

컴퓨팅 사고 향상을 위한 UMC 수업전략의 개발과 적용

전수진 · 한선관

경인교육대학교 컴퓨터교육과

요 약

본 연구에서는 SW교육에서 Computational Thinking의 향상을 위한 놀이활동 중심의 UMC(Use-Modify-Create) 전략을 개발하였다. UMC 수업 전략은 학생들이 예제 알고리즘을 분석하고 이를 수정, 확장하여 새로운 SW를 개발하는 모형이다. 먼저 예비교사를 대상으로 1학기 동안 UMC 수업을 적용하였다. UMC전략에 대한 학생들의 self-CT 향상에 대한 t-검사 결과, UMC 수업전략이 CT개념과 실습능력에 도움을 주었다. 또한 Modify 단계와 직접교수식 단계보다 Use단계와 Create 단계의 탐색과 자기주도적 개발과정이 CT능력 향상에 도움을 주고 있음을 알 수 있었다.

키워드 : 컴퓨팅 사고, UMC교수학습전략, 소프트웨어 교육

Development of UMC Teaching and Learning Strategy for Computational Thinking

Soojin Jeon · Seonkwan Han

Dept. of Computer Education, Gyeong-in National University of Education

ABSTRACT

We have developed a UMC (Use-Modify-Create) strategy for the improvement of computational thinking in SW education. UMC model is an instructional strategy that students analyze examples of algorithms, modify-expand it and to develop new SW. First, we applied the UMC classes during 1 semester targeting pre-service teachers. To explore the effectiveness of the proposed model, we conducted a t-test for improving students' self-CT. Through a result, we found that is the UMC teaching-learning strategy helped the CT concepts and practice skills. In addition, we could see the Use and Create steps help to improve the ability of students' CT than directly teaching and Modify steps.

Keywords : Computational Thinking, UMC Teaching-Learning Model, SW Education

교신저자 : 한선관(경인교육대학교 컴퓨터교육과)
논문투고 : 2016-02-27
논문심사 : 2016-03-02
심사완료 : 2016-04-14

1. 서론

교육부와 미래창조과학부는 최근 SW교육을 강화하기 위해 초등학교에서는 실과 교과에서 17시간 교육을 포함하고 중학교에서는 ‘정보’ 과목을 의무화하여 컴퓨팅 사고(Computational Thinking: CT)를 신장시키기 위해 34시간 교육하며, 고등학교에서는 ‘정보’ 과목을 일반선택으로 전환하여 다양한 분야와 융합한 알고리즘의 설계 및 프로그램 개발 중심의 교육을 실시하도록 하였다.

한편, SW교육에서 컴퓨팅 사고력의 중요성이 여러 학자들에 의해 주장되고 있다[2][3][5][7][8][9]. 특히 Wing(2006)은 컴퓨팅 사고력이 기존의 3Rs 기본 소양 교육과 마찬가지로 21세기를 살아가는 모든 사람들이 갖추어야 할 기본 사고 능력이라고 주장하였다[2].

그러나 이러한 개념은 초보학습자들에게 매우 생소하고 어렵기 때문에 2015 개정 교육과정에서 적용되는 2018년에 SW교육이 학교 현장에 조기 정착되려면 보다 효과적인 SW교육의 교수학습방법을 개발하여 보급하는 것이 중요하다. 컴퓨팅 사고력을 중점에 둔 선행 교수학습 설계 및 모형 개발 관련 연구는 몇 가지 진행되었으나 초중등 현장에서 다양하게 활용하기에는 부족한 상황이다.

따라서 본 연구에서 몇몇 연구자들에 의해 최근 제시되고 있는 UMC 모델을 분석하여 예비교사들에게 적용하고 그 전략의 초중등 수업에서의 적용 가능성을 검증하는 데 목적을 두고 있다.

2. 이론적 배경

2.1 CT 향상을 위한 교수학습방법

최근 해외뿐 아니라 국내에서도 SW교육에 있어서 CT 역량을 향상시키기 위한 교수학습방법에 관한 연구가 일부 진행되어 왔다.

이은경(2009)은 CT 능력 향상 지원을 위해 교육용 로봇을 활용한 프로그래밍 교수학습 모형을 개발하였다. 교수학습 단계는 과제 확인, 지원, 과제 해결, 통합의 4단계로 이루어졌다[1].

