

초등정보교육에서의 계산적사고 교육을 위한 언플러그드 컴퓨팅 방법에 관한 고찰

한병래

진주교육대학교 컴퓨터교육과

요 약

컴퓨터과학은 정보처리의 연구, 자동화될 수 있는 것에 대한 연구, 컴퓨터계산의 연구, 자연적 인공적 모두의 정보처리의 연구로 발전하고 있다. 컴퓨터과학의 발전에 따라 정보교육의 모습도 변해 왔고, 최근에는 계산적사고 교육을 강조하고 있다. 본 논문은 정보교육의 발전과정을 살펴보고, 계산적사고의 교육방법으로 언플러그드 컴퓨팅을 적용하는 것에 대해 탐색해 본다. 연구자는 또한 언플러그드 컴퓨팅 방법을 적용하는데 있어서의 고려사항을 교사준비, 교육내용, 교수자원의 측면으로 나눠 살펴본다. 교사준비 측면에서는 교사의 인식전환, 컴퓨터과학 내용의 이해, 학생의 사고 촉진방법에 대한 이해가 필요하고, 내용 측면에서는 추상화 수준 설정, 내용 난이도 조절, 내용 심화 및 통합이 필요하다. 그리고 교수자원의 측면에서는 시간관리 및 자료준비 방법에 대한 고려도 필요하다.

키워드 : 정보교육, 컴퓨터과학교육, 계산적사고, 언플러그드 컴퓨팅, 초등정보교육

The Research of Unplugged Computing Method for Computational Thinking in Elementary Informatics Education

Byoungrae Han

Chinju National University of Education, Dept. of Computer Education

ABSTRACT

Computer Science has been developed thru the study of information processing, the study of what can be automated, the study of computation, and the study of information processes both natural and artificial. As developing the computer science, the aspect of informatics education was changed. Recently we focus on computational thinking education. In this paper, I review the progress of informatics education and research the applying of unplugged computing as a method of computational thinking education. I also research the considerations about applying of unplugged computing in aspects of teacher preparation, educational content, and teaching resources. As result I found that teachers change their perceptions about computer education, understand the computer science contents, and understand the promotion of students thinking skills in point of view at teacher preparation, that teachers can set the levels of abstraction and can enrich and integrate of contents, and that teacher can manage class time and educational resources preparation in point of view at educational resources.

Keywords : Informatics education, Computer science education, Computational thinking, Unplugged computing, Elementary Informatics education

이 논문은 2010학년도 진주교육대학교 가정학술연구재단의 지원을 받아 작성된 것임.

논문투고 : 2013-04-12

논문심사 : 2013-04-13

심사완료 : 2013-06-04

1. 연구의 필요성

컴퓨터과학은 정보처리의 연구, 컴퓨터의 환경에 대한 현상연구, 자동화 될 수 있는 것에 대한 연구, 컴퓨터계산의 연구로, 최근에는 자연적 및 인공적인 상황에서의 정보처리의 연구로 발전하고 있다[18]. 변천과정에서 용어의 의미도 일부 변화되고, 시기에 따라 관심의 중심 분야도 바뀌어 왔다. 일반인들에게 알려진 컴퓨터과학 분야는 컴퓨팅공학, IT, ICT 등의 다양한 이름으로 불려왔다. 용어의 혼란에 대해 유럽을 중심으로 정보학(Informatics)이라는 용어로, 미국을 중심으로는 컴퓨팅이라는 용어로 사용한다[22]. 본 연구에서 언급하는 컴퓨터과학은 컴퓨터과학, 컴퓨팅 및 정보학을 의미한다.

컴퓨터과학의 발전과정을 살펴보면 초기에는 수학, 공학, 과학의 세 기둥이 컴퓨터과학을 이끌었고, 현재에도 큰 영향을 미친다. 그래서 컴퓨터과학에 필요한 사고는 수학, 공학, 과학의 교과에서 길러 줄 수 있을 것이라고 생각할 수 있으나 컴퓨터과학은 발전과정에서 독자적인 사고방법을 발전시켜왔다.

