

Exploratory Research on Early Childhood Software Education: Focusing on the Computational Thinking of In-service Teachers¹⁾

Choi, Myunghee (Shingu University, Assistant Professor)

Kwon, Sukjin²⁾ (Howon University, Associate Professor)

< ABSTRACT >

The purpose of this study is to explore the viability of software education in early childhood education. To this end, year five class teachers were trained in computational thinking for the purpose of employing software education. Through in-service teacher training, they developed an understanding of sub-constructs such as problem decomposition, pattern recognition, algorithm, abstraction, and debugging thinking, which compose computational thinking. After that, they try to evaluate children's computational thinking. This series of training programs improved teacher computational thinking from an average of 2.2 to 3.3(5-point Likert scale). In addition, it was confirmed through content and word cloud analysis that the computational thinking of the teacher was relatively close to expert level. Considering the link with elementary education, which requires software education, efforts are needed to provide in-service teacher training through a decentralized approach developed according to the situational needs of early childhood education institutions. Furthermore, software education is needed to improve the computational thinking of preservice teachers in preparation for societal demand.

Key Words : Early childhood education, software education, early childhood software education, computational thinking, inservice teacher training

1) This research was supported by 2019 Math-Science Curriculum Improvement Research Program through the NCSOFT Nursery School funded by the NCSOFT Corporation.

2) Corresponding Author: Kwon, Sukjin, Associate Professor, Howon Univ, 64 Howondae 3gil, Impi, Gunsan, Jeollabuk-do, Korea, 54058 / E-mail: sjkwon@howon.ac.kr

유아교육에서 소프트웨어 교육 가능성 탐색: 교사의 컴퓨팅 사고 변화를 중심으로¹⁾

최명희 (신구대학교, 조교수)

권숙진²⁾ (호원대학교, 부교수)

< 요약 >

본 연구는 유아교육에서 소프트웨어 교육의 가능성을 탐색해보는데 그 목적이 있다. 이를 위해 만 5세 영유아교사 5명을 대상으로 소프트웨어 교육의 목적인 컴퓨팅 사고에 대한 교사 연수를 시행하였다. 우선 이들로 하여금 컴퓨팅 사고의 하위 요소인 문제분해, 패턴인식, 알고리즘, 축약화, 디버깅 사고 등에 대한 개념이해를 하게 한 후, 내재된 방식으로 컴퓨팅 사고를 교수학습 활동에 적용해보도록 하였으며, 루브릭으로 유아의 컴퓨팅 사고를 평가해보게 하였다. 이러한 일련의 연수 과정을 통해 교사의 컴퓨팅 사고는 평균 2.2에서 평균 3.3으로 향상되었다. 또한 질적 평가를 위해 수업 일지 자료를 활용하여 내용 분석과 워드클라우드 분석한 결과, 교사의 컴퓨팅 사고는 비교적 전문가 수준에 가깝게 형성되었음을 확인하였다. 향후 소프트웨어 교육이 의무화된 초등교육과 연계성을 고려하여 유아교육기관의 상황과 요구에 따라 분산적 접근 방식으로 교원 연수를 제공하는 노력이 요구된다. 더 나아가 미래사회에 대비하여 현직교사뿐만 아니라 예비교사 양성을 위한 교직 과목 내에서 문제해결사고와 관련하여 컴퓨팅 사고를 다룰 필요가 있다.

주요어 : 유아교육, 소프트웨어 교육, 유아소프트웨어 교육, 컴퓨팅 사고, 교사 연수

1) 본 논문은 2019년 (주)엔씨소프트의 지원을 받아 수행된 연구 과제 결과임.

2) 교신저자: 권숙진, 부교수, (54058) 전북 군산시 임피면 호원대3길 64 호원대학교 / E-mail: sjkwon@howon.ac.kr
논문투고: 2019. 11. 17 / 심사일자: 2019. 11. 19 / 게재확정일자: 2019. 12. 3

I. 서론

4차 산업혁명으로 인해 21세기 미래 인재에게 요구되는 역량 중 하나로 컴퓨팅 사고가 강조되고 있다. 교육 분야에서 컴퓨팅 사고자(computational thinker)를 양성하기 위해 미국의 경우 STEM 교육을 STEM-C(STEM-Computing K-12 Education)로 전환한 바 있으며, 국내에서는 2015 개정교육과정에서 초·중등 소프트웨어 교육을 의무화하였다. 소프트웨어 교육을 강화함에 따라 담당 교원의 교과 지도 전문성과 지능정보역량 강화가 요구되고 있으며(Lim, Ryu, & Kim, 2017), 국정과제 54(미래 교육 환경 조성 및 안전한 학교 구현)를 통해 다양한 연수가 진행되고 있다(MOE, 2018). 연수 내용으로는 프로그래밍 언어를 활용한 학생 참여 중심·놀이(체험) 중심 교육, 교과 융합·실생활 문제 해결 교육을 통해 교육과정의 성취기준 등을 다루고 있다. 또한 최근 스토리텔링, 콘텐츠, 창의성 등의 인문학이나 수학·과학 등의 인접 학문과 연계·융합하는 방식이 연수에서 권장되고 있다.

교육과정 개정 연수로 인해 초·중등교원을 대상으로 소프트웨어 교육 연수가 시행되면서 교원연수 시간이 증가되었다. 그럼에도 불구하고, ICT4ED 전문가회의에 따르면 여전히 소프트웨어 교육을 위한 교사의 준비가 미흡하다고 지적되고 있다(KERIS, 2017). 초·중등교육에서 소프트웨어 교육 및 교사 연수의 중요성과 마찬가지로 유아교육에서도 소프트웨어 교육에 대한 필요성이 인지되어, 관련 연구가 이루어지고 있다. 국내 선행연구로는 유치원 교사의 소프트웨어 교육에 대한 관심을 관심중심수용모형으로 조사(Park, Jung, & Kang, 2018), 인식과 요구 내용을 조사(Jo, Park, & Hong, 2017)하거나 누리과정 기반 유아소프트웨어 교육과정 구성 방향 탐색(Jo & Hong, 2017), 유아 교사를 위한 소프트웨어 교육 연수 프로그램 개발(Hong, Jo, & Park, 2017) 등이 있다. 기존 연구들의 공통점은 현직 교사를 대상으로 한 소프트웨어 교육에 대한 가능성 탐색과 방향 설정이라는 수준에 머물러 있다. 그러나 보다 근본적으로 소프트웨어 교육의 목적인 컴퓨팅 사고에 초점을 두어야 하며, 이에 대한 유아 교사의 이해 수준 향상과 교수학습 활동에 적용할 수 있도록 실제적인 지원 노력이 중요하다.

본 연구는 소프트웨어 교육에 있어 유초연계가 필요한 만 5세반 담당 영유아교사를 대상으로 하였다. 유아소프트웨어 교육을 위해 컴퓨팅 사고의 개념 이해, 교수학습활동에 컴퓨팅 사고의 적용, 유아의 컴퓨팅 사고 평가방법 등의 내용으로 집체연수를 시행하였으며, 이를 통해 교사의 컴퓨팅 사고 변화를 관찰하였다. 교사들로 하여금 기존 수·과학 수업에서 유아의 컴퓨팅 사고 향상에 필요한 활동을 내재시키도록 하였는데, 이는 2015 소프트웨어 교육 운영 지침에서 컴퓨팅 사고력 향상에 수·과학 수업과의 연계를 권장하고 있기 때문이

다(MOE, 2018).

