

SW 교육에서의 컴퓨팅 사고력 신장을 위한 개발 중심 모형의 설계 및 효과

전수진

상미초등학교

요 약

본 연구의 목적은 소프트웨어 교육에서 학습자들의 컴퓨팅 사고력을 신장시키기 위한 개발 중심 모형(Discovery-Design-Development: DDD)의 효과를 검증하는 것이다. 개발 중심 모형은 탐구, 설계, 구현의 3단계를 거쳐 실생활 문제를 해결하기 위한 SW를 개발하는 과정을 학습하는 수업모형이다. 이러한 개발 중심 모형의 효과를 검증하기 위해 SW 교육 수업 경험이 없는 초급 수준의 G 대학교 1학년 학생을 대상으로 실험하였다. 그 결과 개발 중심 모형을 통해 SW 개발을 경험한 학생들의 SW 교육에 대한 학습 동기는 사전-사후 간 유지되었으며 특히 자신감 영역은 유의미하게 높아졌다. 또한, 학생들은 특히 설계(Design) 단계가 CT 능력 향상에 도움이 되었으며 개발 중심 모형은 문제분해와 알고리즘의 개념 이해에 상당한 도움이 된다고 응답하였다.

키워드 : SW 교육, 컴퓨팅 사고력, 개발 중심 모형, 교수학습방법, 수업 모형

Design and Effect of Development-Oriented Model for Developing Computing Thinking in SW Education

Soojin Jun

Sangmi Elementary School

ABSTRACT

The purpose of this study is to verify the effectiveness of the Development-oriented model (Discovery-Design-Development: DDD) for developing students' Computational Thinking skills in software education. DDD is a class model that learns the process of developing SW to solve real-life problems through three stages of Discovery, Design, and Development. In order to verify the effectiveness of the DDD, first grade students at G university who did not have SW education class were tested. As a result, the students who experienced SW development through DDD maintained the learning motivation of SW education before and after, especially the confidence area increased significantly. In addition, students also noted that the Design phase was particularly helpful in improving CT abilities and the DDD was a significant help in understanding of decomposition and algorithms.

Keywords : SW Education, Computational Thinking, Development-oriented Model, Teaching and learning method, Learning Model

1. 서론

4차 산업혁명의 도래로 인해 최근 모든 산업이 컴퓨팅의 영향력 하에 있게 됨에 따라 세계 각국에서는 소프트웨어(Software: SW) 교육의 중요성을 인식하고 국가 차원의 교육과정을 마련하여 SW 교육을 강화하고 있다[1][2][3]. 이러한 배경에는 여러 가지 문제를 컴퓨팅 관점에서 바라보고 효율적이고 해결할 수 있는 사고능력을 의미하는 컴퓨팅 사고력(Computational Thinking, CT)이 미래의 핵심 역량으로 강조되고 있기 때문이다. 뿐만 아니라 SW 원리에 대한 이해를 갖춘 인력 확보가 곧 국가경쟁력을 좌우할 것으로 예상되고 있다[6]. 이러한 시대적 변화에 따라 세계 여러 나라는 SW 교육을 위한 다양한 계획과 실천을 진행하고 있다[13].

우리나라의 경우 2015년 SW 교육운영지침과 2015년 개정 교육과정을 통해 SW 교육을 강조하기 시작하였고 그에 따라 2018년부터 중학교에 정보과목을 통한 SW 교육이 필수로 도입되며 2019년부터는 초등학교 5~6학년 학생들에게 SW 교육이 의무화된다[11][12]. 또한, 최근 SW 중점대학들을 통해 인문계 학생 등을 대상으로 대학 내 SW 기초교육을 확대하고 있는데 모든 비전공자에게 전공별 특성에 맞는 SW 기초 교육을 의무화하고 있다[10]. 이에 따라 현장 교사들 뿐 아니라 초중고 교사를 준비하는 교대 전체 학생 및 사범대 전공자들에게 이러한 새로운 교육과정에 맞춘 영력이 요구되고 있다[10].

이와 같이 SW 교육이 강조되고 있는 상황이지만 실제로 많은 학생들은 SW 학습을 어렵게 느끼고 있는 것이 현실이다[18]. 또한, 이러한 변화와 흐름에 비해 교육 현장에서는 SW 교육의 필요성이나 목적에는 동의하나 실제로 SW 교육 수업을 어떻게 설계하고 적용할 수 있을지에 대한 고민이 제시되고 있다[19].

