

초등학교의 소프트웨어 교육에서 학습자의 절차적 사고를 신장시키기 위해 과정 중심 평가를 기반으로 한 수업설계 및 적용

임 화 경¹ · 안 형 모^{2*}

¹부산교육대학교 컴퓨터교육과 교수

^{2*}녹산초등학교 교사

Instructional Design and Application based on Process fortified assessment to enhance Students' Procedural Thinking ability in Software Education in Elementary School

Hwa-Kyung Rim¹ · Hyung-Mo Ahn^{2*}

¹Professor, Department of Computer Education, Busan National University of Education, Busan 47503, Korea

^{2*}Teacher, Noksan Elementary School, Busan 47503, Korea

[요 약]

소프트웨어 교육에서 절차적 사고를 위한 수업은 프로그램 구조와 프로그래밍 결과물 중심의 평가보다 학습자의 절차적 사고의 변화를 관찰하는 과정 중심 평가를 기반으로 설계해야 한다. 학습자의 절차적 사고의 신장은 학습 과정에서 학습자 사고를 관찰하는 것이 중요하기 때문이다. 본 논문에서는 초등학교 5, 6학년 실과교과 내 컴퓨팅 사고영역을 지도하기 위해 과정 중심 평가를 기반으로 수업을 설계하고 적용하였다. 그리고 학습자의 학습자료와 토의·토론 과정을 관찰하면서 나타난 사고의 변화를 추적하여 제안한 수업의 효과를 분석하고 제안하였다.

[Abstract]

Classes for the concept of procedural thinking in software education should be designed based on process fortified assessment that observes the procedural thinking of students rather than the evaluation of implemented software. This is because it is important to observe changes in thinking to evaluate the concept. This study designed and applied classes based on process-fortified assessment to guide the area of computational thinking in Practical Art courses for 5th and 6th grade students at elementary school. It analyzed whether the classes were effective in enhancing the procedural thinking of students by observing and tracing the student's learning data and communication process and made suggestions.

색인어 : 소프트웨어 개발 프로세스, 컴퓨팅 사고력, 절차적 사고, 과정 중심 평가, 수업설계

Keyword : Software development process, Computational thinking, Procedural thinking, Process fortified assessment, Instructional design

<http://dx.doi.org/10.9728/dcs.2022.23.2.247>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 18 January 2022; **Revised** 07 February 2022

Accepted 07 February 2022

***Corresponding Author; Hyung-Mo Ahn**

Tel: (저자요청에 의한 비공개)

E-mail: rim@bnue.ac.kr

I. 서론

우리 사회는 정보화시대에 접어들면서 경제적 성장이 가속화되고 있다. 이런 현상은 첨단 디지털 기술의 산업화 및 사회화 속도가 급속하게 진행되고 있기 때문이다. 현재, 인공지능 기술(artificial intelligence technology) 중심의 4차 산업혁명 시대에 접어들면서 산업화 시대로 진화하는 과정은 더욱 가속화되고 있다. 디지털 기술의 진화는 사회의 모든 분야에 영향을 주고 있으며, 교육 분야에서도 미래사회를 위한 인재양성을 위해 교육정책의 방향과 핵심과제를 발표하고 진행하고 있다[1], [2].

사회의 패러다임 변화에 대비하여 융합인재 양성의 필요성이 2015 개정 교육과정의 기본방향으로 제시되었다[3]. 교육과정의 총론에도 학생들의 컴퓨팅 사고(computational thinking)력 함양을 위해 소프트웨어 교육이 필요함을 강조하고 있다. 이에 따라 초등학교 5, 6학년 실과교과의 기술시스템 영역으로 소프트웨어 개념과 컴퓨팅 사고와 관련한 내용 요소가 추가되었다. 또한, 교수학습의 질을 향상하기 위해 학습 과정에서 학습자의 사고 변화를 관찰하고 피드백하는 과정 중심 평가(process fortified assessment)의 중요성을 강조하고 있다[3], [4]. 전통적으로 수행하고 있는 학습 결과를 중심으로 평가하는 방법을 지양하고 학습 과정에서 나타나는 학습자의 문제를 발견하고 즉시 피드백하면서 활발한 상호작용을 통해 사고력 신장을 강조하는 과정 중심 평가를 지향해야 함을 강조하고 있다[4]-[8].

소프트웨어 교육에서 컴퓨팅 사고력은 주어진 문제를 해결하기 위해 컴퓨터의 기본 원리를 기초로 해결하는 사고 능력이다[3]-[5]. 특히 소프트웨어를 구현하는 과정에서 컴퓨팅 사고에 대한 수업은 학습자 스스로 다양한 문제 분석 과정과 문제 해결 방법의 효율성에 대해 사고하고 비판하는 과정이 가장 중요한 활동이다. 이때 교사는 모든 학습 과정에서 학습자 사고의 변화를 관찰하고 피드백하고 평가해야 한다. 결과를 중심으로 평가하는 전통적인 방법은 문제 이해와 분석 및 알고리즘 설계과정에서 관찰되는 학습자의 컴퓨팅 사고의 변화에 대한 평가를 간과하고 직관적 코딩을 통해 생성된 결과물 중심으로 평가하기 때문에 적합하지 않다. 따라서 학습 과정에서 학습자의 컴퓨팅 사고의 변화를 관찰하려면 이러한 특징을 고려하여 수업설계를 계획하고 평가해야 한다.