II. 이론적 배경

1. 유아 수 · 과학교육에서 컴퓨팅 사고

컴퓨팅 사고는 Seymour Papert(1980)의 저서 *Mindstorms*에서 처음 언급된 후 Wing(2006)에 의해 대중화되었다. 초기에 컴퓨팅 사고는 컴퓨터 과학자들이 사고하는 방식으로 정의되었다. 컴퓨터 과학자들은 복잡한 문제를 이해하기 위해 문제를 분해하고, 해결에 불필요한 부분을 없애는 축약화 사고 및 다양한 해결책을 탐색하기 위해 발견적(경험적) 추론, 그리고 반복적으로 절차를 활용하는 재귀적(알고리즘) 사고를 한다. Wing(2006)은 이러한 문제해결력을 컴퓨터 과학자의 사고에 국한시키지 않고 모든 이가 모든 분야에서 범용적으로 적용 가능한 분석적 사고로 정의하였다. 이후 학자와 기관마다 상이하게 정의하고 있는데, 이를 Bers(2017)는 두 가지로 범주화하였다. 첫째, 컴퓨팅 사고를 문제해결 과정으로 보는 관점이다(Wing, 2011; ISTE & CSTA, 2011). 컴퓨팅 사고는 문제를 수립하고 해결책을 만들어 컴퓨팅 시스템 혹은 인간과 같은 정보처리 에이전트(information processing agent)를 통해 효과적으로 수행되도록 하는데 필요한 사고 과정으로 정의된다. 둘째, 컴퓨팅 사고를 표현의 기술로 보는 견해이다(Bers, 2017). 컴퓨팅 사고란 디지털 미디어 혹은 컴퓨터 미디어와 같은 표현 에이전트(expressive agent)로 아이디어를 표현하고 커뮤니케이션하는 수단으로 정의된다.

Zhong, Wang, Chen, & Li(2016)는 기존의 두 관점을 혼합하여 컴퓨팅 사고를 문제해결 과정에 필요한 사고이면서 시각화 사고로 정의하고 있다. 따라서 컴퓨터 사고를 하는 사람은 문제분해, 패턴인식, 알고리즘, 축약화, 디버깅 사고를 통해 문제를 분석할 수 있으면서, 디지털 기기로 자신의 생각을 표현하고 타인과 공유할 수 있게 된다. 이러한 관점을 취하고 있는 Brennan & Resnick(2012)은 컴퓨팅 사고를 교육하고 이를 평가하기 위해 개념, 실천, 관점 등 세 차원으로 구분해야 함을 주장하였다. 컴퓨팅 사고 개념(computational concepts)이란 프로그래밍하는데 사용되는 주요 용어(개념)에 대한 이해로서, 이벤트, 조건, 루프가 그 예이다. 컴퓨팅 사고 실천(computational practices)이란 문제해결에 사용되는 실제 사고 과정인 문제분해, 패턴인식, 축약화, 알고리즘, 디버깅 사고를 말한다. 컴퓨팅 사고 관점(computational perspectives)이란 자신의 생각을 표현하고 동료와 함께 공유하는데 필요한 역량으로 프로그래밍에 흥미를 가지는 태도 및 자신감, 프로그래밍 효능감 등을 말한다.

이러한 컴퓨팅 사고 향상을 위해 우리나라의 경우 연계성 높은 학문 분야인 수학과 과학을 통합하여 실생활 문제 해결을 중심으로 교실 수업이 이루어지도록 권장되고 있다(MOE, 2018). 미국의 경우 국가연구위원회(National Research Council, NRC)가 제시한 차세대과학 교육표준(NGSS)에서 컴퓨팅 사고가 강조되었으며, 이를 위해 수·과학 교육 분야에서 컴퓨팅 사고 향상에 노력을 기울이고 있다(Kwon, 2019). 대부분의 나라처럼 우리나라 초·중등 교육에서도 저학년의 경우 통합 교과로, 고학년의 경우 독립 교과로 운영되고 있으며(KERIS, 2017), 최근 유아교육 분야에서도 컴퓨팅 사고력의 중요성을 인지하여 현장 적용 연구들이 부분적으로 이루어지고 있다. 그러나 아직까지는 일상과의 관련성이나 수·과학 수업과의 연계 측면을 강조하기보다 코딩을 위한 교구 개발(Kim, Hong, & Kim, 2016), 교사나 학부모의 인식 및 요구 조사(Jo, Park, & Hong, 2017), 언플러그드 컴퓨팅 활동 효과 분석(Jung & Park, 2018) 등에 초점을 두고 있다. 반면 기존 수·과학 수업에서 컴퓨팅 사고력 향상을 위한 수업 설계나 교사나 유아를 대상으로 한 컴퓨팅 사고력 평가와 관련된 연구는 다소 부족한 편이다(Kwon, 2019). 이는 유아교사가 관련 수업을 진행하기에 초중등교원에 비해 컴퓨팅 사고력에 대한 연수 기회가 적어 이해 수준이 상대적으로 낮거나 그렇지 않더라도 유아를 대상으로 한 컴퓨팅 사고 변화를 측정할 수 있는 평가 방법이 제한적이기 때문이다.

수·과학 수업에서 컴퓨팅 사고를 향상시키고 그 효과를 측정하기 위해서 무엇보다 관련 개념에 대한 교사의 이해가 선행되어야 한다. 초·중등교육의 경우 국가주도로 교사 대상의 소프트웨어 교육 연수가 시행되어 컴퓨팅 사고에 대한 개념 이해 및 수업 활동을 지원하고 있다. 또한 지필검사와 코딩 능력 평가 도구들이 개발되어 있어 학생들의 컴퓨팅 사고 변화를 명시적으로 측정가능하다. 그러나 유아교육의 경우 별도의 연수 지원이 부족한 상황이며 교사가 수업 적용에 참고할 수 있는 활동이 다양하게 개발되어 있지 않은 편이다. 뿐만 아니라 초중등 학생을 대상으로 한 평가방법을 유아에게 적용하는데 한계가 있으며, 무엇보다 근본적으로 문제해결과정이라는 컴퓨팅 사고력 향상의 본연의 목적을 고려한다면 유아를 포함한 학생 평가 방법은 교사의 주관적인 관찰 평가에 무게를 두어야 함이 강조되고 있다(KERIS, 2017). 이를 종합적으로 고려하였을 때 컴퓨팅 사고와 관련된 개념 이해와 수·과학 수업에서 컴퓨팅 사고 향상을 위한 수업 방법, 교실 수업 상황에서 유아의 컴퓨팅 사고의 변화를 관찰하는 방법에 대한 교사 연수가 필요하다.

2. 컴퓨팅 사고 향상을 위한 유아소프트웨어 교육

초·중등교육의 경우 컴퓨팅 사고 개발의 중요성을 인식하고 2015개정교육과정에서 소프

트웨어 교육(이하 SW 교육)으로 의무화하였다. 초등학교는 2019년 5·6학년 ‘실과’ 과목에서, 중학교는 2018년부터 ‘정보’ 과목에서 소프트웨어 필수 교육이 시작되었다. 이러한 교육과정 반영과 함께 SW 교육 담당교원의 역량강화를 위한 연수가 지속적으로 시행되고 있다. SW 교육을 위한 교사역량 강화를 지원하는 기관인 과기부는 한국과학창의재단을 실행기관으로 선도학교를 운영하고 있으며, 교육부는 한국교육학술정보원을 실행기관으로 하여 연구학교를 운영하고 있다. 이처럼 과기부는 교육부와 협력하여 2015년부터 교원들의 소프트웨어 교과지도 전문성 및 수업 능력 향상을 위한 연수과정을 개발하여 운영하고 있다 (MSIT, 2017). 소프트웨어 교육이 학교 현장에서 잘 이루어질 수 있도록 2018년에 초등학교 5·6학년 담임교사와 중등 ‘정보·컴퓨터’ 자격교원을 대상으로 심화연수를 제공한 바 있다.

