 수 있다. 이는 SW 프로그래밍 학습이 학생들의 SW에 대한 효능감을 높였음을 나타낸다.

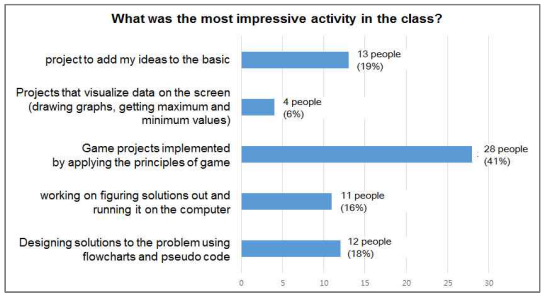

그림 7. SW 프로그래밍 활동에 관한 선호도
Fig. 7. SW programming activities preference in SW programming education

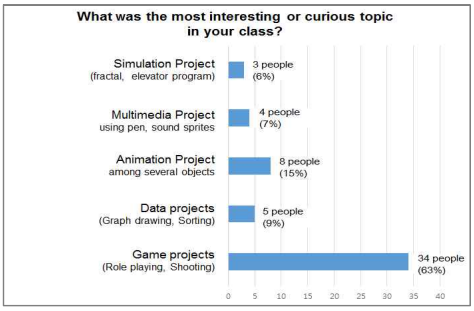

그림 8. SW 프로그래밍 콘텐츠에 관한 선호도
Fig. 8. SW content preference in SW programming education

다음으로 그림 8은 SW 프로그래밍 수업 중 흥미를 느꼈던 콘텐츠에 관한 질문에는 압도적으로 게임 개발 프로젝트를 선택하였음을 보여 준다. 이는 신입생들이 전공 공부를 본격적으로 시작하기 전으로 전공 계열의 특성보다 개인적으로 즐겼던 프로그램을 직접 개발할 수 있다는 점에서 더욱 몰입할 수 있었던 것으로 추정할 수 있다.

그림 9에서 알 수 있듯이 교과 과정을 이수하면서 SW에 관한 인식의 변화를 묻는 질문에는 'SW의 동작을 이해하게 되었다'는 응답이 61%로 가장 높았다. 이는 수강생들이 사용자 입장에서 SW를 바라보던 관점으로부터 SW 프로그래밍 교육을 통해 컴퓨팅 환경에서 동작하는 절차로서 SW를 인지하게 되었다는 것을 의미한다. 특히, 10%의 학생은 '개발자 입장으로 SW를 바라보게 되었다'고 응답하여 SW에 대한 인식이 확장되었음을 알 수 있다.

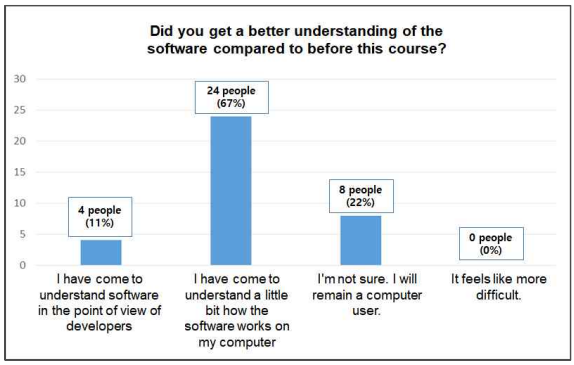

그림 9. SW 프로그래밍 교육 이후 학습자의 SW 이해도
Fig. 9. Degree of SW understanding after SW programming education

이번 연구에서는 SW 프로그래밍 교육과정을 실제 비전공자 대상의 SW 교과목에 적용하였을 때 SW 교육목표 달성에 필요한 학습자의 흥미유발 및 효능감을 중심으로 평가하였다. 향후에는 프로그래밍 언어 중심의 기존의 SW 교육과정을 적용한 대조군과의 비교 실험을 진행하여 본 논문에서 제안한 비전공자 SW 프로그래밍 교육과정의 적정성과 함께 해결안의 시각화 표현을 중심으로 한 교육 콘텐츠 개발 모형이 SW

적 문제해결을 위한 협업 역량에 미치는 효과를 측정할 필요가 있다.

Ⅴ. 결 론

최근 사회 모든 분야에서 SW 역량이 강조됨에 따라 SW 교육의 대상과 범위는 빠르게 확대되고 있으나 대학에서 비전공자 대상의 SW 교육은 전공에 따른 다양한 학습 배경과 진로에 비해 획일적이고 최적화되지 못한 것이 현실이다.