본 논문에서는 컴퓨팅 사고의 특징을 고려하여 토의토론과 과정 중심 평가[5]-[10]를 기반으로 수업을 설계하였다. 설계한 수업은 초등학교 5, 6학년 실과교과 내 컴퓨팅 사고 내용 요소영역에서 주제를 선정하였고 다음 사항을 고려하여 설계하였다. 컴퓨팅 사고는 해결해야 할 문제에 대해 다양한 해결방법이 존재한다는 점과 학습자가 문제를 이해하고 분석하는 조건에 따라 다양성을 갖는 점이 특징이므로 학습자 각각의 문제분석 방법에 대해 타당성과 비판적 의견을 나눌 수 있는 활동으로 토의토론 과정이 필요하다[9], [10]. 토의토론 활동을 통해 자신이 분석한 내용을 수정보완하는 과정

을 거치면서 사고력이 향상되기 때문이다. 각 학습자가 제시한 문제분석 결과를 알고리즘으로 표현하고 프로그램화하는 단계에서도 알고리즘의 효율성을 판단하기 위해 동일한 학습 활동이 필요하다. 이 모든 과정에서 교사는 학습자의 학습 활동을 관찰, 코치하고 피드백하면서 학습자 사고의 변화를 평가한다. 이러한 점을 고려하여 설계한 수업을 현장에 적용하였다. 학습의 전 과정에서 학습자의 학습자료와 토의토론 과정을 관찰하면서 나타난 사고의 변화를 추적하고 교수학습 효과를 분석하였다.

본 논문의 구성은 다음과 같다. 2장은 컴퓨팅 사고와 관련하여 교육과정에서 제시하고 있는 교수학습 방법과 과정 중심 평가의 중요성에 대해 살펴보았다. 3장은 토의토론과 과정 중심 평가를 중심으로 설계한 교수학습 절차를 제시하였다. 4장은 제시한 교수학습 절차를 수업에 적용하면서 학습 자료와 토의토론 내용 등을 관찰하여 나타난 학습자 사고의 변화에 대해 분석하였다. 5장은 마지막으로 적용한 수업의 효과에 대해 결론을 내린다. 본 논문의 내용에서 컴퓨팅 사고개념에서 가장 중요한 ‘알고리즘(algorithm)’ 용어는 2015 개정 교육과정에서 정의하고 있는 ‘절차적 문제 해결(procedural problem solving)’ 또는 ‘절차적 사고(procedural thinking)’라는 용어와 함께 혼용하여 맥락에 따라 표현하였다.

II. 과정 중심 평가와 절차적 사고

본 논문의 주제는 현장 교과의 수업 설계가 중심인 연구이므로 연구배경으로 교육과정의 교육내용과 체계 및 교수학습 평가 방향을 중심으로 살펴보았다.

2-1 과정 중심 평가

2017년에 교육부와 한국교육과정 평가원에서 교수학습의 질을 신장시키기 위한 평가 방법으로 과정 중심 평가의 중요성에 대해 강조하기 시작하였다[5]-[7]. 이 평가는 전통적으로 진행해왔던 학문적 교육 시스템의 평가방법인 학습자들의 학습 결과를 중심으로 평가하는 것이 아니라 학습을 수행하는 과정에서 나타나는 문제점을 발견하고 그 즉시 학생에게 피드백하는 방법이다. 교사가 모든 학습 과정을 관찰하면서 나타난 학생의 사고 변화와 학습 신장에 대한 정보를 수집하고 분석하여 피드백함으로써 학생과의 활발한 상호작용을 강조하는 평가방법이기도 하다[6]-[8]. 이 평가방법은 4차 산업혁명 시대에 필요한 창의융합 인재를 양성하기 위하여 교수학습의 질을 신장시키기 위한 중요한 개념으로 급부상하고 있다.

과정 중심 평가에서 교사의 역할은 학습자 중심으로 참여하는 수업을 위해 학습 경험 및 자료 등을 제공하고, 사고를 유도하는 내용을 제시하는 교수학습 방법을 설계하는 것이 중요한 요소라는 것을 강조하고 있다[6], [8]. 2015 개정 교

육과정 총론[3]에서도 “성취기준을 기반으로 학생의 역량을 육성하는 수업과 학습을 통해 기대하는 수행 능력을 기르는 수업을 지향한다”라고 강조하고 있으며, 단순 지식 암기보다는 학생 중심의 참여와 전 학습 과정에서 학생의 수행 능력을 관찰하고 피드백을 통하여 역량을 평가하도록 강조하고 있다. 특히, 공학 기반인 컴퓨터 영역에서 과정 중심 평가는 학습 과정 중에 나타난 수행 활동과 학습 산출물을 증거 자료로 활용하도록 하고 있다[3], [5]. 모든 평가는 목적을 다양하게 해석하고, 학습자의 성취도뿐만 아니라 학습자의 장·단점을 피드백하여 학업적 성장을 신장시키기 위해 적극적으로 활용하는 데에 있다는 것을 강조하고 있다.

2-2 절차적 사고와 과정 중심 평가

초등학교 2015 개정 교육과정의 실과교과 내 소프트웨어 교육의 내용 체계 중 절차적 사고에 대한 학습 과정은 과정 중심 평가를 적용하기에 가장 적합한 내용 요소로 볼 수 있다. 내용 요소는 소프트웨어 개발 프로세스를 기반으로 컴퓨팅 사고의 학문적 기초지식을 내포하고 있으며 다음과 같이 학습자의 컴퓨팅 사고력 신장에 초점을 두고 있기 때문이다.