SW 교육을 위한 교사연수 목표는 여섯 가지로 구분되어 있다(KOFAC, 2015). SW 교육에 대한 태도 변화, SW 교육 이론에 대한 기본 지식획득, SW 교육환경 조성, 학생들의 SW 교육 산출물을 평가할 수 있는 능력 배양, 창의적 SW 교육을 위한 다양한 수업방법에 대한 이해와 적용, 교육과정과 연계된 SW 교육 실시 등으로 설정되어 있다. 초·중등 SW교육에서 교사연수 목적을 활용하여 <표 1>과 같이 컴퓨팅 사고의 필요성 및 유아소프트웨어 교육, 컴퓨팅 사고 핵심 구성요소에 대한 이해, 유아의 컴퓨팅 사고 평가방법, 요소별 교수학습 활동, 누리과정과 연계 등 교사 연수에 대한 시사점을 도출할 수 있다.

<표 1> 초·중등 SW교육 교사 연수 목표에 따른 유아SW교육

교사 연수 목표	유아소프트웨어 교육 연수 적용 사례
① SW 교육에 대한 태도 변화	4차 산업혁명 시대의 핵심역량으로 컴퓨팅 사고와 유아소프트웨어 교육
② SW 교육 이론에 대한 기본 지식획득	컴퓨팅 사고의 핵심 구성요소 (분해, 패턴인식, 추약화, 알고리즘, 디버깅)
③ SW 교육환경 조성	유치원에서 언플러그드/플러그드 활동 사례
④ 학생들의 SW 교육 산출물을 평가할 수 있는 능력 배양	핵심 구성요소별 유아의 컴퓨팅 사고 변화 평가 방법
⑤ 창의적 SW 교육을 위한 다양한 수업방법에 대한 이해와 적용	핵심 구성요소별 유치원에서의 교수학습 활동
⑥ 교육과정과 연계된 SW 교육 실시	누리과정 연계된 수·과학 탐구 활동으로 수업 적용

Jo, Park, & Hong(2017)은 유치원 교사를 대상으로 유아소프트웨어 교육의 필요성에 대해 설문조사하였는데, 이들은 보통의 인식 수준과 낮은 관심도를 보이고 있으며, Park et al.(2018) 연구에서도 관심 단계 중 가장 낮은 지각 수준에 머물러 있음이 확인되었다. 대부

분의 교사들은 유아소프트웨어 교육에 대한 경험이 없기 때문에 비사용자 패턴을 보였다고 할 수 있다. Hong, Jo, & Park(2017)의 경우 유아소프트웨어 교육에 대한 연수 프로그램의 중요성을 주장하고, 이를 15차시로 구성하여 개발하였지만 현장에 적용하여 그 효과성을 검증해보지 못했다는 한계를 가지고 있다. Jung & Park(2018)의 경우 생활도구 주제로 수·과학 중심의 언플러그드 활동을 설계하여 교사로 하여금 수업에 적용해보게 하였고, 그 효과를 분석한 바 있다. 그러나 기존 수업과 연계보다는 초등에서의 소프트웨어 교육처럼 코딩 교육에 초점을 두고 진행되었다. 이러한 점에 근거하여, 교사 연수를 통해 미래사회에 요구되는 핵심역량인 컴퓨팅 사고 개발의 중요성을 인식하게 함과 동시에 컴퓨팅 사고의 핵심개념을 이해한 후 기존 수업에 적용할 수 있도록 유도하는 것이 필요하다. 또한 수업 적용 후 효과를 측정하게 함으로 교사의 컴퓨팅 사고에 대한 이해 및 적용 수준 정도를 평가할 수 있을 것이다.

Ⅲ. 연구방법

1. 연구 대상

연구 대상은 경기도에 위치한 A 어린이집에 재직 중인 만 5세 담임교사 5명이었다. A 어린이집은 만 5세가 두 반인데, 1반의 경우 15명(남: 9명, 여: 6명), 2반의 경우 18명(남: 7명, 여: 11명)으로 구성되어 있었다. 2018년 5월, 집체연수를 시행한 후 2019년 1월까지 9개월 동안 교사들로 하여금 생활주제별로 유아의 컴퓨팅 사고 향상을 위한 수·과학 수업 진행과 평가를 병행하게 하였다.

2. 연구 도구

만 5세 수·과학 수업의 목적을 유아의 컴퓨팅 사고 향상에 두었다. 이를 위해 유아교육 및 교육공학을 전공한 박사 2인이 5명의 교사에게 컴퓨팅 사고와 유아소프트웨어 교육의 관계 및 컴퓨팅 사고의 핵심 구성요소, 구성요소별 교수학습 활동과 누리과정과 연계, 유아의 컴퓨팅 사고 평가방법 등의 내용으로 구성된 교사 연수 및 컨설팅을 제공하였다.

우선 교사들로 하여금 유아교육에서 핵심역량 중 하나인 컴퓨팅 사고의 필요성과 개념을 다루었다. 컴퓨팅 사고는 여러 유형의 사고 중 분석적 사고로 분류되는데, 공통적으로 문제 분해(decomposition), 패턴인식(pattern recognition), 추약화(abstraction), 알고리즘(algorithm),

디버깅(debugging) 등 다섯 가지 핵심 요소로 구성되어 있다(Csizmadia et al., 2015). 문제분해는 문제해결전략 중 하나로 한 번에 해결하기 어려운 문제를 여러 개의 작은 부분으로 나눈 후 작은 단위의 하위 문제들을 하나씩 해결하는 사고 방법이다. 패턴인식 사고는 여러 문제들간 혹은 문제 내에서의 유사성을 찾는 것이다. 축약화 사고는 문제 해결과 무관한 내용을 제외하고 중요한 정보에만 초점을 두는 것을 말한다. 알고리즘은 문제 해결에 필요한 순차적인 사고 단계이며, 이러한 절차를 수행하는 과정에서 발생하는 오류를 찾아내고 새로운 방법을 찾아 적용해보는 활동이 디버깅 사고이다.

둘째, 교사들로 하여금 유아의 컴퓨팅 사고 향상을 위해 별도의 활동을 마련하기보다 기존의 교육과정에 관련 활동을 내재시켜 수업을 진행하도록 하였다(Yadav, Zhou, Mayfield, Hambrusch, & Korb, 2011). 수·과학 수업의 경우 컴퓨팅 사고를 향상시키기 위한 수업 전략으로 내재된 방식(embedded strategy)이 권장되고 있다(Weintrop et al., 2016; Yadav, Hong, & Stephenson, 2016). 해당 방식은 컴퓨팅 사고를 별도로 다루기에 충분한 활동이 개발되지 않은 교육 현장에 적합하며, 이에 교사들의 수업 부담도 경감된다. 또한 컴퓨팅 사고 향상을 위한 내용이 수·과학과 통합되어 진행된다면 모든 학생들이 수업에 참여할 수 있게 된다는 장점이 있다. 별도로 진행하여 흥미를 느끼지 못하는 학생들이 배제되는 경우를 방지할 수 있기 때문이다.

마지막으로, 교사들은 유아의 컴퓨팅 사고 역량을 측정하는데 사용하는 루브릭(Kwon, 2019)을 훈련받은 후 평가에 활용하였다. 해당 루브릭은 컴퓨팅 사고력을 유형별(문제분해, 패턴인식, 알고리즘, 축약화, 디버깅)로 구분하여 각각의 사고를 수준별(발현, 발달, 숙련, 완성 단계)로 측정하고, 그 이유를 기술할 수 있도록 구성되었다. 연구자들은 수업에서 교사들이 유아의 컴퓨팅 사고의 변화를 얼마나 정확하게 측정하는지 그 과정을 모니터링하면서 이들의 컴퓨팅 사고 이해 정도를 평가하였다.