수학교과에서 수학적 사고를 강조하고, 과학교과에서 과학적 사고를 강조하듯이 정보교과에서는 교과 특성의 반영하는 사고력교육을 강조할 필요가 있다. 우리는 알고리즘적 사고를 컴퓨터과학의 대표사고로 인식하여왔다.

Wing은 컴퓨터과학교육 및 모든 교육에서 계산적 사고의 중요성을 주장하며, 계산적사고는 모든 이를 위해 필요하다고 주장한다[25]. 보통교육에서의 기본 교육 3R(읽기, 쓰기, 셈하기)에 포함하여 4번째 요소로 계산적사고가 들어가야 한다고 주장한다. 한국에서는 “ICT를 활용한 문제해결력 향상 교육”을 강조하였지만, 교육현장에서는 ICT를 활용하는 것에 치우친 반면 문제해결력을 길러주는 데에는 한계가 있었다. 제7차 교육과정에서 ICT활용 교육의 목적을 살펴보면 ICT를 활용하는 것이 목적이라기보다는 문제해결력을 길러주는 것이 더 중요한 목적이었다. “ICT를 활용한 문제해결 능력의 향상”은 ICT활용보다는 문제해결 능력 향상에 초점을 맞춰야 한다. 학교 현장에서 ICT 활용 부분을 더 강조하게 된 것은 학교교육에서는 문제해결능력을 길러주는 것을 당연히 강조

되는 것으로 인식하기 때문이다.

컴퓨터과학의 교과명을 정보로 바꾸고 있는 것은 교육의 중심이 컴퓨터를 둘러싼 현상에서 그 현상을 포함한 정보처리과정 전반에서 필요한 사고력교육에 초점을 맞추려는 것으로 이해할 수 있다.

컴퓨터과학 발전과정에는 컴퓨터과학교과의 사고를 길러줄 수 있는 학습 재료가 풍부하다. 또한, 학문 분야의 내용과 사고의 형식은 밀접하게 관련되어 있기 때문에 컴퓨터과학 내용의 교육이 계산적사고 교육에 많은 도움을 줄 수 있음을 알 수 있다. 구체적 조작단계의 초등학생들은 구체물과 상황 및 활동을 통해 학습하는 것이 좋다. 언플러그드 컴퓨팅 방법은 체험 활동과 구체물 사용의 방법을 통해 학생들에게 컴퓨터과학내용에 기반한 문제해결 상황을 해결함으로써 계산적사고를 기를 수 있는 기회를 제공할 수 있다.

본 연구에서는 계산적사고의 개념, 정보교육의 발전과정, 계산적사고 교육의 필요성, 일상에서 일어나고 있는 계산적사고의 모습들을 알아보려고 한다. 또한 초등정보교육에서 계산적사고를 기르기 위해 언플러그드 컴퓨팅 교육방법은 어떻게 적용되어야 하는가에 대해서도 살펴본다.

2. 이론적 배경

2.1 정보교육발전과정

정보교육의 발전과정은 컴퓨터의 발전과정 및 사회적 환경과 밀접한 관련을 맺고 있다. 정보교육은 사회에서의 영향 및 인식의 변화에 따라 전산교육, 컴퓨터교육, ICT활용교육, 정보교육으로 그 명칭이 변해왔다. 학교 정보교육의 변천과정을 교육과정철학을 중심으로 살펴보면 인식중심의 교육, 프로그래밍 중심 교육, 응용소프트웨어 활용 중심 교육, 멀티미디어 활용 교육, ICT 활용 교육의 순서로 발전하여 온 것을 알 수 있다[7][8].

각 과정에서의 사회적 상황과 목적들을 다시 살펴보자. 첫째, 인식중심의 교육은 컴퓨터의 특성을 이해하는 것을 핵심목적으로 삼는다. 당시의 상황을 고려해 보면 컴퓨터의 성능과 잠재력은 사회에 상상 이상

의 충격을 주었고, 대중에게 전달된 컴퓨터의 능력은 증폭되었다. 교육자들은 부풀려진 컴퓨터에 대한 잘못된 이해를 정확하게 바로잡는 교육이 필요했다고 판단할 수 있다.