첫째로, 소프트웨어 개발 프로세스[10]-[14]는 개발하려는 콘텐츠에 대하여 문제 정의(problem definition) 단계를 거쳐 그 문제에 대한 타당성을 분석(problem validity analysis)하고 분석을 통해 결정된 문제 해결 절차를 알고리즘으로 설계(algorithm design)하는 단계로 진행된다. 이 과정에서 개발자의 창의력과 문제해결력에 의해 콘텐츠의 독창성과 효율성이 결정되며 콘텐츠의 질을 결정하는 데 영향을 주게 된다. 즉, 동일 문제에 대해 콘텐츠를 개발할 때도 개발자의 컴퓨팅 사고에 따라 문제를 해결하는 방법이 다양하게 설계될 수 있기 때문이다.

이 과정에서 문제 해결 방법은 그대로 구현되는 것이 아니라 알고리즘의 효율성에 대해 분석과정을 거쳐야 한다. 효율성은 컴퓨터의 처리속도(time)와 컴퓨팅 자원(space)에 영향을 주기 때문이다[12], [14]-[17]. 효율성에 대한 분석도 개발자 간의 반복적인 토론과 해당 단계를 반복적으로 실행하는 과정을 거치면서 가장 효율적인 방법을 선택하게 된다. 이러한 과정을 거쳐 결과물인 콘텐츠를 완성하게 된다. 콘텐츠가 완성된 후에도 효율성 향상을 위해 이 과정은 반복되며 유지보수 기간을 통해 지속하게 된다. 만일 이러한 과정을 간과하고 구현한 콘텐츠에 대해 결과 중심의 가치평가만 수행한다면 심각한 자원낭비를 초래할 수 있기 때문이다.

이러한 절차와 평가는 소프트웨어 교육에서도 반드시 고려되어야 한다. 표 1은 2015 개정 교육과정의 실과(기술·가정) 및 정보과 교육과정에서 기술시스템 영역[5]의 소프트웨어 성취기준의 내용 요소와 소프트웨어 개발 프로세스와 연관 지어 나타냈다.

표 1과 같이 첫 번째 ‘절차적 문제 해결’은 문제를 해결하기 위해 단계별로 처리하는 과정을 절차적 사고로 표현하는

과정으로 설명하고 있다. 이는 소프트웨어 개발 프로세스의 문제 정의 단계, 문제의 타당성 분석 단계, 알고리즘 설계단계와 연관된다. 두 번째 ‘프로그래밍 요소와 구조’는 교육용 프로그래밍 언어를 사용하여 문제 해결 방법을 프로그램화하는 과정을 설명하고 있다. 교육과정에서 이 두 개의 내용 요소를 구분하여 설명하고 있지만, 소프트웨어 개발 프로세스의 다음 단계인 프로그램 언어를 통해 프로그램의 구조를 학습하는 코딩단계, 실행단계, 결과 출력 단계와 연관된다. 즉 소프트웨어 개발 과정을 교육 내용 체계의 내용 요소에서 연관 지어 설명하고 있음을 볼 수 있다.

표 1. 기술시스템 영역의 소프트웨어 관련 내용 요소와 소프트웨어 개발 프로세스

Table 1. Software-related content elements and software development process in the technical system area

Content Element	Software development process
- Procedural problem solving -Apply a procedural thinking process to a problem to solve the problem efficiently	① Problem definition phase ② Problem validity analysis phase ③ Algorithm(&flowchart) design phase
- Programming elements and structure -Understand the programming language structure and experience the process of programming using EPL language	④ Coding phase ⑤ Execution(run) phase ⑥ Output phase

표 2. 기술시스템 영역의 소프트웨어 관련 내용 요소의 교수학습 방법[5]

Table 2. Teaching and learning methods of software-related content elements in the technical system area

Content Element	Teaching and Learning Method
- Procedural problem solving (Algorithm stage)	- Find examples in everyday life where procedural thinking can be applied, and understand the process of procedural problem-solving through various activities such as play-oriented physical activities and puzzles. - Guide students to utilize computational thinking focusing on problematic situations that occur in real life.
- Programming elements and structure (Programming stage)	- Guide students to understand methods and procedures to solve problems without computers, in addition to activities using computers. - Minimize the teaching of how to use application software or the grammar learning of programming languages, and focus on developing computational thinking skills necessary for problem solving.

표 2는 소프트웨어 교육의 내용 체계와 성취기준에서 언급하고 있는 교수학습 방법 및 평가내용이다. 주어진 문제 상황에 따라 문제를 해결하는 방법과 절차를 이해하도록 지도하고 컴퓨팅 사고를 스스로 적용할 수 있도록 지도하며 프로그램의 구조를 학습하는 것이 아니라 프로그래밍을 통해 컴퓨팅 사고력을 신장시키는 것에 초점을 두고 지도하도록 안내하고 있다. 해당 영역의 교수학습 방법에 따른 다양한 평가와 과정 중심 수행평가가 이루어질 수 있도록 강조하고 있다.

절차적 사고와 과정 중심 평가 두 주제를 고려한 관련 연구를 살펴보면 중등과정의 2015 개정 정보 교육과정 분석을 통해 컴퓨팅 사고력 기반의 문제 해결 과정을 개발하고 각 단계에 적합한 성취기준을 제시하고, 평가모형을 적용한 결과에 대한 분석을 진행한 연구[18], [19]와 초·중등과정의 소프트웨어 교육에서 과정 중심 평가를 지원하기 위한 평가도구 개발에 관한 연구[20] 등이 있다. 이 연구들은 교수학습 설계보다는 대부분 초·중등과정의 정보 교과와 성취기준을 근거로 평가모형을 제시하는 것에 중점을 두고 있다. 교수학습 관련 연구는 학습자의 다양한 사고와 절차의 효율성을 판단하는 사고를 관찰하기 위한 설계보다 프로그래밍 과정을 체험하고 결과를 중심으로 평가를 하는 기존의 수행평가를 기반으로 한 연구들이 대부분이었다. 이러한 점을 고려하여 본 논문에서는 기존의 결과 중심 수행평가를 위한 교수학습 설계를 벗어나 학습자의 절차적 사고를 관찰하는 과정 중심 평가에 적합한 교수학습 설계안을 제시하였다.