3. 자료 분석

가. 내용 분석

만 5세의 수·과학 수업에서 교사의 컴퓨팅 사고 이해 수준을 탐색하고자 Krippendorff (2004)가 제시한 단위화, 표본추출, 기록과 코딩, 간소화, 귀추적 추론, 서사화 단계에 따라 수업 일지에 기록된 내용을 분석하였다(Choi, Jung, & Jung, 2016). 이 단계는 순차적으로 진행되지 않으며 필요에 따라 순환되었다. 수업 내용을 분석하기 위한 컴퓨팅 사고의 코딩 범주를 하위 요소인 문제분해, 패턴인식, 축약화, 알고리즘, 디버깅으로 설정하였다. 만 5세 담

임교사들이 작성한 수업 일지를 연구자들이 전체적으로 검토하면서 교사가 보고한 내용이 유아의 컴퓨팅 사고를 향상시키기 위한 수업 경험과 관련되어 있는지를 검토하였다. 또한 연구자들은 교사 면담을 통해 이들이 구술한 내용 중에서 의미 있는 진술들을 발견하여 의미를 구성하였다. 유아의 컴퓨팅 사고를 측정할 수 있는 루브릭을 활용하여 많은 양의 자료를 간소화한 후 귀추적 추론을 통해 맥락적 현상을 발견하고, 서사화 과정을 거쳤다.

나. 워드클라우드 분석

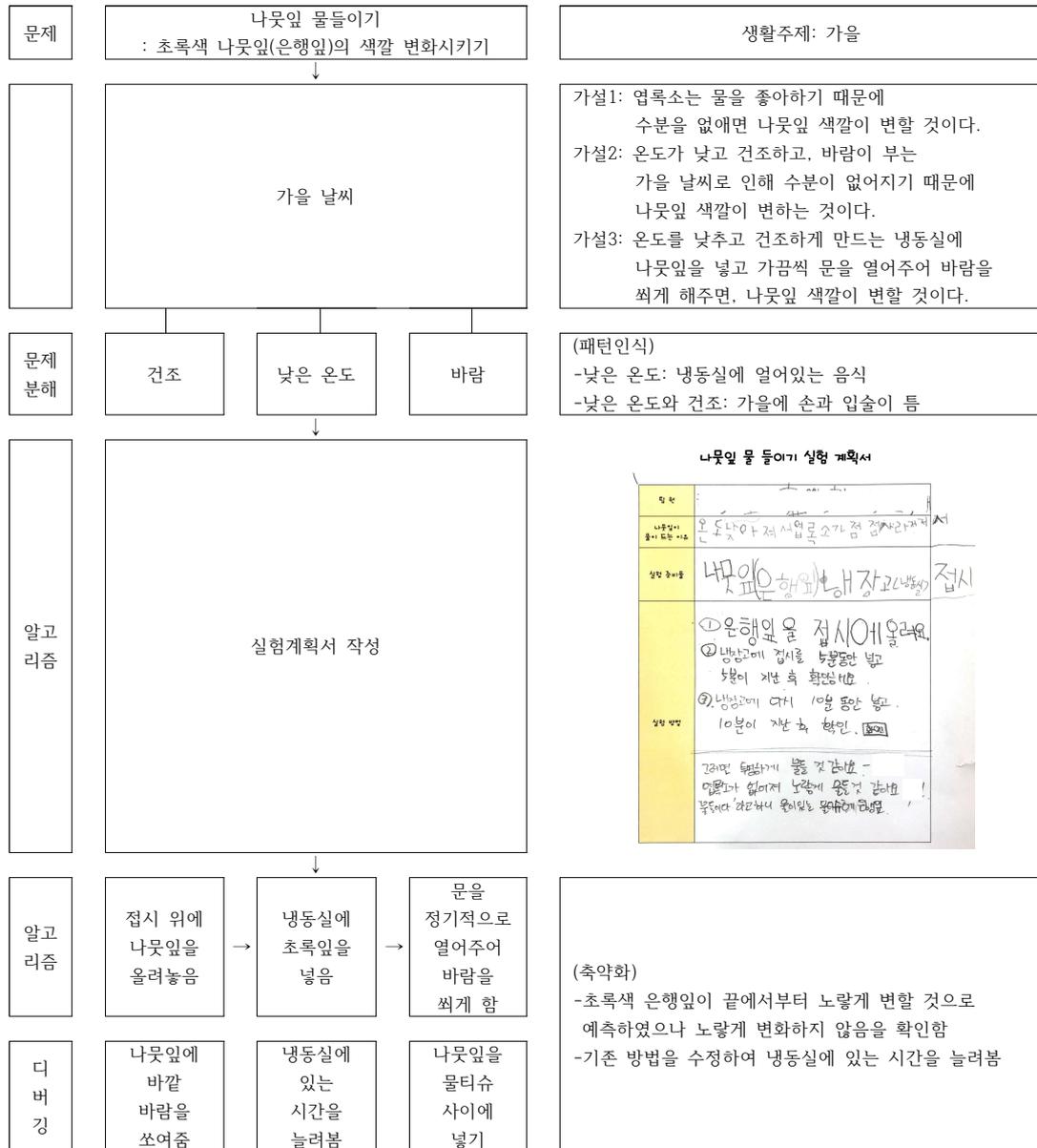
워드클라우드 기법은 출현 빈도가 높은 핵심어를 중심부에 큰 글씨로 표현해주어 직관적으로 의미를 파악할 수 있게 해준다. 본 연구에서 교사들로 하여금 만 5세 유아의 컴퓨팅 사고 향상을 위한 수업을 진행한 후 자신이 이해한 컴퓨팅 사고를 하위 요소별(문제분해, 패턴인식, 추약화, 알고리즘, 디버깅)로 정의해보게 하였다. 이들의 이해 수준을 한 눈에 파악할 수 있도록 시각화하기 위하여 워드클라우드를 활용하였다.

Ⅲ. 연구결과

1. 컴퓨팅 사고 향상을 위한 수·과학 수업

2018년 10월, ‘가을’이라는 생활주제로 진행된 수업을 컴퓨팅 사고의 하위 요소별로 구분하였다. [그림 1]은 만 5세 교실에서 교사로부터 엽록소의 특징과 잎의 색에 대한 설명을 들은 후 유아가 초록색 나뭇잎의 색을 변화시키기 위한 방법에 대해 논의해보는 상황을 구조화한 것이다. ‘문제에 대한 정의’ (나뭇잎 물들이기: 낙엽처럼 초록색 나뭇잎의 색깔을 변화시키기)가 교사에 의해 이루어진 후 유아들이 해당 문제를 해결하기 위한 방법을 예측해보면서 하위 요소를 다음과 같이 분해하기 시작하였다. 유아들은 ‘해결방법을 예측’하기 위해 나뭇잎 색깔이 변화하는 요소 중 ‘가을’이라는 계절의 특징에 초점을 두었다. 유아들은 가을이 옴으로 수반되는 날씨 변화에 대해 자신이 체감했던 건조, 낮은 온도, 바람이라는 세 가지 측면을 ‘패턴인식’ 사고로 도출하였다. ‘하위 과정(문제)으로 분해’하기 위해 건조와 낮은 온도, 바람에 초점을 두고 “나뭇잎을 냉동실에 넣는다”와 “바람을 쏘여 주기 위해 냉동실 문을 열어준다”라는 해결책을 제시하였다. 그 외 나뭇잎 물들이기라는 주제이기 때문에 ‘물’에 초점을 둔 유아는 “물티슈” 사이에 나뭇잎을 넣는 방법을 제안하기도 하였다. 유아들은 구체적인 실험 방법과 예측되는 결과까지 ‘알고리즘’ 사고를 통해 실험계획서에 작성하였다. 실험을 진행한 결과 예측되었던 변화가 일어나지 않았으

며, 오류를 수정해보는 ‘디버깅’ 사고를 통해 냉동실에 잎을 두는 시간을 늘려보기, 잎을 교실 밖에 놓아 바람을 쐬게 하기 등이 나타났으며, 일련의 과정을 통해 불필요한 내용들이 없어지고 핵심적인 내용만 남게 되는 ‘축약화’ 사고로 이어졌다.