Brennen외 2인(2014)은 ‘창의 컴퓨팅 가이드북’을 통해 DBL 교육 프로그램을 제안하였다[4]. 창의 컴퓨팅이란 창의성, 역량의 강화, 컴퓨팅에 관련된 것으로, 문제를 해결하기 위해 디자인, 컴퓨팅, 융합의 과정을 거쳐 창의적으로 문제를 해결할 수 있는 소프트웨어를 산출하는 활동이라고 정의하였다. 이러한 창의 컴퓨팅 교육 방법은 컴퓨팅 사고를 바탕으로 ‘디자인하기’, ‘개인화하기’, ‘협력하기’, ‘되돌아보기’의 네 단계로 되어 있다[4].

김수환과 한선관(2012)은 CT를 향상시키기 위한 디자인 기반 학습을 제안하였다. 이 연구에서 디자인 기반 학습은 self-프로그래밍 능력과 흥미도를 긍정적으로 변화시켰고 자신의 아이디어를 창작물로 제작하고 의미 있는 결과물을 만드는 교육으로써 효과를 가져왔음을 밝혔다[8].

채경진 외(2012)는 발견학습을 기반으로 하여, 프로그래밍 교육을 할 수 있는 수업을 설계하여 교육현장에서 초등학교 학생들에게 적용할 프로그래밍 교육에 대한 방법적인 모형을 제안하였다. 이는 CT를 언급하지는 않았으나 학생들의 자기주도성과 사고력 중심의 교수학습 방법임을 알 수 있다[6].

2.2 창의컴퓨팅 교수학습방법

<Table 1> CT Framework of MIT[5]

Category	Concept
CT Concept	sequence, loops, parallelism, events, conditionals, operators, data
CT Practice	experimenting and iterating testing and debugging reusing and remixing abstracting and modularizing
CT Perspective	expressing, connecting, questioning

MIT 대학의 미디어랩은 ‘창의 컴퓨팅 가이드북’을 통해 개념, 실습, 관점의 집합으로써의 Computational Thinking에 대한 프레임워크를 컴퓨팅사고의 개념(CT Concept), 컴퓨팅사고의 실습(CT Practice), 컴퓨팅사고의 관점(CT Perspective)로 나누어 제시하였다[5].

<Table 1>에서 컴퓨팅사고의 개념(CT Concept)은 시퀀스, 반복, 병렬처리, 이벤트, 조건, 연산, 데이터가 있다. 컴퓨팅사고의 실습(CT Practice)은 실험하기와 반

복하기, 테스트와 디버깅, 재사용과 재구성, 추상화와 모듈화가 있다. 컴퓨팅사고의 관점(CT Perspective)은 표현하기, 연결하기, 질문하기가 있다. 본 연구에서는 학생들로 하여금 자기평가를 통해 컴퓨팅사고의 개념(CT Concept)과 컴퓨팅사고의 실습(CT Practice)을 대상으로 평가하여 분석하였다.

Allan(2010)은 로봇 교육에 있어서 CT를 향상시키기 위한 UMC모형을 제시하였다. 이 모델에서 Use 단계는 학생들이 인터페이스와 도구를 포함한 기술과 다른 사람이 만든 여러 종류의 산출물들을 사용하는 방법을 배우는 것이라고 했다. 이 단계에서는 CT와 관련된 더 높은 수준을 위한 기초를 다지게 된다. Modify 단계는 기존의 프로그램이나 프로젝트를 수정함으로써 실험한다. 이 단계에서 학생들은 다른 결과들을 가져오는 매커니즘을 통제할 수 있는 방법들을 이해한다. Create 단계에서 학생들은 자신의 프로젝트를 만들기 위해 CT 기술을 적용하며 이러한 작업은 추상화와 자동화의 수준을 높여줄 것이라며 CT향상을 위한 로봇 교육에서의 UMC 모델을 제시했다[10].