이에 본 연구에서는 미래사회를 대비하여 비전공자에게 필요한 SW 역량을 전공 분야의 전문성을 바탕으로 소프트 콘텐츠를 설계할 수 있고 SW 전문가와 원활한 소통과 협업을 통해 구현할 수 있는 능력이라고 보고, 이 역량을 강화하기 위한 비전공자 SW 프로그래밍 교육과정과 콘텐츠 개발모형을 탐색하였다.

비전공자를 위한 SW 프로그래밍 교육과정은 'SW적 문제해결을 위한 시각적 프로그래밍 역량 함양'을 목표로 하며, 수행 목표 항목으로 '일상의 문제를 컴퓨팅 환경의 문제로 재구성하기', 'SW 해결안의 시각적 표현', '시각화된 SW 해결안으로 소통, 협업하기'를 함께 제시한다. 첫 번째 단계에서 SW로 제어하는 컴퓨팅 기기의 기본적인 동작을 이해한다. 두 번째 단계는 일상의 문제를 컴퓨팅 세계에서 동작할 수 있는 문제로 변형한다. 이 과정에서 필요한 데이터를 특정하고, 동작의 조건분기와 반복을 활용하여 해법을 절차적으로 표현하는 훈련을 한다. 이때 해법의 코딩보다 내부 알고리즘을 다이어그램으로 명확하게 시각적으로 표현하는 활동에 중점을 둔다. 마지막 단계에서는 문제의 발견부터 해석, 설계, 해결안의 구현까지 완성하는 SW 제작 활동을 한다. 이 때 팀 내 원활한 협업을 위해 다이어그램으로 시각화한 해결안을 활용해 소통하도록 유도한다.

SW 교육은 컴퓨터 과학 이론뿐만 아니라 실제로 결과물을 제작하는 공학적 요소를 포함한다[16]. 따라서 문제를 해결할 SW를 실제로 설계, 구현하고, 나아가 공유, 공감하는 과정이 동반되어야 한다. 이를 만족하기 위해 교육과정 전반부에 일상의 문제를 컴퓨팅 문제로 재구성할 수 있도록 컴퓨터의 기본 구조와 동작, SW적 문제해결 과정을 학습한다. 중반부에는 애니메이션, 게임, 학습 콘텐츠, 시뮬레이션 등 다양한 주제의 문제 해결 알고리즘을 다이어그램으로 표현하는 작업을 반복함으로써 SW 해결안을 시각화하는 역량을 강화한다. 후반부에는 직접 아이디어를 발굴하고 이를 구현할 SW의 설계와 개발, 발표를 포함한 팀별 과제를 수행한다. 모든 미션은 학습자 스스로 문제를 해석하고 적절한 해결안을 시각화(visualization)한 후 직접 설명하도록 하였다.

비전공자 SW 프로그래밍 교육과정의 효과를 확인하게 위해 C 대학교의 경영학부 신입생 39명 대상의 SW 교육에 실제 적용하였다. 사전 설문 조사에서 11%의 학생은 컴퓨팅 사고력이라는 용어를 접한 적이 있으나 6%의 학생만이 SW 프로그래밍을 학습한 적이 있다고 응답하였다. 즉, 실험군의 학생은 프로그래밍 경험이 거의 없는 초보 개발자이다. 한 학기 동안 SW 프로그래밍 수업을 진행한 후 SW에 대한 인식 변화를 조사하였는데, 약 72%의 학생이 SW의 동작을 잘 이해할 수 있게 되거나 개발자 입장에서 SW를 바라보게 되었다고 응답하여 SW에 대한 자신감이 상승한 것을 알 수 있었다.

본 연구에서 제시한 비전공자 SW 프로그래밍 교육과정과 콘텐츠 모형을 통해 비전공자가 실전적인 SW 학습을 진행함으로써 각자의 전문 분야에서 사회가 기대하는 융합적 SW 역량의 토대를 마련할 수 있을 것이다. 향후, SW 프로그래밍 교육의 콘텐츠 모형을 다양한 전공계열로 확대 적용하여 전공 분야별 학습자의 특성에 따른 학습 성취도와 효과의 차이를 분석하고 이를 기반으로 비전공자를 위한 SW 학습 모델을 강화할 계획이다.

참고문헌

[1] SPRi:Software Policy & Research Institute. SW-centric society that the Fourth Industrial Revolution will bring [Internet]. Available: https://spri.kr/posts/view/15300?code=column.

[2] ByungSuk Yang, the Role of Software in a Software-centric Society, Digieco Report, 2015. Available: http://www.digieco.co.kr/KTFront/report/report_issue_trend_list.action

[3] The Open University. Introduction to computational thinking [Internet]. Available: http://www.open.edu/openlearn/science-maths-technology/computing-and-ict/introduction-computational-thinking/content-section-0?intro=1.