이에 예비교사 및 비전공자를 위한 체계적인 CT기반 SW 교육이 필요하다. 특히 초보 학습자들에게는 SW 교육에 대한 학습동기의 유지가 매우 중요하다. 학습자들이 SW 교육 내용을 어려워하거나 습득하기 어려워한다면 학생들의 미래 사회의 기본 핵심 역량으로써의 역할을 하는 것이 어려울 것이다. 또한, 상급 학교에서의 학습이나 전문가 양성 과정에 도전하고자 하는 동기를 갖기 힘들 것이다.

그럼에도 불구하고 현재 CT를 중심으로 한 교수학습 방법에 대한 연구가 일부 진행되고 있으나 기존의 교수

학습방법이나 전략수준에 국한되어 있다[14][17][20]. 이에 실질적으로 SW 교육의 특성을 살린 교수학습방법에 대한 연구는 매우 부족하며 그 효과에 대한 검증이 충분히 이루어지지 못하고 있다.

이에 2016년에 한국교육학술정보원에서는 SW 개발을 위한 실습 중심의 교수학습 교수학습모형들을 개발하여 제시하였다[7]. 이 연구 결과 시연 중심 모형, 재구성 중심 모형, 개발 중심 모형, 디자인 중심 모형 등을 제시하였으며 일부 모형들에 대해 검증이 이루어지고 있다[9][15][16][19]. 그러나 아직 검증되지 못한 모형에 대해서는 여러 학교급 또는 대상에 따라 심체적인 적용과 함께 그 효과 및 모형에 대한 분석이 필요하다.

따라서 본 연구에서는 컴퓨터 비전공자인 1학년 대학생들을 대상으로 개발 중심 모형(Discovery-Design-Development: DDD)을 기반으로 한 SW 교육을 실시하여 그 효과성을 검증하고 CT 학습에 미치는 영향에 대해 분석하고자 한다. 본 연구의 구체적인 연구 문제는 다음과 같다.

첫째, DDD 모형을 적용한 SW 교육에 참여한 학생들의 학습동기가 향상 및 유지되었는가?

둘째, DDD 모형의 각 단계와 CT 향상에 미치는 영향은 어떠한가?

2. 관련연구

2.1 컴퓨팅 사고력과 SW 교수학습법

CT는 Wing(2006)에 의해 처음 제시되었고 그 중요성이 부각됨에 따라 많은 컴퓨터 과학자 및 교육자들에 의해 연구되어 왔다[5]. Wing(2006)은 CT가 컴퓨터 과학자의 문제해결 방법을 위한 사고 과정이며 미래에는 모든 사람들에게 필요한 역량이 될 것이라고 주장하였다[5]. 그는 추상화와 자동화가 CT의 하위 요소라고 소개하였다. 그 후 미국의 컴퓨터과학교사협회(CSTA), Google for Education, 영국 등에서 CT의 개념과 하위 요소의 조작적 정의를 세부적으로 정의하였다[1][2][3]. Google for Education에서는 이러한 CT의 구성요소는 초창기에 분해-패턴인식-추상화-알고리즘-자동화의 5단계로 제시되었고 그 후 더 세부적인 요소들이 정의되며 CT 교육에 대한 중요성이 부각되기 시작했다[5][2].

또한, SW 교육에서 학생들의 CT 신장과 더불어 보다 학교현장에 적용하기 쉬운 형태의 수업 모델의 필요가 높아지고 있다. SW 교육에서 학생들의 CT를 신장시키기 위해 프로젝트 학습법, 짝프로그래밍, 디자인 기반 학습법 등의 다양한 교수학습 방법과 전략들이 적용되어 연구되어 왔다[14][17][20]. 그러나 이러한 교수학습방법 및 전략은 기존의 보편적인 교수학습방법들을 SW 교육에 적용한 것이거나 일부 수업전략 형태에 국한되어 있어 SW 교육의 특성을 반영한 교수학습모형으로서는 부족할 수 있다. 이에 한국교육학술정보원은 2015년에 ‘SW 교육 교수학습 모형 개발 연구’에서는 SW 교육의 실습을 위한 교수학습 모형들을 개발하여 보급하였다[7].

이 연구에서는 시연-모방-제작의 단계로 이루어진 시연 중심 모형((Demonstration-Modeling-Making: DMM), 놀이-수정-재구성단계로 이루어진 재구성 중심 모형(Using-Modify-reCreate: UMC), 탐구-설계-개발의 단계로 이루어진 개발 중심 모형(Discovery-Design-Development: DDD), 요구분석-디자인-구현-공유의 단계로 이루어진 디자인 중심 모형(Need-Design-Implementation-Share: NDIS) 등을 제시하였다[7].