한선관(2015)은 창의컴퓨팅 이슈리포트에서 창의 컴퓨팅을 학교 수업에 적용하기 위한 교수학습 모형을 U-M-C(Use, Modify, Create), D-D-C(discovery, Design, Create), D-C-C(Design, Computing, Convergence) 모델 3가지로 제안하였다[9]. U-M-C 모델은 초급 수준의 학생들이 간단한 놀이와 체험을 통하여 컴퓨팅에 대한 흥미를 갖게 하고 반복된 체험에 따라 기능을 수정·보완하여 확장시키는 활동을 하게 하는 것이다. D-D-C 모델은 주어진 문제의 문제점을 발견하고 수정하면서 기능, 원리, 개념을 익히고 디자인 사고력을 기반으로 한 설계 과정을 거쳐 구현하는 방법이다. D-C-C 모델은 설계 후 구현과정에서 다양한 컴퓨팅 도구를 활용하여 타 학문과 융합하는 난이도 높은 과정을 위한 모델이다. 이렇듯 충분한 경험을 하고 난 뒤에 배운 것을 토대로 프로그래밍을 하는 활동으로 발전시키는 전략을 사용하였다.

또한, 한국교육학술정보원의 SW교육 교수학습 모형 개발 연구(2016)에서는 한선관의 연구[9]를 발전시켜 재구성중심모형을 제안하였다[3]. 재구성중심모형은 UMC(Use(놀이)-Modify(수정)-reCreate(재구성))모형으로 동기유발로 놀이를 통해 배우고자 하는 학습 모듈을 학

생들이 탐색하고 사전에 준비된 모듈의 수정과정을 통하여 기능과 개념을 이해한다. 놀이 활동과 수정활동과 연계된 일련의 재구성 활동을 통해 학생들이 컴퓨팅 사고의 전반적인 능력을 이해하게 된다.

이러한 선행 연구들은 수업에 따른 경험적 사례로 제시된 것이며 국외 프로그래밍 교육의 사례로 2018년 정식교과가 되는 SW 교육의 방법을 기반으로 하는 한국 실정에 맞는 재검증이 필요하다.

3. 연구 내용 및 방법

3.1 교수학습 설계

본 연구에서는 창의 컴퓨팅 교수학습 방법들에서 제시되고 있는 UMC 모형의 공통된 특징들을 분석하고 구체화하여 예비교사에게 적용해 봄으로써 그 효과성을 검증하기 위한 교수학습을 설계하였다. 이러한 UMC 교수학습 단계에 따른 주요학습 방법과 세부 전략은 <Table 2>와 같다.

Use(탐색) 단계에서는 교수자에 의해 주어진 프로젝트나 스크립트를 학생들이 놀면서 실행해보고 스크립트를 탐구 분석해 봄으로써 프로젝트의 스토리, 객체, 스크립트의 논리 등의 개념을 발견하게 된다.

<Table 2> UMC Model with Inquiry-based Learning

Step	Strategy	Contents
Use	Function understand, analyze, explore, implement,	<ul style="list-style-type: none"> • With simple games theme, the teacher explains the main functions and scripts or students run the project and navigate directly to watching play. • Learners analyzes and notes the logic of the given script. • Writing a script meaning, representing a picture, the object analysis, story writing, configuration representation of the screen, displaying the movement route
Modify	Design, add, modify,	<ul style="list-style-type: none"> • Simply, the students provide additional ideas in the game or design to expand the contents(Adding and editing new

		<ul style="list-style-type: none"> sprites, adding a variable, expanding stage) • Students put together two or more projects and add some functionality or design of a project to another project. • Visualizing the story, adding objects to add functionality, change existing game rules, etc. • Design elements: subject, story, screen configuration, the required object, the object-specific detailed implementation plan
Create	Implementation, sharing feedback	<ul style="list-style-type: none"> • The students design and implement into the existing game content and additional conditions with programming languages. • Students present their work and receive feedback. Students should undergo to repeat the design and implementation through the debugging process. • Implementing alone, with a partner to implement, paired programming

이를 위해서 실제 수업에서는 사전에 핵심 기능이 포함되어 있으며 완성되어 실행 가능하지만 복잡하지 않고 간단한 프로젝트를 제시해 주었다. 학생들은 제시된 프로젝트를 실행해보고 가지고 놀아보면서 전체적인 프로젝트의 스토리를 이해한다. 이 과정에서 학생들은 문제를 분해해 보며 알고리즘의 절차와 논리를 이해한다.