[4] KyungKyu Kim, JongYun Lee, "Analysis of the Effectiveness of Computational Thinking-Based Programming Language", *The Journal of Korean Association of Computer Education,* Vol. 19, No. 1, pp. 27–39, 2016.

[5] Min Young Lee, Seok Ju Chun, "A Study on Improving Logical Thinking Ability of Elementary School Students with Entry and Scratch", *The Journal of Korea Elementary Education,* Vol. 28, No. 1, pp. 173–185, 2017.

[6] Hyun Jong Choe, "The Programming Education Framework for Programming Course in University", *The Journal of Korean Association of Computer Education,* Vol. 14, No. 1, pp. 69–79, 2011.

[7] JeYoon Park, "Direction and prospect of SW education in the 2015 revised curriculum", *KEDI journal of educational policy,* Vol. 42, No. 3, pp. 85–89, 2015.

[8] Soo Hwan Kim, "Analysis of Non-Computer Majors`

Difficulties in Computational Thinking Education", *The Journal of Korean Association of Computer Education,* Vol. 18, No. 3, pp. 49–57, 2015.

[9] Jung Sook Sung, Soo Hwan Kim, Hyeoncheol Kim, "Analysis of Art and Humanity Major Learners` Features in Programming Class", *The Journal of Korean Association of Computer Education,* Vol. 18, No. 3, pp. 25 –35, 2015.

[10] Soojin Jun , "Analysis of Research Trends and Learners` Preference for Subject Area of SW Education Content", *The Journal of Korean Association of Computer Education,* Vol. 20, No. 1, pp. 39–47, 2017.

[11] Wing, J. M., "Computational Thinking", *Communications of the ACM,* Vol. 49, No. 3, pp. 33–35, 2006.

[12] CSTA, *"A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee"*, October, 2003,

[13] The CSTA Standards Task Force, "CSTA K-12 Computer Science Standards", Revised 2011.

[14] Gover, S., Pea, R. , "Computational Thinking in K-12: Review of the State of the field.", *Educnational Researcher,* Vol. 42, No. 1, pp. 38-43, 2013.

[15] Jung Sook Sung, Hyeon Cheol 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.

[16] Tim Bell, Ian H. Witten and Mike Fellows, Computaer Science Unplugged, CS Unplugged: An enrichment and extension programme for primary-aged students [Internet]. Available: http://csunplugged.org/

[17] Kanis Charntaweekhun, Somkiat Wangsiripitak, "Visual Programming using Flowchart", *International Symposium on Communications and Information Technologies,* pp. 1062-1065, 2006

[18] Sangjin An, Youngmin Seo, Youngjun Lee, "A Review and Synthesis of Research in Educational Programming Language", *The Journal of Korea Society Association of Computer Education,* Vol. 20, No. 1, pp. 139–142, 2012.

[19] SungWon Kang, "Directions for the Secondary Education to Nurture Software Development Capability", *Communications of the Korean Institute of Information Scientists and Engineers ,* Vol. 35, No. 4, pp. 34–43, 2017.

[20] Niko Myller, Roman Bednarik, Erkki Sutinen, Mordechai Ben-Ari, "Extending the Engagement Taxonomy : Software Visualization and Collaborative Learning", *Proceedings of the 37th SIGCSE technical symposium on Computer science education,* Vol. 9, No. 1, pp. 1–27, 2009.

[21] Jin-Dong Kim, Gwon-Woo Yang, "The Effect of Algorithm Learning in Real Life Case on Logical Thinking Ability", *The Journal of Korean Association of information Education,* Vol. 14, No. 4, pp. 555–560, 2010.

[22] KyungSun Oh, SeongJin Ahn, "A study on development of educational contents about computational thinking", *The Journal of Korean Association of Computer Education,* Vol. 19, No. 2, pp. 11–20, 2016.

[23] Si-Hoon Lee, Jeong-Hye Han, "Analysis on Creativity and Solving-Problem Ability with Hackathon-based Elementary SW Education", *The Journal of Digial Content Society,* Vol. 18, No. 5, pp. 995–1000, Aug. 2017.

이민정(Minjeong Lee)

1996년 : KAIST (공학석사)

1996년～2000년: ㈜ LG전자 LG종합기술원
2000년～2010년: ㈜ 아이에이
2011년～2015년: ㈜ 삼성전자 소프트웨어센터
2016년～현 재: 중앙대학교 컴퓨터공학부 조교수
※관심분야 : SW교육, 기계학습, 플랫폼, 에듀테크, AR/VR등