둘째, 1970년대 후반의 프로그래밍 중심의 교육은 상용화된 응용프로그램의 부족으로 자신이 직접 소프트웨어를 개발해야 하는 사회적 상황과 깊은 관련이 있다. 당시 인식 중심의 교육과정으로 인해 많은 사람들이 컴퓨터에 대해 이해는 어느 정도 하고 있었으나, 컴퓨터를 자신의 일에 활용하여 문제를 해결하는데에는 어려움이 많았다는 것을 알 수 있다. 그래서 소프트웨어를 개발하기 위한 당시의 프로그래밍 교육은 당연한 것이었다.

셋째, 1980년대 중반의 응용소프트웨어 활용중심의 교육[7]을 살펴보자. 1970년대 프로그래밍 교육을 통해 다수의 소프트웨어들이 개발되었다. 개발된 소프트웨어들은 사회에서 경쟁을 시작하였고, 경쟁에서 적자생존 한 소프트웨어는 시장을 지배하였다. 이 소프트웨어는 사용자에게 해당 분야의 표준을 제공하게 되었다고 이해할 수 있다. 개인의 입장에서 볼 때, 일상적인 일을 해결하기 위해서는 직접 소프트웨어를 개발하는 것보다 상업용 소프트웨어를 구입하는 것이 경제적이었다. 상업용 소프트웨어를 학교의 교육과정에 편입함으로써 프로그램 사용설명서와 교과서가 큰 차이가 없어지는 현상과 소프트웨어가 갱신되면 교과서가 갱신되어야 하는 현상도 나타났다.

넷째, 1990년대 멀티미디어 활용중심 교육[7]은 하드웨어의 급격한 발달로 인해 나타난 현상이다. 다중 감각을 통한 정보전달 방식으로서의 변화는 사용자를 폭발적으로 증대시키는 결과를 낳았다. 사용자들은 업무 및 문제해결 이외의 여가 및 오락 시간에도 컴퓨터를 활용하게 되었다. 이 시기의 관심은 저작도구를 이용하여 멀티미디어 자료를 동기화시켜 제시하는 것이었다. 교육의 측면에서도 다량의 정보를 제공하는 것에 대한 매력은 매우 큰 것이었다.

다섯째, 2000년대 인터넷과 다양한 형태의 정보기기의 보급으로 인해, 컴퓨터교육은 ICT활용교육으로 모습이 변화한다[8]. 컴퓨터를 포함하는 다양한 정보기기는 정보에 대한 접근을 용이하게 하는 측면이 있다. ICT활용교육 시기는 컴퓨터를 포함한 정보기기를

도구로 이용하여 자신의 업무능력을 확장하는 것이 중심이다. 교사들은 학생들의 업무인 학업에 ICT를 활용하여 학습의 효과를 높이려고 노력하였다. 그러나 학생이라는 특수한 시기는 지식을 습득하는 것보다 지식 습득 과정과 습득능력을 길러가는 과정에 초점을 맞추어야 한다. 결국 학생들의 생각하는 능력(사고력)을 개발하는 것이 가장 중요하다.

최근의 정보교육은 정보기기를 자신의 생활에 이용하는 것과 정보처리 과정에 대한 이해를 높이려는 것으로 변화하고 있다. 학생들은 정보처리를 위한 컴퓨터과학 발달과정에서 컴퓨터 과학자들이 경험한 것을 학습함으로써 컴퓨터 과학자들과 유사하게 사고할 수 있는 기회를 가지게 된다.