Ⅲ. 과정 중심 평가기반의 교수학습 절차

본 장에서는 과정 중심 평가를 기반으로 설계한 교수학습 절차와 학습 진행 과정과 활동 내용을 설명하였다.

3-1 교수학습 절차

과정 중심 평가는 학습 과정 중에 학습자의 사고 관찰과 교사의 피드백이 중요하므로 2장의 표 1과 표 2로 나타낸 것과 같이 소프트웨어 개발 프로세스를 기반으로 교수학습 절차를 설계하였다. 그림 1은 소프트웨어 개발 프로세스에 따른 교수·학습 절차와 과정 중심 평가를 위한 학습자의 토의·토론 및 교사의 관찰평가 단계를 나타내고 있다.

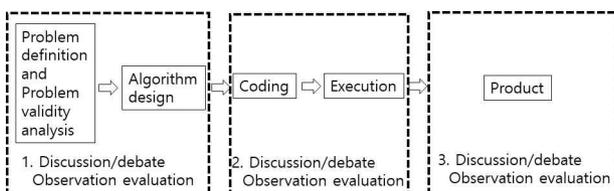


그림 1. 과정 중심 평가를 위한 교수·학습 절차도

Fig. 1. Teaching and learning process diagram for process-fortified assessment

제시한 교수학습 절차는 세 단계로 진행하였다. 첫 번째 단계는 문제분석과 알고리즘을 설계하는 과정에서 효율성과 다양성에 대한 사고를 관찰하고, 두 번째 단계는 코딩과 실행 과정에서 효율성에 대한 사고를 관찰하며, 마지막 단계는 결과물에 대한 학습자의 평가를 관찰하는 것으로 진행하였다. 토의·토론 과정을 지원하기 위해 교사의 관찰, 자기평가, 동료 평가 자료를 기반으로 피드백과 코치를 통해 지도하였다. 수업환경은 온오프라인을 병행하여 지원하였다. 학습자 사고를 관찰하기 위해 활동지[13]와 온라인 실시간 협업을 위해 공유환경을 제공하는 구글 드라이브[21]를 활용하였다.

3-2 학습 진행 과정

수업은 6학년 27명을 대상으로 두 명을 한 개 조로 구성하여 실과교과 시간을 중심으로 진행하였다. 학습주제와 학습자가 토의·토론한 내용을 중심으로 진행한 학습은 다음과 같다. 학습 주제는 난이도를 조절하여 세 개의 주제로 단계를 나누어 진행하였다. 각 주제에 대해 진행한 활동은 표 3으로 정리하였다.

첫 번째 학습 단계는 컴퓨팅 사고의 기초원리를 이해시키기 위해 절차적 사고 표현에 익숙하도록 단순한 주제를 활용하여 수업을 진행하였다.

출발지에서 정해진 도착지까지 경로를 결정하는 단순한 문제를 선택하였으며 결과를 시각적으로 체험할 수 있도록 피지컬 로봇을 활용하였다. 난이도가 가장 낮은 단순한 문제이지만 다양한 절차적 사고를 생각할 수 있도록 몇 개의 경로가 동시에 존재하는 문제로 구성하였다. 학습 과정은 다음과 같다. 먼저, 문제를 해결하기 위해 주어진 제한조건과 사용 가능한 명령어에 대해 의견을 나누는 활동을 진행하였다. 문제 이해에 대한 활동이 끝난 후 문제를 해결하기 위해 경로를 탐색하고 절차적 사고로 표현하는 활동을 진행하였다. 이 과정에서 교사는 학습자 간 탐색한 경로에 대해 절차적 사고로 표현하는 것과 경로를 선택하게 된 타당한 이유에 대해 토의·토론하도록 지도하고 그 과정을 관찰하면서 피드백을 하였다. 그런 후 교사의 피드백과 학습자 간 나눈 의견을 반영하여 효율적인 절차적 사고로 수정보완하는 활동을 진행하였다. 이후 실행 결과물에 대해서도 같은 방법으로 토의·토론을 진행한 후 자기평가와 동료평가 등을 진행하였다.

두 번째 단계의 학습은 첫 번째 단계에서 진행한 문제를 기초로 절차적 사고를 확장하여 제시하였다. 문제는 백신 로봇이 바이러스를 제거하는 경로를 탐색하는 활동으로 다양한 문제 해결 절차와 효율성에 대해 논의할 수 있도록 제한조건을 설정하였다. 경로의 복잡성을 위해 바이러스 개수를 3개, 7개, 10개로 제한하였다. 바이러스를 제거하는 백신은 혈소판과 백혈구가 있는 곳은 지나갈 수 없으며, 한 번 지나간 길은 다시 지나갈 수 없도록 제한하였다. 또한, 바이러스를 모두 제거하면 출발지로 다시 복귀하도록 하였다.