[그림 1] 수·과학 수업에서 유아의 컴퓨팅 사고

2. 수·과학 수업에서 교사의 컴퓨팅 사고 변화

교사는 수업을 진행하면서 관찰된 유아의 컴퓨팅 사고를 일지에 기록하였다. 연구자는 수업 초반부에 교사들에게 생활주제와 무관하게 모든 유아의 컴퓨팅 사고를 기록하게 하였으며 중후반부부터는 생활주제를 중심으로 기록하게 하여 교사의 컴퓨팅 사고의 변화를 분석하였다.

가. 수업 초기에 관찰된 교사의 컴퓨팅 사고

교사들은 수업에서 컴퓨팅 사고 루브릭으로 만 5세 유아들의 컴퓨팅 사고 수준을 측정함과 동시에 관련된 유아의 활동을 일지에 기록하였다. 연구자들은 교사의 루브릭과 일지를 검토하면서 이들의 컴퓨팅 사고에 대한 개념 이해 수준을 평가하였다. 수업 초기에 교사는 컴퓨팅 사고를 구성하는 다섯 가지 하위 요소에 대해 오개념을 형성하거나 부분적으로 개념을 이해하고 있었으며, 개념을 이해하더라도 루브릭 수준을 적절하게 판단하지 못하는 경우가 관찰되었다. 특히 여러 개념 중 축약화의 경우 모든 교사가 이해하기가 어렵다고 토로하였으며, 알고리즘 사고의 경우 다른 사고에 비해 비교적 정확하게 이해하는 편이었다. 이를 종합하여 유형별로 분류하면 다음과 같다.

첫째, 교사가 용어에 대해 생소함을 느끼면서 이해하기 어려운 경우이다. 모든 교사들은 컴퓨팅 사고를 구성하는 다섯 가지 개념 중 특히 축약화 사고를 이해하는 데에 어려움을 토로하였다. 다른 개념에 비해 평소에 접해보거나 일상에서 사용하지 않는 용어이기에 생소함을 느꼈고, 수업과 연계하거나 관찰에 어려움을 겪었다. 교사의 일지를 분석한 결과, 유아의 축약화 사고를 구별해내거나 수업에 적용하는데 있어 오류가 확인되었다. 축약화 사고란 제시된 문제의 복잡성을 줄이기 위해 문제해결에 있어서 중요한 부분과 그렇지 않은 것을 구분한 후, 중요하지 않은 부분을 제거하고, 주어진 문제나 문제의 해결책의 핵심 요소를 추출할 수 있는 능력이다(Kwon, 2019). 이에 근거하면, 곡류를 종류별로 분류하기 위해 유아들이 도구를 선택하는 과정(문제 해결 과정)에서 곡류 크기에 초점을 두게 되는 사고 활동을 기술해야 한다. 그런데 교사는 곡류 분류를 위한 문제해결책 도출(도구)에만 초점을 두었으며, 이를 축약화 사고로 판단하였다.

유아들은 바구니에 담긴 곡류(콩, 팥, 쌀)를 분류하는 과정에서 수저를 사용하였는데, 수월하게 고르기 힘든 문제 상황에 직면하였다. 교사는 직조형태의 바구니를 제안하였고, 유아들은 이를 활용하여 쌀을 거른 후 바구니에 남은 콩, 팥을 분류하기 시작하였다. (교사 1)

둘째, 교사가 개념에 대한 부분적 이해를 형성한 경우이다. 예를 들면, 문제분해 사고를 하는 유아는 가장 먼저 문제를 정의한 후, 성공적인 문제 해결방법을 예측하고, 성공적 해결을 위해 정의된 문제를 하위 요소들로 분해하는 활동을 한다. 다음은 교사가 이러한 세 가지 요소 중 일부에 초점을 두고 기술한 내용이다. 교사의 일지를 분석한 결과, 유아는 거북이의 언어를 이해하기 위해 언어탐지기라는 문제해결책을 도출하였으나 언어를 탐지하기 위해 구체적으로 어떤 하위 요소들이 요구되는지 분해하는 활동이 누락되어 있다. 단지 설계도를 그리고, 이를 미술재료를 활용하는 방법에 초점을 두고 있었다.

김○○은 거북이를 관찰하던 중 거북이 언어탐지기 필요성을 느꼈으며, 미술재료로 만드는 방법을 생각하고, 설계도를 그리기 시작하였다. 교실에 있는 미술재료를 사용하여 설계도대로 거북이 언어탐지기를 만들었다. (교사 2)

셋째, 유아의 컴퓨팅 사고 수준을 루브릭에 한 단계 낮게 혹은 높게 평정하는 경우이다. 교사는 컴퓨팅 사고의 하위 요소를 구분하면서 동시에 발현, 발달, 숙련, 완성 등 수준을 선택해야 한다. 예를 들면, 교사 2는 유아가 언어탐지기의 설계도를 그리고 그에 따른 제작 활동을 수행하여 창의적 결과물을 도출하는 일련의 활동을 관찰한 후 이에 대해 알고리즘 사고로 정확하게 판단하였다. 그런데 교사는 유아가 문제 상황의 의미를 스스로 정의하는 모습을 관찰하였으나 흥미를 확장시키는지 여부를 구분하지 못하고 이를 숙련 단계로 체크하였다. 유아가 스스로 언어탐지와 관련된 지식을 대입하면서 흥미를 확장시키는 모습이 관찰되었기에 이는 한 단계 높은 완성 단계로 평정되어야 한다.

마지막으로, 알고리즘 사고의 경우 대부분의 교사들이 비교적 정확하게 이해하는 편이었다. 알고리즘 사고란 문제 해결을 위한 과정을 순서가 있는 절차로 나열하고 문제 해결 단계에 적절하게 배치할 수 있는 능력을 말한다. 일반적으로 최적의 해결책을 찾기 위해 알고리즘 사고는 반복적으로 수행되는데, 유아 A가 거북이 언어탐지기를 만들기 위해 설계도를 그리고 제작하는데 일어난 일련의 사고 과정을 알고리즘 사고로 볼 수 있다.

나. 수업 중반부에 관찰된 교사의 컴퓨팅 사고

초기에 교사에게 관찰된 오류들이 연구자와의 조율 및 재연수 과정을 통해 점차적으로 줄어들면서, 교사의 컴퓨팅 사고는 연구자들의 이해 수준과 유사해지기 시작하였다. 이에 교사는 루브릭으로 유아의 컴퓨팅 사고 변화를 비교적 정확하게 측정하고 기록하기 시작하였다. 다음은 교사가 제시한 나뭇잎의 색변화라는 주제(문제)에 대해 유아는 호기심을 보이고 하위 과정들로 분해하기 시작하였다. 다음은 이러한 문제분해 사고와 함께 이전의 경험과

연관시켜 문제해결방법을 찾는 패턴인식 사고가 기록된 내용이다.