2.2 ICT활용 문제해결능력

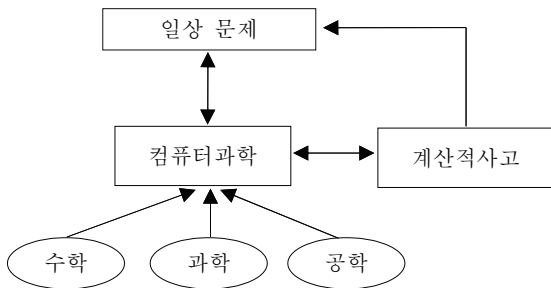
Denning의 주장에 따르면 컴퓨터는 도구이고, 컴퓨팅은 과학이라고 강조한다[18]. 컴퓨터를 안다는 것은 도구의 특성을 아는 것이고 컴퓨팅을 안다는 것은 도구를 자신의 일에 적절하게 사용할 수 있다는 것을 의미한다. 도구를 자신의 일에 적용하는 것에 있어 중요한 부분은 자신의 일을 파악하는 것이고 도구의 선정이나 적용 방법은 부수적인 문제이다. 한동안 ICT활용을 강조한 측면에는 도구에 대한 친근감을 길러주기 위해 일시적으로 도구의 사용을 더 장려한 측면이 있다. 교육의 궁극적 목적은 학습자 자신의 문제에 적용할 수 있는 능력 길러 주는 것이다. ICT를 활용한 문제해결능력의 핵심은 학습에서 단순히 ICT를 활용하는 것을 강조하는 것이 아니라, 자신의 업무(학업)에 주체적으로 ICT를 활용할 수 있어야 하는 것이라고 이해해야 한다. 자신의 일에서 ICT를 활용할 수 있는 것이 자연스러워지려면 학생의 업무인 학업에 ICT를 활용하는 것에서 시작하여 미래 자신의 업무에 자신이 능동적으로 ICT를 활용하는 것으로 확장 되도록 하는 것이 이 교육의 목적이라고 생각할 수 있다.

이러한 생각에 기반하여 접근하지 못하고 단순히 ICT를 활용하는 것에 중심을 두어 장려함으로써 지난 10년간의 막대한 비용의 투입에도 불구하고 궁극적인 목적에 도달하지는 못한 것 같다.

ICT활용 문제해결능력 교육과정에서 강조하고자 한 것은 컴퓨터를 사용하여 문제해결 능력을 길러 주는 것이 목적이었다. 그러나 교사들의 ICT활용능력의 부족 및 처음 만들어진 교육과정에 대한 이해의 한계로 인해 도구의 사용법을 익히 것과 도구 자체를 사용하고 있다는 것에 의의를 두었다.

2.3 계산적사고의 출현 배경

컴퓨터과학의 발전과정을 살펴보면 다수의 과학자, 공학자, 수학자들의 노력으로 많은 문제를 해결 해온 것을 알 수 있다. 수학, 과학, 공학의 사고들이 복합적으로 컴퓨터과학의 발전을 촉진하고, 반대로 영향을 주기도 하였다.



(그림 1) 수학, 과학, 공학과 컴퓨터과학의 관계

컴퓨터과학의 발전은 다양한 과학 분야에서 이용되었다. 컴퓨터계산 과학(Computational Science)은 과학을 하는 전통적인 이론과 실험의 방법에 새로운 방법을 추가하였다[17][19]. 계산적사고(computational thinking)는 컴퓨터계산(computation)이 과학의 각 분야에서 도움을 주는 과정에서 생겨난 것으로 이해할 수 있다.

Wing은 계산적사고를 다음과 같이 정의하면서 컴퓨터과학의 중심 사고로 주장하고 있다[25]. “계산적사고는 문제와 해결책을 형식화해서 해결책이 정보처리에이전트에 의해 효과적으로 실행될 수 있는 형식으로 표현되어지는 것을 포함하는 생각의 과정이다[26].”

여기에서 해결책은 인간이나 기계 혹은 인간과 기계의 조합으로 수행된다고 언급하고, 정보처리 에이전트는 기계인 컴퓨터를 넘어 컴퓨터를 조작하는 인간을 포함하는 시스템의 의미로 이해하여야 한다.

계산적사고는 컴퓨터과학의 개념에 기초하여 문제를 해결하는 것, 시스템을 설계하는 것, 인간행동을 이해하는 것을 포함한다[25][31]. 이것은 컴퓨터과학이 초기에 수학, 과학, 공학의 영향을 받았다는 것을 생각해 보면 쉽게 이해할 수 있다. 계산적사고는 수학적 사고, 공학적 사고, 과학적 사고와 상당한 관련이 있다. 문제를 해결하기 위한 일반적인 절차들에 있어 수학적 사고와 유사한 모습을 보이고, 크고 복잡한 시스템을 설계하고 구현하는데 있어서는 공학적 사고와 밀접한 관련을 맺고 있으며, 컴퓨터계산가능성computability, 지능, 마음과 인간 행동을 이해하는 접근은 과학적 사고와 밀접한 관련을 맺고 있다[18][20].