특히, 본 연구에서 사용한 DDD 모형은 탐구학습모형을 기반으로 하며 소프트웨어 공학적인 측면에서 SW 개발의 전 과정을 이해하는 모형이다. 따라서 학습자가 개발하고자하는 SW에 대한 탐구를 통해 기초 설계를 하며 SW 개발과정을 경험하게 하는 과정을 거친다[7]. 이러한 과정은 실제 컴퓨터 과학자들이 프로그램을 설계하고 구현하는 과정과 흡사하므로 학생들 또한 그 과정을 통해 컴퓨팅 사고과정을 경험하게 될 것이다.

2.2 CT기반 SW 교수학습 방법의 효과 연구

이와 같은 변화에 따라 학습자의 CT 역량을 향상시키기 위해 앞에서 제시한 SW 교육실습을 위한 다양한 교수학습방법들을 개발하고 검증하는 연구들이 다음과 같이 연구되고 있다.

전용주(2017)는 중학교 SW 수업에서 CT 기반 창의적 문제해결(CT-CPS) 수업모형을 제시하고 인지적·정의적 효과성을 분석하였다. 연구 결과 인지적 및 정의적 영역에 해당하는 대부분의 요소에서 통계적으로 유의미한 향상을 보였음을 검증하였다[19].

전수진과 한선관(2016)은 재구성 중심 모형(UMC)이 학습자의 CT 개념과 실습 능력에 도움을 주었음을 검증했다. 특히, Use 단계와 reCreate 단계의 탐색과 자기 주도적 개발과정이 학습자의 CT 능력 향상에 상당한 도움을 주었음을 밝혔다[15].

디자인 중심 모형(NDIS)에 대해서는 김미승(2017)은 NDIS 기반 ESD 프로그램을 개발 및 적용하여 초등학생들의 창의력과 컴퓨팅 사고력이 증가하였고 지속가능발전에 대한 인식의 변화를 확인하였다[9]. 또한, 전수진(2017)은 SW 교육에서 중급수준의 대학생들을 대상으로 디자인 중심 모형(NDIS)과 전통적인 프로젝트학습법을 비교하여 NDIS 모형 수업이 더 높은 CT 서술형 수행평가 점수를 보였음을 검증하고 컴퓨팅을 이용한 실생활 문제 해결에 대한 자신감과 CT 향상에 대해 긍정적인 인식을 보였음을 밝혔다[16].

이러한 연구들은 학습자의 CT 향상에 기여할 수 있는 새로운 교수학습방법을 개발하고 검증하여 학습자의 CT 신장에의 가능성을 보여주었다. 그러나 앞으로 SW 교육을 필수로 실시해야 하는 학교현장의 다양한 교육 여건과 학습자의 수준 등을 고려한 교수학습방법에 대한 다각도의 연구가 더욱 필요한 실정이다.

3. 연구 방법 및 절차

3.1 CT 기반 DDD 모형의 설계

본 연구에서는 기존 연구에서 제시된 DDD 모형에 CT 요소를 매칭 시켜 세부적인 수업 전략을 설계 하였다. 이러한 DDD 모형의 주요 학습방법과 해당 CT 요소는 <Table 1>과 같다.

<Table 1> Key learning method of DDD model

Steps	Key learning method	Major CT elements
Discovery	Understand the problem, analyzing, exploring	Decomposition
Design	Design, plan	Pattern recognition, Abstraction, Algorithms
Development	Implementation, sharing, output	Automation

<Table 2> Curriculum for SW education

Weeks	Subjects	Contents & DDD Strategies
1	CT concept	Concepts of Decomposition, Pattern recognition, Abstraction, Algorithm, Automation
2	Rabbit and Tortoise	Basic programming through storytelling: Sequence, Loops, Conditionals, Control, Event
3	Robotic vacuum	D1: Disassemble the moving stage of the cleaning robot D2: Create an algorithm to run the robots cleaning in order and repetition D3: Implementing Entry Programming
4	Milk check sheet	D1: How to record students who took milk D2: Algorithm expression by milk display abstraction and structuring D3: Implementing Entry Programming
5	Elected program	D1: Comparison and analysis of general voting process and electronic voting process D2: Voting data abstraction using variables, voting program algorithm representation D3: Implementing Entry Programming
6	Smart Fan	D1: Analysis of principle of automatic fan using sensor D2: Structuring and finding patterns of sensor-based fan operating principles, expressing algorithms D3: Implementing Entry Programming
7	Smart potted plants	D1: Analysis of plant condition changes according to amount of potted water D2: An abstraction of the operating conditions of the automatic water cycle pollen, expression of the water cycle algorithm D3: Implementing Entry Programming
8	Bean bag classifier	D1: Bean bag classification method analysis and problem decomposition D2: Finding key elements and abstraction of classifier operating conditions D3: Implementing Entry Programming
9	Project planning	- Students explore and design problem situations by themselves
10	Implementation and share	- Detailed design and implementation of the project (report generation) - Presentation and Feedback

D1: Discovery, D2: Design, D3: Development

탐구 단계에서의 주요학습방법은 문제이해, 분석, 탐색이며 CT 하위요소로는 문제분해 요소를 포함하여 지도하였다. 설계 단계에서는 설계 및 계획하기 활동을 통해 패턴인식, 추상화, 알고리즘 요소를 익히도록 하였다. 개발 단계에서는 구현, 공유, 개발, 산출활동을 통해 자동화를 완성하도록 하였다.