Modify(수정 및 보완 설계) 단계는 앞 단계에서 탐색한 프로젝트를 바탕으로 학생들 스스로 새로운 계획을 추가해 본다. 스토리를 확장하여 객체를 추가하거나 기존의 객체에 스크립트를 추가해 본다. 이 과정에서 학생들은 자신들이 계획한 내용을 설계함으로써 문제를 재구성하고 추상화하는 과정을 경험한다.

Create(개발) 단계는 자동화를 구체화하는 단계이다. 설계된 내용을 활용하여 코딩으로 산출물을 만든다. 이때 다양한 방법으로 작품을 공유하고 발표함으로써 피드백을 주고받아 서로의 프로젝트를 탐구하고 재설계와 재구현을 반복함으로써 확장시키고 디버깅하는 과정을 돕는다.

<Table 3> Curriculum of using UMC strategy

No	Subject	UMC Strategy
1	Scratch Overview, Creating a fishbowl	U: Listening and writing on comprehension and explanation of the instructor D: Describe project tasks, project implementation and sharing
2	Dance party and fireworks	U: Scripts analysis of each project, to analyze objects M: Designing to reconstruct the two projects C: Implementing, to share
3	Wizard Star	U: Memo to the logic of the script, writing story M: Adding Story design, adding objects C: Implementing, to feedback
4	Avoid rocket missile	U: Note to object analysis, view draw a path of movement of the object M: Add object design and game rules C: Implementing, to feedback
5	Word games	U: Structuring the logic of the script M: Expressing additional function and rule as a pseudo code C: Implementing, to feedback
6-8	Implementing Team project	M: Writing stories, designing objects, expressing a pseudo code C: Implementing
9-10	Feedback of Team project	C: Feedback and questions and answers

이러한 UMC 모형전략을 적용한 교육과정은 <Table 3>과 같이 설계하였다. 이러한 교육과정을 바탕으로 학생들은 매 주차에서 탐구, 수정 및 설계, 개발의 단계를 반복적으로 경험한다.

3.2 평가 및 분석 도구

UMC모형을 적용하여 그 수업의 효과성을 검증하기 위하여 다음과 같은 분석 도구를 사용하였다.

첫째, 수업 단계별 반응을 분석하였다. 수업단계별 반응은 수업 후 다음 두 가지 질문에 대해 설문을 통해 얻었다. 학생들은 매 수업 시간에 반복된 수업 단계들

을 상기하면서 프로그래밍의 숙달과 관련된 ‘CT Concept 능력’과 컴퓨팅 사고력과 관련된 ‘CT Practice 능력’의 향상에 도움이 되었다고 생각하는 단계에 대해 1~5까지 순위를 매기도록 하였다.

1. CT Concept 능력을 향상시키는 데 도움이 된 단계는?
2. CT Practice 능력을 향상시키는 데 도움이 된 단계는?

UMC 모형의 각 단계에 대한 효과를 효과적으로 분석하기 위해서 수업 단계는 <시범따라하기>, <Use(답색)>, <Modify(수정 및 설계)>, <Create(개발)> 단계로 제시하였다. <시범따라하기> 단계는 기존의 직접교수법 형식의 프로그래밍 수업에서 교수자의 시범을 따라하는 방식과 비교하기 위해 제시하였다. 대부분의 학생들은 그와 같은 이전에 프로그래밍 수업을 받았기 때문에 그때를 상기하여 질문에 응답하였다.

둘째, 사전-사후 self-CT의 향상도를 분석하였다. self-CT는 Brennan(2012)가 제시한 CT Concept과 CT Practice를 포함한 평가 도구의 자료를 참고하여 제시하였다. 이러한 사전 사후 데이터는 대응표본 t-검정을 통해 분석하였다.

3.3 연구 대상 및 절차

본 연구의 실험은 k교육대학교 학부 1학년생 63명을 대상으로 주당 2시간씩 10주 동안 총 20시간의 수업을 통해 이루어졌다. 실험 대상 학생들의 사전 프로그래밍 경험 수준에 대한 응답은 다음 <Table 4>와 같이 중간 정도의 경험 수준이 57.1%로 가장 많았다.