또한 계산적사고는 문제해결에 있어 답을 원하는 것보다는 답을 해결하는 방법을 찾기를 원한다. 방법을 찾으면 답은 구해지고, 방법은 다른 유사한 문제에 재사용가능하다. 계산적사고는 컴퓨터를 이용한 문제해결에 기능, 습관, 접근을 통합하는 것과, 정보와 과제를 체계적, 효과적으로 처리하는 것을 포함하는 사고방식, 컴퓨터를 이용하여 문제를 정의하고, 이해하고, 해결하는 것, 추상화의 여러 수준에서 추론하는 것, 자동화를 이해하고 적용하는 것, 규모의 차원을 이해하는 것 등을 포함한다.

Denning은 계산적사고가 과학을 하는 사고의 한 부류라는 측면에서 과학적 사고와 맥락을 같이한다고 주장한다. 그러나 Denning은 컴퓨터과학 교육에 있어 계산적사고 이외 다양한 것들이 포함되어야 한다고 강조한다. 컴퓨터과학을 대표하는 사고로는 부족하다는 주장이다[19][20].

일상에서 다양한 컴퓨터과학의 원리들이 적용되고 있다[11]. 은행, 극장 등에서 도착한 순서대로 일을 처리하는 모습에서 큐의 원리를 알고 있고, 접시를 쌓아놓고 사용하는 데에서 스택의 원리를 알 수 있다. 목수가 공구함을 허리에 차고 일을 하는 모습에서 캐시의 원리를 볼 수 있다. 그 외에도 백트래킹, 인터럽트와 우선순위, 정렬과 검색, 파이프라이닝의 원리가 적용되는 모습들을 생활 곳곳에서 발견할 수 있다.

계산적사고의 교육측면에서 본다면 계산적사고를 이용한 접근은 생활 곳곳에서 펼쳐있다. 생활에서 이루어지는 일들은 계산적사고를 교육하기 위한 학습 자료로 활용할 수 있다.

2.4 언플러그드 컴퓨팅

언플러그드 컴퓨팅 학습은 뉴질랜드의 팀 벨 교수와 그의 동료들에 의해 개발된 프로젝트를 포함하는 학습 방법으로 컴퓨터과학의 원리를 컴퓨터 없이 학습하는 방법이다[2][3][4][6][9][14][15][16]. 컴퓨터과학의 내용을 정보의 표현, 알고리즘, 절차표현 및 작동 원리의 영역으로 나누어 학습할 수 있는 학습활동을 제안하고 있다. 컴퓨터의 기본원리를 컴퓨터를 이용하지 않고 이해시키고, 느끼게 하고, 새로운 방법을 찾아내도록 하는 것에서 컴퓨터과학자들이 경험한 것을 학생들이 경험할 수 있도록 하는 것에서 큰 의미를 가질 수 있다. 국내에서도 언플러그드 컴퓨팅 방법을 이용한 다수의 연구들이 있다[2][3][4][5][9][14][15]. 이 연구들은 학생들이 놀이를 통해 컴퓨터과학의 원리들을 학습할 수 있다는 것을 알려 준다.

언플러그드의 장점으로는 짧은 시간에 문제를 파악하고 문제해결을 위한 핵심에 접근할 수 있다는 것이다. 이에 비해 프로그래밍 학습도 컴퓨터과학의 깊이 있는 면을 학습할 수 있으나 준비 기간이 많이 투입되어야 한다는 것이 가장 큰 차이점이다. 교육용으로 개발된 LOGO, Squeak eToys, Dolittle, 스크래치 같은 언어들[27][28][29][30]은 언어의 명령어 사용법 및 프로그래밍 구조의 학습시간을 줄일 수 있는 방안들을 제시하고 있으나 언플러그드 방식에 비해서는 많은 시간의 투입은 불가피하다.