이와 같은 DDD 모형 설계를 바탕으로 <Table 2>와 같이 주당 1시간씩 10주 동안 총10시간의 수업 계획하여 실시하였다. 수업의 내용은 기존 개발된 교재[8]의 학습 내용을 DDD 모형에 맞추어 수업 설계하고 CT 관점으로 재구성하였다.

1주차~2주차에는 CT의 개념과 하위요소들을 강의하였고 ‘토끼와 거북이’ 스토리텔링 프로젝트를 통해 엔트리의 기본 기능을 익히게 하였다. 3주차~8주차에는 DDD 모형을 적용하기 위해 매주 1개씩 6개의 실생활 문제 중심의 주제로 구성하였다. 이에 ‘로봇 청소기’, ‘우유확인표’, ‘투표 프로그램’, ‘스마트 선풍기’, ‘자동 물주기 화분’, ‘콩주머니 분류기’의 주제를 CT 기반의 DDD 모형에 따라 지도하였다. 9주차~10주차에서는 ‘Smart School’이라는 큰 주제를 가지고 학생들이 직접 자신의 실생활문제를 찾아 탐색하고 설계하여 개발하도록 하였다. 이를 통해 학생들은 자신의 작품에 들어간 5가지 CT의 하위 요소를 기술하는 보고서와 함께 ‘우유배달로봇, 스마트 환기 시스템, 홍채인식 사물함 시스템, 스마트 데스크, 출석확인 시스템, 스마트 에어컨, 독서 기록판, 사칙연산로봇, 체점프로그램, 당번 정하기 프로그램, 스마트 분리수거’ 등의 엔트리 프로젝트를 제작하였다.

이러한 DDD 모형에 대한 세부 교수학습 지도방법은 다음과 같다.

첫째, 탐구 단계에서 사용되는 주제는 교사에 의해 제시된 자료를 사용하며 문제를 분해하여 이후의 설계와 개발 단계를 쉽게 달성할 수 있도록 안내한다. 또한, 수집된 데이터나 모듈 간의 패턴을 학생 스스로 탐구하도록 하여 설계 단계의 추상화 및 알고리즘의 핵심요소를 예상하도록 한다. 즉, <Table 3>에서 보는 바와 같이 탐구 단계에서는 문제 상황을 제시하고 문제해결의 목표 상태를 육하원칙에 따라 파악하고 콩주머니 분류 방법을 세부적으로 분해하여 분석하는 ‘문제분해’ 활동을 하도록 하였다.

<Table 3> Examples of CT in the Discovery step

CT	Project example (bean bag classifier[8])
Decomposition	

둘째, 설계 단계에서는 순서도 및 의사코드 등의 전통적인 설계 방법 뿐 아니라 다양한 공학 설계 방법도 사용할 수 있다. 학습자가 SW 구현 경험이 많을 경우에는 기능과 절차를 설계하는 등의 SW 공학적인 접근 방법을 사용할 수 있다. 개발을 위한 설계의 과정에서 패턴인식, 추상화, 알고리즘 등의 CT의 구성요소를 단계별로 진행하며 접근함으로써 구현을 용이하게 준비하도록 한다. 즉, <Table 4>에서 보는 바와 같이 설계 단계에서는 콩주머니 분류 패턴을 파악하여 조건에 따른 예상 상황을 트리형태로 간략히 추상화 하고 프로그램의 절차를 구조화하여 분석하도록 하며 최종 알고리즘을 의사코드 형태로 작성하도록 하였다.

<Table 4> Examples of CT in the Design step

Project example (bean bag classifier[8])		
Pattern recognition	Abstraction	Algorithms

셋째, 개발 단계에서는 개발과정을 학습자 중심으로 진행한다. 다만 개발에 사용되는 프로그래밍의 문법 등의 기술적 문제는 교사가 준비하고 조언하도록 한다. 개발 산출물은 수업시간을 고려하여 시뮬레이션이나 프로

토타입의 형태로도 할 수 있다. 개발 내용에 대한 공유와 평가는 학습자들 간의 피드백을 통한 자기반성의 관점에서 중요하다. 즉, <Table 5>에서 보는 바와 같이 설계 단계에서는 설계한 알고리즘에 따라 교육용 프로그래밍 언어인 엔트리 프로그램을 이용하여 구현하도록 하였다.