<Table 4> Prior programming experience level

level	number(n=63)	percentage(%)
high	16	25.4
mid	36	57.1
low	11	17.5
total	63	100.0

4. 연구결과

4.1 self-CT능력 사전사후 분석

먼저 UMC 전략이 학생들의 CT 능력 향상에 도움을 주었는지에 대해 사전 사후로 설문한 결과는 다음과 같다.

<Table 5> Pre-post tests of self-CT skill

Item(n=63)	Avg	Std	t	Sig	
self-CT skill	Pre	3.33	1.44	-5.683	.000***
	Post	4.11	1.00		

(p<0.001***)

<Table 5>에서 보는 바와 같이, 자신의 CT 능력 정도에 대한 질문에 사전에는 평균 3.33에서 사후에는 4.11로 상대적으로 높아졌으며 이것은 통계적으로 유의미한 차이를 보였다(p<0.001).

따라서 학생들이 본 수업을 통하여 스스로 인식한 CT 개념과 CT 실습 역량이 신장되었음을 알 수 있다.

4.2 CT 실습과 CT 관점의 수업단계별 반응 분석

UMC수업 전략에 따른 단계별 반응 분석 결과는 다음과 같다. 먼저, <Table 6>과 같이 자신의 CT 개념(Concept) 능력 향상에 도움을 주었던 수업 단계로 Use 단계와 Create 단계를 1순위로 선택한 학생이 33.4%와 31.8%로 가장 많았다. 이는 기존의 ‘시범따라하기’의 1순위 비율인 27.0%보다도 많이 나타났다. 또한, Modify 단계는 CT 개념(Concept) 능력향상에 도움이 되었다고 1순위로 응답한 학생은 22.2%였다.

<Table 6> Impact on CT Concept skill(%)

level	Demonstration	Use	Modify	Create
1st	27.0	33.4	22.2	31.8
2nd	34.9	33.3	9.5	23.8
3rd	23.8	17.5	23.8	20.6
4th	14.3	15.8	44.5	23.8
total	100	100	100	100

또한, <Table 7>과 같이 자신이 인식한 CT 실습(Practice) 능력 향상에 도움이 된 수업 단계로도 Use

단계와 Create 단계를 1순위로 선택한 학생이 39.7%와 31.8%로 가장 많았다. 이 두 가지 Use 단계와 Create 단계의 1순위 비율은 '시범따라하기' 방식의 1순위 비율인 22.2%보다도 더 많았다. Modify 단계는 CT 실습(Practice) 능력 향상에 도움이 되었다고 1순위로 응답한 학생은 23.8%였다.

<Table 7> Impact on CT practice skill(%)

level	Demonstration	Use	Modify	Create
1st	22.2	39.7	23.8	31.8
2nd	36.6	22.2	20.6	22.2
3rd	20.6	20.6	28.6	20.6
4th	20.6	17.5	27.0	25.4
Total	100	100	100	100

다른 단계들보다는 다소 낮은 순위로 응답을 보였으나 이 Modify 단계는 자신들의 CT 개념(Concept) 능력보다는 CT 실습(Practice) 능력 향상에 도움을 주었다고 응답한 학생들이 더 상위 수준에 많이 분포되어 있었다(CT 개념 능력 1~2순위 누적 백분율: 31.7%, CT 실습 능력 1~2순위 누적 백분율: 55.6%).

5. 결론

본 연구의 결론은 다음과 같다.

첫째, 학생들은 UMC수업모형에서 Use 단계와 Create 단계가 다른 단계들 보다 더 자신의 CT 개념(Concept) 능력과 CT 실습(Practice) 능력 향상에 도움을 주었다고 응답하였다. 교사의 강의식 설명보다도 자신이 직접 코드를 분석해보고 탐색하는 과정과 직접 구현해 보는 과정이 더 유익했다는 것을 보여준다.

둘째, Modify 단계는 자신들의 CT 개념(Concept) 능력보다는 CT 실습(Practice) 능력 향상에 더 도움을 주었다고 응답하였다. 이는 학생들이 Modify 단계에서 새로운 스토리나 객체를 추가해보고 설계하는 과정에서 논리를 추상화하고 자동화하는 과정을 구상하게 되기 때문인 것으로 보인다.