표 3. 학습주제와 절차적 사고 활동

Table 3. Learning topics and procedural thinking activities

Learning Topics	Various Procedural Thinking Activities
- A problem of navigating a simple route using a physical robot	<ul style="list-style-type: none"> - Set conditions for navigating the route from the starting point to destination - Explore procedural problem-solving methods based on conditions (1st feedback) - Discuss the searched route and modify and supplement - Explain feasibility and efficiency (2nd feedback) -Results (evaluation)
- A problem of navigating a route to remove viruses using a vaccine robot	<ul style="list-style-type: none"> - Set conditions for virus removal - A condition that all viruses must be removed and the vaccine robot must return to the starting point - A condition that the vaccine robot can not pass the same route twice - A condition in which white blood cells and platelets can not pass and turn into a virus when they encounter a virus - Explore various procedural problem-solving methods(1st feedback) - Discuss the searched method - Explain feasibility and efficiency (2nd feedback) -Results (evaluation)
- A problem of navigating a route to extinguish forest fires using a fire drone	<ul style="list-style-type: none"> - Set conditions to put out forest fires using a fire drone when forest fires occur - A condition that the fire drone must return to the starting point after extinguishing all fires in all areas where the fire occurred - A condition that the fire drone can not pass the same route twice - A condition of forest fires in different sizes - Explore various procedural problem-solving methods(1st feedback) - Discuss the searched methods - Explain feasibility and efficiency of the methods (2nd feedback) -Results (evaluation)

그렇지 않으면 바이러스에 감염되어 백신이 바이러스로 변하도록 제한하였다. 학습 과정은 첫 번째 단계와 동일 방법으로 진행하였다.

세 번째 단계의 학습은 두 번째 문제가 갖는 절차적 해결방법에 대해 다른 유형의 복잡도를 갖는 문제로 확대하여 설정하였다. 산의 여러 곳에서 산불이 발생한 상황을 가정하였고 드론을 사용하여 산불을 진압하기 위해 산불을 찾아가는 경로를 탐색하는 문제이다. 이 문제 또한 다양한 문제 해결 절차와 효율성에 대해 논의할 수 있도록 제한조건을 설정하였다. 경로의 복잡성을 위해 산불 개수와 산불의 규모를 다양하게 제한하였고 산불을 진압한 곳은 다시 지나갈 수 없으며 모두 진압한 후 반드시 출발지로 복귀하도록 제한하였다. 두 번째 학습 문제는 바이러스와 혈소판 그리고 백혈구의 위치에 따라 다양한 경로를 탐색하도록 하였고 세 번째 학습 문제는 산불의 크기에 따라 다양한 경로를 탐구하도록 하였다. 경로

탐색의 조건은 조원들이 의견을 나누어 결정하도록 하였다. 학습 과정은 첫 번째 단계 학습과 동일 방법으로 진행하였다.

모든 학습 과정에서 교사는 학습자가 문제 해결 방법을 절차적 사고에 따라 표현하지 못한 경우에 순서도 카드를 활용하여 해당 절차 또는 명령어를 기입하도록 제공하여 절차적 표현을 할 수 있도록 피드백하였다. 또한, 프로그래밍 구조인 순차, 선택, 반복의 구조를 적합하게 적용하였는지를 확인하기 위해 프로그래밍의 기본 구조를 이해할 수 있는 예제를 각각 제공하는 것으로 피드백하였다.

IV. 적용과 관찰

본 장에서는 설계한 교수학습 절차를 적용하고 학습자를 관찰하면서 나타난 사고의 변화에 대해 분석하였다. 과정 중심 평가는 설문을 통해 수집된 정량적인 수치로 분석하는 것보다 학습 과정에서 나타나는 사고의 변화를 관찰하고 추적하는 것이 중요하기 때문에 학습 과정을 중심으로 분석하였다. 세 개의 학습 주제는 계층적으로 확장한 문제이므로 세 번째 산불을 제거하는 소방드론의 경로 결정 문제를 중심으로 학습자의 변화를 살펴보았다.

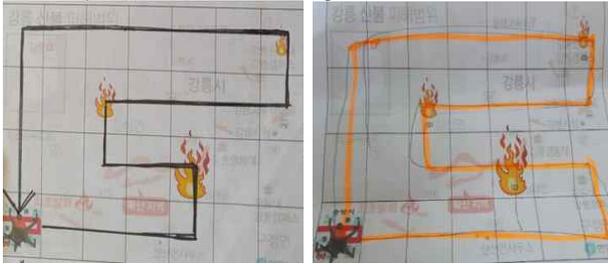
4-1 다양한 문제 해결 방법

첫 번째 단계인 문제 해결 방법을 탐색하는 과정에서 관찰된 내용을 살펴보았다. 이 단계에서는 학습자의 사고 신장을 위해 문제에서 주어진 조건 외에 조별로 토의·토론을 통해 문제 해결 방법과 추가 조건을 결정하고 그 이유를 설명하도록 하였다. 병행하여 교사의 피드백이 수행되었으며 이 과정에서 탐색한 방법은 하나의 정답을 찾아가는 과정에서 벗어나 다양한 방법의 절차가 존재할 수 있는 부분과 각자 결정한 방법에 대해 타당성을 설명하고 다른 의견을 경청하는 활동으로 상호작용을 할 수 있도록 진행하였다.