<Table 5> Examples of CT in the Development step

CT	Project example (bean bag classifier[8])
Automation	

3.2 연구 대상

본 연구의 실험은 G 대학교 1학년생 28명을 대상으로 실시되었다. 먼저, 학생들의 '컴퓨터 관련 수업 경험' 수준을 알아보기 위해 초중등학교와 학원에서의 교육기간을 모두 포함하여 선택하도록 하였으며, 5단계로 구분하여 1단계는 '관련 수업을 들어본 적 없다', 2단계는 '1학기 수강', 3단계는 '2학기 수강', 4단계는 '3학기 수강', 5단계는 '4학기 이상 수강'으로 구분 지었다. 또한, '프로그래밍 구현 자신감'도 5점 척도로 점수화 하여 평균을 산출하였다.

설문 결과 컴퓨터 관련 교육 경험은 <Table 6>에서 보는 바와 같이 평균 2.04단계(SD: 1.45)로 관련 수업을 들어본 적이 없는 학생들이 상당수 있는 것으로 나타났다. 또한, 프로그래밍 구현에 대한 자신감에 대해서는 평균 1.86으로 매우 낮게 나타난 것으로 나타났다. 이 결과로 본 연구에 참여한 학생들은 컴퓨터를 다루거나 프로그래밍을 거의 해본 적 없는 초보 학생들을 알 수 있다.

<Table 6> Prior ability of the subject(n=28)

		1	2	3	4	5	M	SD
Computer class experience	N	15	6	2	1	4	2.04	1.45
	%	53.6	21.4	7.1	3.6	14.3		
Programming confidence	N	16	7	4	1	0	1.86	1.04
	%	57.1	25.0	14.3	3.6	0		

3.3 연구 분석 방법

본 연구에서 DDD 모형이 학습자의 학습동기에 미치는 영향과 CT 개념 이해에 대한 학습자들의 인식을 분석하기 위해 다음과 같은 분석 도구를 사용하였다.

첫째, 사전-사후 학습동기 검사 도구는 Keller (1987)의 The Course Interest Survey를 번안 수정하여 제작한 Likert 5점 척도의 34문항으로 구성된 설문지를 사용하였다[4]. 이 학습동기 검사도구의 문항들을 하위 요소별로 살펴보면 <Table 7>와 같다.

이 SW 교육에 대한 학습동기 검사 도구는 주의집중 8문항, 관련성 9문항, 자신감 8문항, 만족감 9문항으로 총 34문항으로 구성되어 있다. 각 문항의 답변은 5점 척도로 측정하였으며, 각 문항 번호 중 *표시는 부정문으로 된 문항들로 반대로 코딩 하였다. 설문 후 분석을 위해서 각 영역별 문항 점수의 평균을 구하였다.

<Table 7> Test tool for learning motivation

Component	Item No.	Total	α
Attention	1,4*,10,15,21,24,26*,29	8	.834
relevance	2,5,8*,13,20,22,23,25*,28	9	.908
Confidence	3,6*,9,11*,17*,27,30,34	8	.645
Satisfaction	7*,12,14,16,18,19,31*,32,33	9	.935
Total		34	.962

학습동기 설문지의 ‘주의집중’ 영역은 학습에 대해 일관성, 신기함, 변화성의 균형을 갖는지 확인하는 것이며, ‘관련성’ 영역은 수업 내용이 자신에게 유용하고 자신의 목적을 달성하는데 도움이 된다고 인식하는지에 대한 것이고, ‘자신감’ 영역은 학습에서 성공 가능성을 인식하는 것이고, ‘만족감’ 영역은 내적 동기를 의미하며 내재적 강화, 외재적 보상, 공정성에 관련된 질문을 포함하고 있다[4].

이 설문지의 신뢰도는 사후 검사 결과를 가지고

Cronbach’s α계수로 산출하였는데 <Table 7>과 같이 전체 α계수가 0.962로 높은 신뢰도가 확보 되었다. 하위 영역별로 보면, ‘주의 집중’ 영역은 0.834, ‘관련성’ 영역은 0.908, ‘자신감’ 영역은 0.645, ‘만족감’ 영역은 0.935로 나타나 각 하위 영역별로도 신뢰도가 확보되었다. 사전 사후로 학습자의 학습동기 변화 분석을 위해 ‘대응표본 t-검증’ 방법을 사용하였다.