셋째, UMC 수업모형에서 학생들의 CT 능력과 관련된 실습(Practice) 능력과 CT 실습(Practice) 능력이 유

의미하게 향상되었다. 학생들이 직접 설계하고 구현해 보는 과정은 매우 중요하다. 하지만 이 결과는 초보 학습자들이 이해하기 쉽게 교수자가 스크립트의 논리를 설명해주는 것도 중요하지만 학습자가 직접 그 논리를 분석하고 새로운 형태로 그려보고 메모해 가는 과정은 학생들이 이후에 스스로 새로운 것을 설계하고 구현해 보는데 많은 도움을 받을 수 있음을 시사한다.

넷째, 예비교사들뿐만 아니라 대학교육에 있어서 기존의 교수자의 따라하기식 수업 방식에서 벗어나 사전에 교수자가 준비한 프로젝트를 탐색하고 분석해 보는 과정을 제공해주는 것이 학생들에게 매우 유익할 것이다. 따라서 교수자는 학생들이 스스로 탐색하고 분석할 수 있는 다양한 프로젝트와 교수자료를 준비해 주어야 할 필요가 있을 것이다.

참고문헌

- [1] E. K. Lee (2009). Robot Programming Teaching and Learning Model to Enhance Computational Thinking Ability, Korea National University of Education, Doctoral Thesis.
- [2] J. M. Wing (2008). Computational thinking and thinking about computing. *Philosophical transactions of the Royal Society A*, 366, 3717-3725.
- [3] J. S. Kim, S. K. Han, S. H. Kim, S. W. Jung, J. M. Yang, E. D. Jang, J. N. Kim (2016). Study on Development of Teaching and Learning Model for SW Education, Korea Education Development Institute & KERIS Research Report.
- [4] K. Brennan, C. Balch, M. Chung (2015). Creative Computing Guide Book, Harvard University.
- [5] K. Brennan, M. Resnick (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada. 2012(1), 1-25.
- [6] K. J. Chae, J. H. Kim, T. Y. Kim (2012). The Instructional Design of Programming based on Discovery Learning Model for Elementary School

Students. Proceeding of Conference of The Korean Association of Computer Education, 16(1), 52-56.

[7] M. Y. Ryu, S. K. Han (2015). Development of Computational Thinking-based Educational Program for SW Education. *Journal of The Korean Association of Information Education*, 19(1), 11-20.

[8] S. H. Kim, S. K. Han (2012). Design-Based Learning for Computational Thinking. *Journal of The Korean Association of Information Education*, 16(3), 319-326.

[9] S. K. Han (2015). Teaching and Learning Strategies for SW Education, Creative Computing Issue Report 2015-1, <http://computing.or.kr>

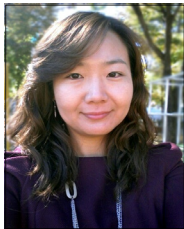
[10] W. Allan, B. Coulter, J. Denner, J. Erickson, I. Lee, J. Malyn-Smith, F. Martin (2010). Computational thinking for youth. White Paper for the ITEST Small Working Group on Computational Thinking (CT).



한 선 관

1991 경인교육대학교(교육학사)
 1995 인하대학교 교육대학원
 (컴퓨터교육학석사)
 2001 인하대학교 전자계산공학과
 (전산학 박사)
 2002~현재 경인교육대학교 컴퓨터교육과 교수
 관심분야: 지능형교수시스템, 초등정보교육, 컴퓨터과학 언플러그드, 게임중독치료, SW교육, 창의컴퓨팅교육, CT
 e-mail: han@gin.ac.kr

저자소개



전 수 진

2000 경인교육대학교(교육학사)
 2005 경인교육대학교 교육대학원
 (초등컴퓨터교육학 석사)
 2014 고려대학교 일반대학원
 (컴퓨터교육학 박사)
 2000~현재 초등학교 교사
 (현 부천부곡초등학교)
 관심분야: 초등정보교육, Computational 리터러시, CSCL, SW교육, 창의컴퓨팅교육, CT
 e-mail: soojin.jun@inc.korea.ac.kr