그림 2는 학습자들이 결정한 문제 해결 방법의 다양한 예를 나타내고 있다. 참고로 활동지에 나타난 산불의 번호는 드론에서 가장 가까운 순서로 1, 2, 3으로 정했다. 학습자들은 다양한 경로를 제시하였고 동일 경로일 때 진행 방향이 다르거나 선택한 이유가 다양했다. 학습자가 결정한 경로에 대한 이유는 크게 세 가지로 나누어 정리할 수 있다. a의 예는 25%의 학습자가 결정한 경로로 드론을 중심으로 거리가 가장 가까운 것부터 또는 현재 산불의 위치에서 가장 가까운 경로를 선택했다고 이유를 설명하였다. ‘중간 위치한 불이 가장 가까워서 먼저 꺾다’, ‘경로가 다시 가는 곳이 없도록 1번부터 꺾다’의 의견이 공통적이었다. 또한, 각각의 경로를 진행하는 방향도 다른 것을 볼 수 있다. b의 예는 가장 많은 학습자가 결정한 경로로 화재의 크기를 고려한 경로이다. 선택한 이유를 보면 ‘2-3-1 순서로 갔다.



a. 거리를 고려한 경로의 예
a. Examples of routes considering distance



b. 산불의 크기를 고려한 경로의 예
b. Examples of routes considering the size of forest fires



c. 명령어의 수를 고려한 경로의 예
c. Examples of routes considering the number of instructions

그림 2. 다양한 문제 해결 방법의 예
Fig. 2. Examples of various problem-solving methods

왜냐하면 2번 불이 가장 크기 때문에 제일 먼저 갔고 3번은 2번 불에서 가장 가까웠기 때문이다’, ‘2번 불에 먼저 간 이유는 2번 불이 가장 크고 불이 많이 번질 것 같아서’, ‘가장 큰불을 먼저 끄고 거리가 가까워서 2, 3, 1 순으로 갔다’, ‘불이 가장 커서 번질 위험이 크기 때문에 2번 불이 가장 커서 작은 불부터 가면 큰불이 번지기 때문이다. 또 1번과 3번은 커져도 2보다 커지지 않기 때문이다’, ‘2, 1, 3 순으로 불이 크기 때문이다’, ‘현재 2번 불이 가장 크기 때문에 다른 불부터 끄기보다는 2번 불을 먼저 끄는 것이 안전하기 때문이다’

등으로 불의 크기를 고려한 의견이 많았다.

기타 의견인 네 번째 예는 a에 해당되지만 첫 번째는 큰 화재부터 방문하고 그 후엔 가까운 불부터 선택했다는 이유를 설명했으며, 출발지점에서 도착점까지 다시 돌아오는 경로가 적기 때문이라며 거리를 고려한 의견도 있었다. 이 경우도 각각의 경로를 진행하는 방향은 모두 다른 것을 볼 수 있다.

마지막으로 c의 예는 20%의 학습자가 결정한 경로로 알고리즘의 명령어 수를 고려한 경우이다. ‘불의 크기가 가장 큰 곳으로 먼저 갔다. 불이 가장 크고 명령어가 적게 들어가서 이렇게 했다’, ‘불이 제일 크고 반복을 사용할 수 있어서’, ‘불이 가장 크고 명령어가 적게 들기 때문이다’, ‘명령어 수가 가장 적을 것 같아서 경로를 이렇게 정하였다’, ‘2번 불이 가장 크고 알고리즘이 가장 적게 들어가기 때문입니다’, 라고 이유를 설명하였다. 사용할 명령어에 대한 구체적 표시까지 절차 표현을 하고 있음을 볼 수 있다. 이 단계에서는 학습자가 결정한 경로가 완벽하지는 않았지만, 학습자 간 다양한 절차적 방법과 이유에 대해 토의·토론하는 과정에서 가장 효율적인 방법을 스스로 찾아갈 수 있도록 교사는 상호작용 활동을 관찰하고 피드백하였다. 이 과정을 통해 결정한 경로를 수정·보완하는 시간도 제공하였다.

4-2 절차적 사고 표현

문제 이해와 분석 단계에서 모둠별로 토의·토론을 거쳐 결정한 문제 해결 방법에 대해 절차적 사고 또는 알고리즘을 표현하는 과정을 관찰하였다. 이 장에서는 절차적 사고보다는 알고리즘 용어를 사용하여 설명하였다. 다양한 알고리즘에 대해 학습자들이 의견을 나눌 수 있도록 실시간으로 공유가 가능한 협업 도구 환경[21]을 제공하였다. 수정 제안 기능과 댓글 기능을 사용하여 의견을 제시하도록 하였으며 교사는 코치 역할을 하였다.

그림 3은 협업 도구에서 학습자들이 활동한 예를 나타내고 있다. a는 협업 도구에 공유된 알고리즘에 대해 학습자들이 상호평가하는 활동의 예이다. 의견을 제시한 부분은 공유된 알고리즘에 사각형 음영으로 활성화되어 학습자 간의 절차적 사고에 대해 서로 생각해 볼 수 있도록 도움을 주었다. b는 상호평가 과정에서 수정 제안 기능을 사용하여 의견을 제시한 예로 조건문의 논리적 표기 오류, 알고리즘의 시작과 끝의 구조의 오류, 절차의 흐름을 나타내는 화살표 오류, 순서도 기호의 오류에 대해 서로 의견을 나누도록 하였다. 이 내용을 바탕으로 알고리즘을 수정하는 활동을 연계해서 진행하였다.

그림 4는 상호평가 과정에서 제시된 의견을 반영하여 최종으로 수정된 알고리즘을 나타내고 있다. 이 과정을 통해 사고가 변화하는 모습을 최종으로 수정된 순서도를 통해 관찰할 수 있었다. 협업 도구를 활용하여 나눈 의견을 관찰한 결과 학습이 진행될수록 알고리즘의 오류와 수정에 대한 의견에서 격려와 칭찬하는 의견으로 변화하는 것도 확인할 수 있었다.