언플러그드 컴퓨팅 방법은 노작, 활동, 게임 놀이 그리기 등의 전통적인 수업방식을 사용함으로써 학생들의 신체를 많이 사용하는 학습을 지원한다. 이는 시스템이 복잡하여 추상적으로 밖에 설명할 수 없는 컴퓨터의 작동원리에 대한 이해에 많은 도움을 줄 수 있다.

3. 계산적사고 교육과 언플러그드 컴퓨팅 방법

3.1 계산적사고 교육의 중요성

Dijkstra의 “의학은 칼 과학이 아니듯 컴퓨터과학은 더 이상 컴퓨터에 대한 것이 아니다[21]”라는 말은 정보교육이 나아가는 방향이 컴퓨터를 조작하는 방법교

육이 아니라 사고를 길러 주는 것임을 말해준다.

계산적사고 교육이 초·중등 교육에서 강조되어야 하는 이유를 살펴볼 필요가 있다. 첫째, 계산적사고는 전이가 가능하다[11]. 전이는 교육의 중요한 가치 중 하나이다. 계산적사고는 문제를 해결하는 과정에서 길러지는 것으로 향후 접하게 될 문제에 적용될 수 있다.

둘째, 계산적사고는 과학적(S), 공학적(E), 수학적(M) 사고를 통합적으로 접근할 수 있다. 수학, 공학, 과학은 컴퓨터과학을 통해 하나의 공통된 표현방법인 프로그래밍으로 의사소통을 할 수 있다[21]. 주어진 문제를 해결하기 위한 프로그래밍 과정에서는 이론, 설계, 모델링 등의 다양한 방법들이 동시에 고려될 수 있다. 또한 컴퓨터과학 내용은 최근 과학에 대한 통합적 접근방법인 STEAM교육의 사례로 사용되기에 매우 적합한 주제를 많이 포함한다.

셋째, 계산적사고는 현실에서 다양하게 적용할 수 있는 사고이다. 스택, 큐, 해싱, 탐색과 정렬, 인터럽트 우선순위, 백트래킹, 최소비용 알고리즘들은 컴퓨터과학이 발전하는 과정에서 컴퓨터과학자들에 의해 컴퓨터과학에 추가된 내용들이다. 이들의 개념은 현실의 문제를 해결하기 위하여 컴퓨터과학에 포함된 내용이고, 이들은 다시 현실의 문제에 적용될 수 있다[26]. 현실의 문제는 추상화, 모델링, 자동화, 효율성 등의 계산적사고로 접근하면 문제를 쉽게 해결할 수 있다.

넷째, 계산적사고 교육은 미래 기술에 대해 접근하는 방식을 제공할 수 있다. 초·중등교육은 보편적인 교육을 지향한다. 계산적사고 교육에 사용되는 교육 내용은 컴퓨터과학에 근간을 두게 된다. 컴퓨터과학의 핵심원리를 아는 것은 정보통신기기를 포함한 미래 기술에 대한 발전과 변화에 적용할 수 있는 능력을 길러주는 것이다.

급변하는 정보사회에서 정보통신기기의 단순 사용법에 치우친 교육은 큰 의미가 없다. 미래의 기술을 접하더라도 원리와 원칙에서 접근한다면 쉽게 기술을 이해하고 활용할 수 있다.

다섯째, 계산적사고는 컴퓨터과학의 발전에 도움을 준다. 계산적사고는 컴퓨터과학의 역사적 경험에서 생겨난 것이라고 할 수 있다. 컴퓨터과학이 발전하면

서 새로운 개념들이 추가되었고, 미래에도 추가될 것이다. (그림 1)에서처럼 컴퓨터과학은 계산적사고를 생성하였고, 계산적사고는 컴퓨터과학의 발전에 도움을 준다. 정보교육의 핵심내용은 컴퓨터과학에 근거하므로 정보교육에서 계산적사고 교육은 피할 수 없다. 컴퓨터과학은 사회과학, 생명과학, 물리과학에 깊이 있게 관여하고 있다[24]. 계산적사고는 컴퓨터과학의 핵심적 사고로 다른 과학 분야에도 많은 영향을 줄 수 있다.