- 중략 -

(지난주 활동을 통해 유아들은 엽록소로 인해 색이 변하는 것에 대해서는 알게 되었음.)
교사가 초록색 잎을 어떻게 물들이 수 있을지 문제를 제시하였다. 유아 A는 물을 들이기 위해서는 엽록소를 없애야 한다고 하였다. 나뭇잎이 물드는 가을에는 추워지면서 건조해지고 손이 튄다는 친구의 말을 듣고, 엽록소를 없애려면 건조해야 한다고 이야기하였다. 유아 B가 나뭇잎을 건조하게 만들며 촉촉하게 만들 수 있는 방법을 모색하던 중 냉동실에 넣어 놓았던 음식이 얼었던 것을 회상하며 냉동실은 음식을 얼릴 정도로 춥기 때문에 냉동실에 나뭇잎을 넣어 나뭇잎을 물들이자고 방법을 제시하였다. 더불어 가을은 바람이 불기 때문에 냉동실에 넣은 나뭇잎에 바람도 쐬어주어야 할 것 같다고 말하였다. (교사 3)

교사가 대집단 활동에서 나뭇잎과 관련된 동요의 가사에서 반복되는 문장 구조(‘오늘 아침 ~에 ~이요’, ‘~하면서 발발 땀니다’)를 밑줄로 표시해주는 도움을 주었고, 유아들은 중요한 부분과 그렇지 않은 부분을 구분하는 축약화 사고를 기록한 내용이다.

- 중략 -

교사: 가사를 보면 노래 가사가 반복 돼서 표현되고 있어.
유아: 2연?
교사: 동시 시간에 했던 것을 기억하고 있었구나. 그렇기도 해. 그런데 잘 보면 비슷한 말이 반복되고 있는데, 찾을 수 있겠니?
(유아들이 먼저 찾지 못하는 모습이어서 ‘나뭇잎’, ‘산새들’에 동그라미를 칩.)
C: 어! 또 찾았어요. ‘소근소근’ 이랑 ‘재잘재잘’ 부분도 비슷한데요?
교사: 그래, 그런데 또 있어. 찾아볼까?
교사: 음, ‘어제 저녁 바람’ 거기는 그냥 똑같고…… ‘담 밑에’ 여긴가? (교사 4)

유아들은 알고리즘 사고를 하면서 또래들과 협의하여 실험계획서를 작성하였고, 절차에 따라 진행하면서 오류를 경험하고 수정하는 과정에서 디버깅 사고를 하였다.

- 중략 -

냉동실에 넣어 물들이자는 가설에 동의한 유아들끼리 모여 그룹을 만들었고 실험 계획서를 작성하였다. 실험 방법을 작성하며 유아 C가 첫 번째 순서로 접시에 나뭇잎을 넣어야 한다고 이야기했고 교사는 이유를 물었다. “냉동실에 넣었을 때 나뭇잎이 얼어서 안 떨어질 수도 있잖아요.” 라고 이야기 한 후 다음 방법을 이어 작성하였다. 옆에 앉은 친구가 냉동실에 5분마다 변화가 있었는지 확인해보는 방법을 제시하여 내용을 추가하여 실험방법을 완성하였다. 반면 다른 그룹은 뜨거워야 건조해지고 엽록소가 없어질 것이라고 이야기하고 뜨겁게 오븐이나 전자레인지에 넣어 열을 가하기로

하자 유아 C는 “오븐이나 전자레인지에 그냥 넣으면 나뭇잎이 타버릴 것 같아.” 라고 이야기했고 “보호할 수 있는 걸 생각해 봐.” 라고 말하며 다른 팀에서 타지 않을 방법을 생각할 수 있도록 말하였다. 실험을 시작하고 냉동실에 은행잎을 넣었으며, 이후 시간에 따라 변화를 관찰하였다. (교사 5)

다. 수업 후반부 교사의 컴퓨팅 사고 수준

연구가 종료된 후 2019년 2월에 교사들의 컴퓨팅 사고 이해 수준의 변화를 측정하기 위해 수업 전과 후를 구분하여 문제분해, 패턴인식, 알고리즘, 축약화, 디버깅 사고별 이해 수준을 5점 척도(1:매우 낮은 편, 2:낮은 편, 3:보통, 4:높은 편, 5:매우 높은 편)로 표시하게 하였으며, 주관식으로 각 요소별 개념 정의를 기술하게 하였다. 컴퓨팅 사고력 이해 수준의 변화를 양적 분석한 결과는 다음과 같다.

첫째, 수·과학 수업 전후로 컴퓨팅 사고에 대한 교사의 이해 수준은 각각 평균 2.2점과 3.3점으로 1점 상향되어 보통 이상의 이해 수준을 형성하였다. 하위요소별로 살펴보면, 문제분해의 경우 전후 2.2점에서 3.2점, 패턴인식의 경우 전후 2.2점에서 3.2점, 알고리즘의 경우 전후 2.8점에서 3.6점, 축약화의 경우 전후 1.4점에서 2.8점, 디버깅의 경우 전후 2.6점에서 3.6점으로 모두 상향되었다. 둘째, 수업 전후를 통틀어 평균보다 높은 점수를 유지한 컴퓨팅 사고는 ‘알고리즘’ 이었다. 이는 수업 초기에 이루어진 질적 분석 결과에서도 알고리즘 사고에 대한 교사 이해도가 높게 나타난 바 있다. 셋째, 축약화는 다른 하위 요소들과 비교했을 때 가장 낮은 이해도를 보였지만 수업 전후를 비교하였을 때 가장 큰 폭(1.4점)으로 점수가 증가하는 특징을 보였다.

교사의 컴퓨팅 사고의 변화를 전문가의 개념과 얼마나 일치하며 변화했는지를 분석하였다. 연구자는 교사에 의해 기술된 내용을 (1)루브릭에 따라 컴퓨팅 사고의 수준을 정확하게 평정하였는지, (2)컴퓨팅 사고의 하위 요소에 해당하는 유아의 행동을 정확하게 기술하였는지와 같은 기준으로 검토되었다. 대체적으로 교사는 컴퓨팅 사고의 개념을 어느 정도 이해하고 있었으며, 관련된 활동들을 구체적으로 잘 기술하는 편이었다. 그러나 루브릭을 활용한 컴퓨팅 사고의 하위 요소별 수준을 판단하는데 있어서 연구 초기에 한 단계 정도 높게 평정하는 경향을 보여 연구자에 의해 일부 하향 조정되었다. 교사와 연구자의 평정자간 신뢰도를 확보하기 위해 퍼센트 일치도를 계산한 결과 1반의 경우 66.7%였으며, 우연에 의해 일치될 확률을 제거하기 위하여 Cohen's Kappa를 구한 결과 0.488로 0.41-0.60에 있어 적당한 일치도를 보였다. 2반의 경우 퍼센트 일치도는 80%, Cohen's Kappa은 0.695로 0.60-0.80에 있어 상당한 일치도를 보였다.

문제분해에 대해 교사들의 개념 이해를 정성적으로 분석하기 위해 워드클라우드를 활용하였으며, [그림 2]와 같다. 핵심어를 활용하여 교사들의 개념 이해를 종합하면, “문제분해

란 문제를 인식하고 또는 호기심/관심/궁금증을 갖고 작은 부분/다양한 방법으로 나누는 것”으로 정의되었다. 교사별로 정의된 내용은 다음과 같다.