것들을 놀려서 게임을 똑똑 만들 수 있을 줄 알았다는 성찰이 대부분이었다. 학습이 진행되면서 ‘컴퓨터로 하는 것보다 알고리즘을 표현하는 게 더 중요한 걸 알았다. 알고리즘을 순서도로 표현하는 과정이 처음에는 어렵고 힘들었지만, 문제가 해결되고 프로그램으로 만들어지는 것을 확인하고 나면 뿌듯하고 재미있었다’, ‘프로그램이 어떻게 만들어지는지를 알고 사용해보니 프로그램에 대해 나름대로 분석도 하게 되고 더 잘 사용하게 되었다’, ‘소프트웨어 수업을 통해 문제를 해결할 때 알고리즘을 작성하는 것이 중요하다는 것을 알았다’, ‘문제를 해결하기 위한 방법을 순서도로 표현하는 과정이 어려웠지만 매우 재미있었다’, ‘프레젠테이션과는 완전히 다르게 문제를 해결하기 위한 알고리즘을 순서도로 표현하고 그 순서도에 맞게 명령어를 하나씩 따라 하여 프로그램이 만들어진다는 것을 새롭게 배우게 되었다’, ‘문제를 분석하고 해결방법을 생각하고, 알고리즘을 순서도로 표현하는 과정이 처음에는 어렵고 지루하게 느껴졌다’, ‘알고리즘을 생각하는 것이 너무 어려웠지만, 친구들의 생각이 많이 도움이 되었다’, ‘똑같은 문제를 서로 다르게 프로그램으로 만드는 것이 재미있었다’ 등의 성찰 의견이 있었다.

마지막 단계에서는 ‘만든 알고리즘을 바탕으로 프로그램이 만들어지는 걸 보니 제 실력도 높아진 것 같아 기분이 좋았다’, ‘제가 만들고 싶은 게임의 스토리만 작성하였는데 지금은 각 캐릭터의 알고리즘을 생각하게 되었다’, ‘소프트웨어는 특별한 사람들만 만들 수 있는 것인 줄 알았는데 알고리즘을 잘 짜면 저도 프로그래머가 될 수 있겠다’, ‘엄청 어렵지만 할 것이라고 생각했는데 쉬운 문제부터 차근차근 하다보니 내가 하고 싶은 것들을 만들어 나갈 수 있게 됐다’, ‘생각을 컴퓨터를 통해 표현할 수 있는 즐거운 경험이었다’, ‘우리가 사용하던 프로그램이나 앱이 어떻게 만들어지는지를 알게 되어서 좋았다’, ‘제 주위에 발생하는 문제도 해결순서를 생각하는 습관도 생겼다’ 등의 성찰이 있었다.

교사가 관찰한 내용은 다음과 같다. 학습을 진행하면서 초반부에는 게임 제작 등으로 인식하고 즉각 코딩에만 관심을 가졌으며 문제분석과 알고리즘 설계에는 관심을 두지 않았다. 다양한 예제를 통해 문제 해결 방법을 절차대로 생각하고 순서도로 표현하는 학습을 진행하면서 재미있어하기도 했지만 어려워하였다. 점차 학습 내용이 진행되면서 절차를 작성할 때 알고리즘과 프로그램이 무슨 관계가 있는지에 대해 질문을 하기 시작하였다. 순서도를 바탕으로 프로그래밍하는 학습을 진행하면서 순서도 작성의 필요성을 이해하는 반응을 보였다. 또한, 자신들의 순서도 설계를 바탕으로 프로그래밍을 체험하면서 알고리즘 작성에 집중하면 프로그램 제작이 훨씬 수월하다는 것을 인지하기 시작했다.

중후반부는 문제분석에 따른 알고리즘 작성에 초점을 맞추고 학습을 진행하였고 문제를 제시하면 문제분석 과정을 시작하기도 전에 학습자 스스로 절차를 머릿속으로 생각하는 모습이 많이 나타났다. 토의토론과정을 통해 수정되는 알고리즘의 수준이 점차 높아지는 것을 볼 수 있었으며, 상호평가

기준이 점점 수준이 높아지는 것을 관찰할 수 있었다. 또한, 소극적이었던 학습자도 자기 생각을 적극적으로 표현하였고 모든 학습자가 적극적으로 토의토론 활동에 참여하는 모습을 보였다. 이 학습이 진행된 후에도 제시된 문제를 분석하는 능력, 실현 가능한 해결방안을 도출하는 능력, 해결방안을 절차대로 표현하는 능력, 순서도 작성능력이 향상되고 있음을 볼 수 있었다. 타 교과에서도 문제 해결 방안이 제시되면 절차를 떠올리는 성향이 관찰되었다.

V. 결 론

지금까지 살펴본 바와 같이 학습자 활동을 관찰하고 사고의 변화를 추적한 결과, 과정 중심 평가방법이 절차적 사고의 다양성과 효율성을 판단할 수 있는 컴퓨팅 사고를 신장시키는데 중요한 학습평가임을 확인할 수 있었다. 과정 평가를 통해 학습이 진행될수록 문제를 해결하기 위해 분석하고 설명하는 능력과 효율적인 해결방안을 도출하고 절차대로 표현하는 능력이 향상되는 것을 볼 수 있었다. 다양한 문제 해결과 절차적 표현 및 문제 해결 방법에 대한 효율성 등을 사고하는 과정이 중요한 컴퓨팅 사고에 대한 수업은 기존의 결과 중심의 수행평가보다 과정 중심 평가를 고려하여 설계하는 것이 적합함을 확인할 수 있었으며 컴퓨팅 사고 수업을 위한 교수 학습 모델로 기본방향을 제시했다는 점에서 본 논문은 의미가 있을 것이다. 제시한 모델은 문제 해결 절차가 하나의 정답이 존재하는 경우는 학습자의 사고 신장에 제한을 줄 수 있으므로 다양한 문제 해결 방법을 찾는 과정에서 토의토론 활동이 활발히 일어날 수 있도록 융복합 문제개발 또한 중요한 과제로 남아있다.