둘째, 학생들의 DDD 모형을 적용한 SW 학습 방법이 CT 개념 이해에 미치는 영향에 대한 인식은 간단한 설문을 통해 분석하였다. 먼저, DDD 모형의 3가지 단계별로 ‘CT를 향상시키는데 도움’을 주었는지에 대한 인식에 대하여 5점 척도로 표시하도록 하여 평균값을 비교 분석하였다. 또한, DDD 모형을 기반으로 한 수업으로 인해 CT의 하위 요소들(문제 분해, 패턴 인식, 추상화, 알고리즘, 자동화)의 각각의 ‘개념을 얼마나 잘 이해할 수 있었는지’ 설문하였다.

4. 결과 분석

4.1 사전 사후 학습동기 분석

본 실험 연구에서 DDD 수업모형을 통한 SW 교육에 대한 학습자들의 사전 사후 학습동기를 분석해 보기 위해 <Table 8>과 같이 분석하였다.

<Table 8> Pre-post test of learning motivation(n=28)

Component	Pre		Post		t	p
	M	SD	M	SD		
Attention	3.57	0.53	3.40	0.69	1.515	.141
relevance	4.04	0.57	3.90	0.68	1.184	.247
Confidence	3.40	0.37	3.60	0.44	-2.831	.009**
Satisfaction	3.47	0.44	3.56	0.58	-.848	.404
Total	14.48	1.91	14.46	2.39	.059	.954

**p<0.01

학습동기의 세부 항목 중 ‘자신감’ 영역은 평균 2.40(SD: 0.37)에서 3.60(SD: 0.44)으로 높아졌으며 이는 통계적으로도 유의미했다(p<0.01). 또한, 학습동기 전체 평균은 사전 14.48(SD: 1.91)에서 사후 14.46(SD:2.39)로 나타났으며 이는 통계적으로도 유의미한 차이를 보이지

않았다. 이는 DDD 모형을 통해 SW를 학습한 학습자들의 수업 후 학습동기가 수업 전과 동일하게 유지되었음을 알 수 있다.

4.2 성취도 수준별 DDD모형 인식 분석

수업 후 실시한 DDD 모형에 대해 학습 단계별로 CT 향상에 도움이 되었는지에 대한 성취도 수준별 인식을 분석하였다.

학생들의 성취도 수준별 인식을 분석하기 위해 학생들이 9~10주차에서 설계 및 구현하여 제출한 최종 프로젝트 산출물의 성취도를 상, 중, 하의 3등급으로 나누어 분석하였다. 성취도는 지도강사를 포함한 전문가 2인이 학생들의 보고서와 엔트리 프로젝트를 교차 채점하여 평가하였다. 성취도의 평가 기준은 다음과 같았다.

- 프로젝트 설명 시 CT의 하위 요소들(문제분해, 패턴 인식, 추상화, 알고리즘, 자동화)을 프로젝트에 적용하여 바르게 이해하여 설명하였는가?
- 순차, 반복, 선택, 데이터, 제어, 병렬화, 이벤트의 개념을 적용하여 구현하였는가?

이 결과에 따라 성취도의 등급(상, 중, 하)을 각각 9명, 10명, 9명으로 인원수를 분배하여 설정하였다. 이러한 분석 결과는 <Table 9>와 (Fig. 1)과 같다.

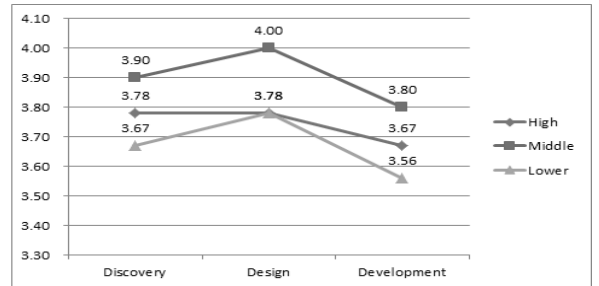
<Table 9> DDD Model Recognition by Level of Achievement

Level	Discovery	Design	Development	Total
High (n=9)	M 3.78 SD 0.97	3.78 0.97	3.67 1.00	3.79 0.83
Middle (n=10)	M 3.90 SD 0.88	4.00 1.05	3.80 0.92	3.86 0.93
Lower (n=9)	M 3.67 SD 0.71	3.78 0.83	3.56 0.73	3.68 0.86
Total	M 3.79 SD 0.83	3.86 0.93	3.68 0.86	

첫째, 세 단계 모두 특히 중간 수준의 학생들이 모든 단계에 대해 CT 향상에 가장 많은 도움을 받았다고 응답했다. 이러한 설문에 대한 응답 신뢰도는 Cronbach's α값 0.902로 높게 나타났다.

둘째, 모든 학생들이 전체적으로 설계 단계가 평균

3.86(SD: 0.93)로 나머지 두 단계에 비해 CT 향상에 도움을 많이 받았다고 응답하였다. 반면, 구현단계는 평균 3.68(SD: 0.86)로 가장 낮은 결과를 보였다.