- 문제를 인식하고 문제 해결 방법을 찾아 해결하는 과정 중 문제를 작은 부분으로 나누는 것 (교사 1)
- 어떤 문제가 주어졌을 때 호기심을 가지고 원인과 가설을 생각해보고 다양한 해결방법들을 생각해 보는 과정 (교사 2)
- 유아가 문제 상황에 대한 관심을 가지고(문제 해결이 필요하다는 문제인식) 문제를 보다 성공적으로 해결하기 위해 작은 부분, 단계로 나눌 수 있는 능력 (교사 3)
- 어떤 사물이나 주제에 관심을 갖고 다양한 방법으로 탐색하는 것 (교사 4)
- 유아가 문제 상황에 직면했을 경우, ‘왜?’라는 질문을 가졌을 때 그것의 지속기간이 짧고 깊이가 얕은 호기심부터, 지속 기간이 길고 깊이가 깊은 호기심까지 모든 궁금증에 대해 생각하는 것 (교사 5)

패턴인식에 대해 교사들의 개념 이해를 정성적으로 분석하기 위해 워드클라우드를 활용하였으며, [그림 2]와 같다. 핵심어를 활용하여 교사들의 개념 이해를 종합하면, “패턴인식이란 이전 / 유사한 경험을 떠올려 / 연관지어 / 대응시켜 해결방법을 생각 / 모색하는 것”으로 정의되었다. 교사별로 정의된 내용은 다음과 같다.

- 문제 해결과정에서 자신의 유사한 경험과 연관지어 문제해결방법에 적용하는 것 (교사 1)
- 문제를 해결하기 위해 이전 경험을 떠올리며 다양한 탐구방법들을 사용해 문제를 해결할 수 있는 방법이 괜찮은 방법인지 생각해 보는 과정 (교사 2)
- 문제 분해 단계에서 도출된 하위 개념 중 보다 능숙하고 효과적으로 문제를 해결할 수 있도록 이전 경험에 근거한 지식을 활용하는 능력 (교사 3)
- 이전 경험을 토대로 일정한 규칙이나 흐름을 파악하는 것 (교사 4)
- 유아가 직면한 문제 상황과 관련되건 관련되지 않건 간에, 유아 스스로 생각했을 때 연결고리가 있는 이전의 경험을 회상하고 현재의 상황과 1:1 대응시켜 해결 방법을 모색해보는 것 (교사 5)



[그림 2] 문제분해(좌)와 패턴인식(우) 워드클라우드

알고리즘에 대해 교사들의 개념 이해를 정성적으로 분석하기 위해 워드클라우드를 활용하였으며, [그림 3]과 같다. 핵심어를 활용하여 교사들의 개념 이해를 종합하면, “알고리즘이란 해결방법을 수행 / 진행하기 위한 순서 / 순차적인 단계” 로 정의되었다. 교사 개별로 정의한 내용은 다음과 같다.

- 문제를 해결하기 위해 일의 순서를 순차적인 단계로 수립하는 것 (교사 1)
- 자신이 수립한 해결방법을 수행하기 위해 순차적으로 방법을 나열해 보거나 직접 순차적으로 해 보는 과정 (교사 2)
- 문제를 해결하기 위한 과정 중 분해한 하위 개념들을 적절한 순서에 맞추어 진행할 수 있는 능력 (교사 3)
- 해결방법을 찾기 위한 순서를 설정하는 것 (교사 4)
- 문제 해결 과정에 지속적인 관심과 호기심을 가지고 참여하며, 그 문제 해결 과정을 논리적으로 생각하고 정리하는 과정 (교사 5)

축약화에 대해 교사들의 개념 이해를 정성적으로 분석하기 위해 워드클라우드를 활용하였으며, [그림 3]과 같다. 핵심어를 활용하여 교사들의 개념 이해를 종합하면, “축약화란 중요한 / 의미 있는 / 필요한 또는 중요하지 않은 / 사용하지 않아도 될 / 불필요한 요소를 구분하는 / 걸러내는 / 제거하는 / 생략하여 / 찾는 / 정리하는 과정” 으로 정의되었다. 교사별로 정의된 내용은 다음과 같다.

- 문제 해결에서 중요한 요소와 중요하지 않은 요소를 구분하여 문제 해결 과정에 적용하는 것 (교사 1)
- 알고리즘을 세워 순차적으로 해결하는 도중 사용하지 않아도 될 방법이나 중요하지 않은 과정을 걸러내는 과정 (교사 2)
- 최상의 문제 해결 과정을 위해 다양한 해결방법 중 불필요한 과정을 제거할 수 있는 능력. 경험과 지식으로 습득한 하위 단계를 생략하여 필요한 해결 방법만 거칠 수 있는 능력 (교사 3)
- 여러 정보 중 자신에게 의미 있는 내용을 찾아 적용해 보는 것 (교사 4)
- 문제 해결 과정에서 알고리즘 작성을 위한 패턴을 찾아 목록화하는 것 (교사 5)



[그림 3] 알고리즘(좌)와 축약화(우) 워드클라우드

디버깅에 대해 교사들의 개념 이해를 정성적으로 분석하기 위해 워드클라우드를 활용하였으며, [그림 4]와 같다. 핵심어를 활용하여 교사들의 개념 이해를 종합하면, “디버깅이란 오류 / 실패 요인을 수정 / 보완 / 개선안 / 다른 방법 / 새로운 방법을 제시하고 적용하는 것”으로 정의되었다. 교사별로 정의된 내용은 다음과 같다.

- 문제 해결 과정 중 오류가 생겼을 때 오류를 인식하고 새로운 문제 해결 방법을 제시, 적용하는 것 (교사 1)
- 자신이 생각한 방법대로 행하였으나 실패했을 때 실패 요인을 생각해 보완하여 해결하는 과정 (교사 2)
- 유아가 세운 알고리즘으로 사고하는 과정 중 의도치 않은 상황이 생겼을 경우 오류를 정확하게 인지하고 개선안을 제시하여 적용시킬 수 있는 능력 (교사 3)
- 문제의 오류를 수정하고 다른 방법으로 접근해 보며 해결하는 과정 (교사 4)
- 오류 해결을 위해 이전에 사용하지 않았던 새로운 방법을 생각할 뿐 아니라 문제에 적용 시도하는 것 (교사 5)



[그림 4] 디버깅 워드클라우드

IV. 결론 및 제언

1. 결론

지능정보사회에서 학교교육의 변화로 인해 교사에게 교수학습 측면의 다양한 역량이 요구되고 있다(KICE, 2019; Lim, Ryu, & Kim, 2017). 본 연구는 A어린이집의 만 5세 교사를 대상으로 미래사회에 요구되는 핵심역량 중 하나인 컴퓨팅 사고의 중요성을 인식하게 함과 동시에 교수학습 활동에 관련 개념을 내재시켜 적용해보도록 함으로 실천적 시사점을 제공하

였다는데 의의가 있다. 또한 수업에서 교사가 유아의 컴퓨팅 사고 변화를 측정하기 위해 사용한 루브릭과 질적 자료를 통해 이들의 컴퓨팅 사고에 대한 개념 이해 정도를 직접적으로 평가했다는데 의의가 있다.

연수 전 교사의 컴퓨팅 사고력은 평균 2.2로 보통 이하였는데 그 이유는 다음과 같다. Jo et al.(2017)은 유치원 교사를 대상으로 컴퓨팅사고 향상을 목적으로 하는 유아소프트웨어 교육의 필요성에 대해 설문조사하였는데, 이들은 보통의 인식 수준과 낮은 관심도를 보이고 있으며, Park et al.(2018) 연구에서도 관심 단계 중 가장 낮은 지각 수준에 머물러 있음이 확인되었다. 대부분의 교사들은 유아소프트웨어 교육에 대한 경험이 없기 때문에 비사용자 패턴을 보였다고 할 수 있다.