감사의 글

본 연구는 2021년도 부산교육대학교 학술연구과제로 지원을 받아 수행되었음.

참고문헌

- [1] Ministry of education, The direction of education policy in the era of artificial intelligence, and key challenges, Nov. 2020.
- [2] Ministry of education, 2022 revised education course, Nov. 2021.
- [3] Ministry of education, 2015 revised education course, No. 2015-74, 2015.
- [4] Korea institute for curriculum evaluation, Development and research of evaluation standards for Practical Art courses/Technology and Home Economics department at

elementary and middle schools according to the 2015 revised curriculum, CRC 2016-2-8, 2016.

[5] Ministry of education, Evaluation standards according to the 2015 revised curriculum (5th and 6th grades in elementary school), 2015-74, 2015.

[6] Ministry of education and Korea institute for curriculum evaluation, How to the process fortified assessment. 2017.

[7] Gyeonggi institute of education, A study on the status of performance evaluation and improvement measures of process fortified assessment, policy 2017-5, 2017.

[8] D. I. Khang and C. K. Jung, *What is process fortified assessment?*, Edunty, 2018.

[9] M. S. Jung, *Discussion and debate teaching method*, Kyoyookbook, 2017.

[10] H. Y. Jung and Y. G. Seo. "Effect of Autonomous Cooperative Learning on the Computational Thinking in Software Education," *Journal of Digital Contents Society*, Vol. 22, No. 12, pp. 1977-1985, Nov. 2021.

[11] H. K. Rim and Y. H. Kim, "Design of App. Contents to Support the Discussion and Debate Learning Process in Elementary School Classes," *Journal of Digital Contents Society*, Vol. 21, No. 1, pp. 55-62, Jan. 2020. <https://doi.org/10.9728/dcs.2020.21.1.55>

[12] S. Ian, *Software Engineering*, pearson, 2015.

[13] H. K. Rim and Y. N. Cho, "A study on the Documentation Format to Improve Elementary School Students' Ability to Express Computational Thinking in Programming Classes," *Journal of Korea Multimedia Society*, Vol. 21, No. 2, pp. 310-322, Feb. 2018. <https://doi.org/10.9717/kmms.2018.21.2.310>

[14] H. S. Kang, J. H. Cho, and H. C. Kim, "A Study on Software Analysis and Design Education Model based on Computational Thinking," *Journal of Digital Contents Society*, Vol. 20, No. 5, pp. 947-954, May 2019. <https://doi.org/10.9728/dcs.2019.20.5.947>

[15] H. K. Rim, "A Study on Teaching using Website 'Code.org' in Programming Education based on Computational Thinking," *Journal of Korea Multimedia Society*, Vol. 20, No. 2, pp. 382-395, Feb. 2017. <https://doi.org/10.9717/kmms.2017.20.2.000>

[16] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithm*, 3rd ed. Mit Press, 2014.

[17] M. T. Goodrich, R. Tamassia, and D. Mount, *Data Structures and Algorithms in C++*, Wiley, 2011.

[18] J. H. Lee, K. H. Lee, and J. W. Cho, "Development of Computational Thinking-Based Problem Solving Processes and Achievement Standards for Course-Based Performance Evaluation," in *Proceeding of the 24th Annual Korean*

Association Of Computer Education, 24(2(A)), pp. 3-7, Aug, 2020. <http://www.dbpia.co.kr/journal/articleDetail?nodeId=NOD E10529455>

[19] J. H. Lee and J. W. Cho, "Process-oriented Evaluation Method for Computational Thinking," *Journal of Digital Convergence*, Vol. 19, No. 10 pp. 95-104, 2021. <https://doi.org/10.14400/JDC.2021.19.10.095>

[20] K. G. Lyu and S. H. Kim, "Development of Debugging Tasks and Tool for Process-centered Assessment on Software Education," *The Journal of Korean Association of Computer Education*, Vol. 23, No. 4, pp. 61-68, 2020.

[21] Google Drive [Internet]. Available: <https://drive.google.com/drive/>.



임화경(Hwa-Kyung Rim)

1993년 : 서강대학교 대학원 컴퓨터공학과 (공학석사)
 1998년 : 서강대학교 대학원 컴퓨터공학과 (공학박사-병렬처리시스템)

2022년~현 제: 부산교육대학교 컴퓨터교육과 교수
 ※관심분야 : 소프트웨어·인공지능 교육(Software·Artificial Intelligence Education), 교수학습 방법 및 설계 (Instructional Method & Design) 등



안형모(Hyung-Mo Ahn)

2013년 : 한국교원대학교 대학원 초등 컴퓨터 교육전공 (교육학석사)

2022년~현 제: 녹산초등학교 교사
 ※관심분야 : 소프트웨어·인공지능 교육(Software·Artificial Intelligence Education), 교수학습 방법 및 설계 (Instructional Method & Design) 등