(Fig. 1) DDD Model Recognition by Level of Achievement

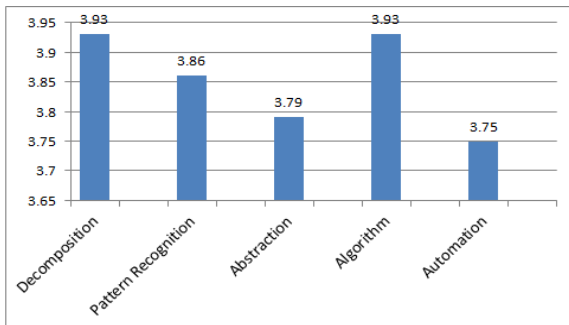
4.3 DDD 모형을 통한 CT 요소별 이해도

DDD 모형에 따른 수업이 어떤 CT 요소들의 개념을 이해에 도움을 주는지에 대한 인식을 분석하였다.

<Table 10> Understanding of CT element by DDD model

CT	Degree	2	3	4	5	M	SD
		N	%	N	%		
Decomposition	N	2	6	12	8	3.93	0.90
	%	7.1	21.4	42.9	28.6		
Pattern Recognition	N	2	6	14	6	3.86	0.85
	%	7.1	21.4	50.0	21.4		
Abstraction	N	3	6	13	6	3.79	0.92
	%	10.7	21.4	46.4	21.4		
Algorithm	N	1	7	13	7	3.93	0.81
	%	3.6	25.0	46.4	25.0		
Automation	N	2	8	13	5	3.75	0.84
	%	7.1	28.6	46.4	17.9		

<Table 10>와 (Fig. 2)에서 보는 바와 같이 DDD 모형은 CT의 '문제분해'와 '알고리즘' 요소의 개념 이해에 도움이 되었다고 응답하였다(문제분해 M:3.93/SD: 0.90, 알고리즘 M:3.93/SD: 0.81). 반면, '자동화' 요소에 대한 개념 이해에 도움을 준 것에 대해서는 평균 3.75(SD: 0.84)로 가장 낮게 나타났다. 이러한 설문에 대한 응답의 신뢰도는 Cronbach's α값 0.962로 높게 나타났다.



(Fig. 2) Understanding of CT element by DDD model

5. 결론 및 제언

본 연구에서는 CT 기반의 DDD 모형을 설계하고 적용하여 그 효과를 검증하였으며 그에 따른 결론은 다음과 같다.

첫째, DDD 모형은 학습자들의 SW 교육에 대한 학습동기를 끝까지 유지 시켜줄 수 있으며, 특히 학습자들의 학습 동기 중 자신감 영역은 유의미한 향상을 가져왔다. 이는 초보 학습자들에게 SW 개발과정을 경험시키는 과정이 인지적 부담을 많이 가져올 수 있음에도 불구하고 본 모형을 통해 자신감을 향상시키고 학습동기를 끝까지 유지하게 했다는 것은 매우 고무적인 결과라고 할 수 있다. 이는 학생들의 피드백에서도 나타났듯이 SW를 구현하는 단계 이전에 실생활 문제를 탐구하고 세부적인 설계과정을 차근차근 안내하며 전체적인 알고리즘 개발과정을 경험하게 한 것이 학습자로 하여금 쉽게 접근할 수 있도록 도운 것으로 분석된다.

둘째, DDD 모형의 설계 단계는 다른 단계들에 비해 학습자들로 하여금 CT 향상에 도움을 준다는 긍정적인 인식을 주었다. 특히, 중간 성취도 수준의 학생들의 CT 향상에 도움을 준 것으로 인식되었다. 또한, CT의 하위요소 중 ‘문제분해’와 ‘알고리즘’에 대한 개념 이해에 효과적임을 알 수 있다. 특히, 이러한 DDD 모형의 학습과정이 성취도 중간 단계 학생들의 산출물의 완성도를 높이는데 도움을 줄 수 있다는 것을 알 수 있었다. 이는 탐구 단계에서 세부적으로 문제를 분석하고 이해하는 단계를 거치고 그 분석 결과

를 바탕으로 설계 단계에서 체계적으로 알고리즘을 설계하는 과정을 여러 차시에 걸쳐 반복적으로 경험했기 때문인 것으로 유추할 수 있다.

결론적으로, 본 연구에서 제시한 DDD 모형은 학습자의 학습동기를 유지하도록 해주며, 주어진 문제 상황에 대한 문제분석 및 분해와 체계적인 알고리즘 설계과정에 대한 경험을 통해 자연스럽게 학습자들의 CT 능력을 신장시키는 데 도움을 줄 것으로 기대된다. 이는 SW 교육이 궁극적으로 CT를 가진 창의·융합 인재를 키우는 것에 목적을 두고 있는 것에 부합하는 교수 학습 방법이라고 할 수 있다.