교사 연수를 통해 문제분해, 패턴인식 등 관련 개념들의 이해와 교수학습 활동에 적용 및 평가 등을 지원하였다. 이러한 개념 이해, 수업 적용, 유아 평가 활동에 이르는 일련의 연수 과정을 통해 교사의 컴퓨팅 사고가 평균 3.3으로 보통 이상의 이해 수준을 형성하였다. 기존 연구들은 유아소프트웨어 교육과 관련된 연수 프로그램을 15차시로 구성하여 개발한 바 있으나 현장에 적용하여 그 효과성을 검증해보지 못했다는 한계를 가지고 있다(Hong et al., 2017). Jung & Park(2018)의 경우 생활도구 주제로 수·과학 중심의 언플러그드 활동을 설계하여 교사로 하여금 수업에 적용해보게 하였고, 그 효과를 분석한 바 있으나 기존 수업과 연계보다는 초등에서의 소프트웨어 교육처럼 코딩교육에 초점을 두고 진행되었다는데 한계가 있었다. 본 연구의 경우, 만 5세 교사 5명을 대상으로 하였기에 연구결과를 일반화하는데 어려움이 있다. 그러나 이러한 교사 연수를 통해 미래사회에 요구되는 핵심역량인 컴퓨팅 사고 개발의 중요성 인식 및 핵심개념을 이해하게 한 후 기존 수업에 적용할 수 있도록 유도하였으며, 수업 적용 후 유아의 변화를 측정하게 함으로 교사의 컴퓨팅 사고에 대한 개념 이해를 실제적으로 평가하였다는데 의의가 있다.

2. 제언

교사의 컴퓨팅 사고 역량 제고를 위해 다음과 같은 제언을 하고자 한다.

첫째, 유아교육에서 현직교사를 대상으로 체계적 연수가 시행되어야 한다. 현직교사를 대상으로 할 때 연수 방식과 연수 내용에 대한 고려가 필요하다. 소프트웨어 교육을 진행하는 주요 8개국이 ‘2017 ICT4ED 국제전문가 회의’에서 현직교사 연수방법으로 중앙집중식과 분산식으로 구분한 바 있다(KERIS, 2017). 중앙집중식 접근은 우리나라가 채택한 정부 주도 방식으로 단시간에 다수의 교사에게 필요한 기초 소양을 제공할 수 있다. 분산식 접근은 미국과 영국 등 대부분의 참여국에서 수행되는 방식으로, 교사 및 학교의 상황과 요구에 따라

적합한 연수를 제공할 수 있다. 이에 중앙집중식에서 발생할 수 있는 현장과의 격차를 줄일 수 있으며, 교사가 자발적으로 소프트웨어 교육에 필요한 지식과 자료를 공유하는 협력적 학습문화가 형성될 수 있다. 유아교육의 경우 현직교사를 대상으로 각 기관의 상황에 적합한 분산적 접근 방식을 채택하는 것이 융통성을 가질 수 있을 것이다.

둘째, 현직교사와 함께 예비교사를 대상으로 한 소프트웨어 교육이 필요하다. 국가 수준에서 초중등 현직교사를 대상으로 교원연수가 이루어지고 있으나 보다 중장기적 관점에서 예비교사에게 컴퓨팅 사고력을 향상시키고, 자신의 전공 교과분야에 적용시킬 수 있도록 교직과목으로 파이프라인 역할을 수행토록 해야 한다(Yadav, Gretter, Good, & McLean, 2017). 예를 들어, 교육심리 과목에서 문제해결과 관련된 내용을 다룰 때 컴퓨팅 사고에 대한 교육이 가능하며, 교육공학 과목에서 공학을 활용한 문제해결과 관련된 내용을 다룰 때 적용가능하다(Mouza, Yang, Pan, Ozden, & Pollock, 2017). 예비유아교사의 소프트웨어 교육에 대한 관심도 수준 측정(Jung, 2017)과 같은 연구는 수행되고 있으나 보다 근본적으로 교직과목에서 교사에게 요구되는 지능정보역량 향상을 위해 컴퓨팅 사고와 같은 내용을 필수 지식으로 다루어야 할 것이다.

References

- Bers, M. (2017). *Coding as a playground*. New York: Routledge.
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. In Annual AERA meeting, Vancouver, BC, Canada.
- Choi, S., Jung, J., & Jung, S.(2016). Concept and procedures of qualitative content analysis. *Journal of Qualitative Inquiry, 21*(1), 127-155.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking: A guide for teachers*. Computing at School.
- Hong, K., Jo, J., & Park, C. (2017). Development of software education training program for early childhood teachers. *Journal of Learner-Centered Curriculum and Instruction, 17*(1), 469-494.
- ISTE & CSTA. (2011). Operational definition of computational thinking for K-12 education.
- Jo, J., & Hong, K. (2017). A study of the direction for developing software curriculum for preschoolers based on Nuri-Curriculum. *Journal of Learner-Centered Curriculum and Instruction, 17*(15), 557-583.
- Jo, J., Park, C., & Hong, K. (2017). Awareness and needs for early childhood software education in early childhood teachers. *Journal of Learner-Centered Curriculum and Instruction, 17*(3), 83-106.
- Jung, J. (2017). Analysis on pre-service early childhood teachers' stage of concerns about software education According to the CBAM. *Journal of the Korea Academia-Industrial Cooperation Society, 18*(7), 431-440.
- Jung M., & Park, S. (2018). The effects of STEAM activities using unplugged computing on young children's creativity and problem solving ability. *Journal of Learner-Centered Curriculum and Instruction, 18*(3), 705-724.
- KERIS (2017). *ICT4ED 2017: case study on the SW education in major countries and implications*. Korea Education and Research Information Service (RM 2017-10).
- KICE (2019). *An exploratory study on the teacher competencies in the 4th industrial revolution*. Korea Institute of Curriculum & Evaluation Brief (2019.10.5).
- Kim, J., Hong, I., & Kim, K. (2016). *A study on teaching materials development for computational thinking through play*. In proceedings of the Korean association of computer education (pp. 187-190).

- KOFAC (2015). *Research on implementation of ICT-SW education in primary and secondary school*. Korea Foundation for the Advancement of Science and Creativity Report (2014-0-00238).
- Krippendorff, K. (2004). *Content analysis: An introduction to its methodology* (2nd ed.). Thousand Oaks, CA: Sage.
- Kwon, S. (2019). The effect of phenomenon-based STEM education on 5-year-old children's computational thinking. *The Journal of Educational Research*, 172, 83-106.
- Lim, J., Ryu, K., & Kim, B. (2017). An exploratory study on the direction of education and teacher competencies in the 4th industrial revolution. *The Journal of Korean Education*, 44(2), 5-32.
- MOE (2018). *Direction of teacher training (2019)*. Ministry of Education press release (2018.10.10).
- Mouza, C., Yang, H., Pan, Y-C., Ozden, S., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of TPACK. *Australians Journal of Educational Technology*, 33(3), 61-76.
- MSIT (2017). *We will be responsible for software education*. Ministry of Science and ICT press release (2017.7.21).
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Park, S., Jung, J., & Kang, M. (2018). Analysis on kindergarten teachers' stage of concerns about SW education: an application of the CBAM. *Journal of the Korea Academia-Industrial Cooperation Society*, 19(2), 462-471.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147.
- Wing, J. (2006). Computational thinking. *Communications of The ACM*, 49(3), 33-35.
- Wing, J. (2011, March 06). *Computational thinking: What and why*. Retrieved from <http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>.
- Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In Rich, P. and Hodges, C. (Eds), *Emerging research, practice, and policy on computational thinking* (pp. 205-220). Springer International Publishing.
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *Tech*

Trends, 6(6), 565-568.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, 6(4), 55-62.

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). *Introducing computational thinking in education courses*. In Proceedings of ACM Special Interest Group on Computer Science Education, Dallas, TX.

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, 5(4), 562-590.