SW 교육이 초중등학교 및 비전공자로의 확대로 이어지고 있기 때문에 학습자의 학습동기를 유지시키는 일은 매우 중요하다. 이와 같이 학습자의 자신감을 바탕으로 한 학습동기뿐 아니라 CT 역량을 향상시킬 수 있는 다양한 교수 학습 모형의 개발과 검증은 앞으로도 많이 요구될 것이다. 따라서 DDD 모형도 추가적으로 다양한 학교급에 대한 적용 및 검증도 뒷받침될 필요가 있다.

참고문헌

- [1] Department for Education. (2013). *The national curriculum in England: Framework document*. Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/210969/NC_framework_document_-_FINAL.pdf.
- [2] Google for education(2015). *CT Lesson Plan*. <https://www.google.com/edu/resources/programs/>
- [3] International Society for Technology in Education & Computer Science Teachers Association. (2011). *CSTA K-12 Computer Science Standards Revised 2011*. Retrieved from http://csta.acm.org/Curriculum/sub/urrFiles/CSTA_K-12_CSS.pdf.
- [4] J. M. Keller, S. Song (2014). *Attractive instructional design* (2nd ed.). Seoul: Science education publisher.
- [5] J. M. Wing (2008). Computational thinking and thinking about computing. *Philosophical trans-*

- actions of the Royal Society A, 366, 3717-3725.
- [6] Jungsook Sung, Hyeoncheol Kim (2015). Analysis on the International Comparison of Computer Education in Schools. *The Journal of Korean association of computer education*, 18(1), 45-54.
- [7] KERIS (2016). 2015 Education Policy Network Training on-site support research : Development of SW Education Teaching and Learning Model. Commissioned research CR 2015-35.
- [8] KERIS (2016). *Creativity travel with software*. Ministry of Education.
- [9] Miseung Kim (2016), Development and Application of the NDIS based ESD Teaching-Learning Program for Software Education, *Jinju National University of Education Master's Thesis*.
- [10] Ministry of Science and ICT. (2016). *Plan to cultivate human resources for SW-oriented society*, Press Releases, 2016.7.21. <http://www.msip.go.kr/web/msipContents/contentsView.do?catelId=mssw311&artId=1270998> .
- [11] Ministry of Education (2015). *2015 Revised Course*
- [12] Ministry of Education (2015). *Guidelines for operating software education*.
- [13] Park Hyo Min(2014). *Global software educational status and tools trends*, Korea Internet & Security Agency Report focus 3
- [14] Seunghoon Lee, Kapsu Kim (2008). Development and Application of Teaching Model on Project-Based Programming for Elementary Students. *Journal of The Korean Association of Computer Education*, 11(2), 23-33.
- [15] Soojin Jun, Sungwan Han (2016). Development of UMC Teaching and Learning Strategy for Computational Thinking. *Journal of The Korean Association of Information Education*, 20(2), 131-138.
- [16] Soojin Jun (2017). The Effect of Design-Oriented Model (NDIS) based on Computational Thinking in SW Education. *Journal of The Korean Association of Computer Education*, 20(2), 13-21.
- [17] Soojin Jun, SunGwan Han, Soohwan Kim (2017). Effect of design-based learning on improving computational thinking. *Behaviour & Information Technology*, 1-11.
- [18] Sookyoung Choi (2017), Design and Application of an Instructional Model for Flipped learning of Programming Class, *Journal of The Korean Association of Computer Education*, 20(4), 27-36
- [19] Yongju Jeon, Teayoung Kim (2017). The Analysis of Cognitive and Affective Effects on the CT-CPS Instructional Model for the Software Education Class in Middle School, *Journal of The Korean Association of Computer Education*, 20(4), 47-57
- [20] Youngho Seo, Miryeong Yeom, Jonghoon Kim (2016) Analysis of Effect that Pair Programming Develops of Computational Thinking and Creativity in Elementary Software Education. *Journal of The Korean Association of Information Education*, 20(3), 219-234.

저자소개



전 수 진

2000 경인교육대학교
초등교육학과(교육학사)

2005 경인교육대학교
컴퓨터교육과(교육석사)

2015 고려대학교 컴퓨터교육학과
(이학박사)

2000~현재 초등학교 교사
(현 상미초등학교)

관심분야 : 정보교육, 컴퓨팅 사고력, SW 교육방법, SW 교육평가, 융합교육

e-mail : soojin.jun@inc.korea.ac.kr