3.2 계산적사고를 위한 언플러그드 컴퓨팅 방법

Wing은 계산적사고의 중요성을 주장하면서 계산적사고는 추상화, 자동화, 재귀적사용, 분할정복, 병렬처리를 포함한다고 말한다[25]. 이러한 계산적사고는 프로그래밍을 통해서 길러질 수 있다. 프로그램을 통한 방법은 언어를 익히는데 많은 시간이 걸리고, 구체적인 수준에서 고려해야 할 사항이 많다는 문제점을 가진다.

초등학교 수준에서 교육이 이루어지기 위해서는 다양한 단계의 추상화 수준에서 학습되어야 한다. 이러한 방법의 하나로 Wing은 컴퓨터과학 언플러그드 교육을 언급한다[26].

Wing이 주장하는 계산적사고의 특징을 살펴보면 다음과 같다[25]. 첫째, 프로그래밍보다는 프로그래밍에 이르는 사고의 과정을 중요시한다. 둘째, 단순 암기기능이 아닌 근본적인 기능을 추구한다. 셋째, 컴퓨터가 생각하는 방식이 아닌 사람이 생각하는 방식이어야 한다. 넷째, 수학적 사고와 공학적 사고의 보완과 조합이다. 다섯째, 인공의 산출물 보다는 아이디어에 초점을 맞춘다. 여섯째, 모든 사람과 모든 곳에 산재하는 특성을 보인다.

접시저울 정렬알고리즘의 예를 들어 설명하면 아래와 같다[10][12]. 학생들은 주어진 학습자료(접시저울과 무게추)를 이용하여 정렬알고리즘을 개발하는 활동이다. 수업 결과 학생들은 이름은 다르지만 선택정렬, 버블정렬, 퀵정렬, 합병정렬, 삽입정렬, 개선된 삽입정렬 등을 개발하였다.



(그림 2) 접시저울을 이용한 정렬알고리즘 학습

학생들은 논의를 통해 각자 개발한 알고리즘에 이름을 붙였다. 자신이 개발된 방법이 컴퓨터과학자에 의해 이미 개발되었다는 사실에 학생들은 아쉬움을 표시하기도 하였다.

알고리즘을 문장으로 표현하는데 있어 재귀적인 호출이 이루어지는 것을 알 수 있었으며, 퀵정렬 및 이중 토너먼트 방식에서 분할정복의 개념과 병렬처리의 개념들에 대해 쉽게 이해하였다. 학생들은 개발한 알고리즘의 발표에 있어 자동화 및 재귀호출에 개념을 사용하였다.

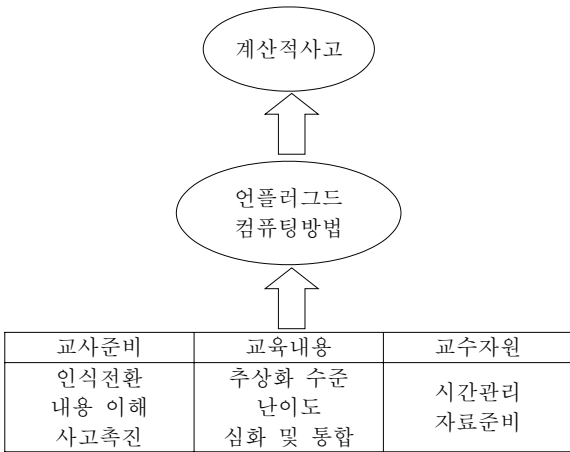
데이터의 저장위치(무게추를 놓는 위치)에 대해 고민하지 않음으로서 학생들은 알고리즘의 핵심부분을 개발하였다. 이는 Wing[25]이 주장하는 프로그래밍에 이르는 사고의 과정을 중요시해야 한다는 것과 아이디어에 초점을 맞추어야 한다는 것에 일치하는 방법으로 사용될 수 있음을 의미한다. 알고리즘을 평가할 때 저울 사용회수를 기준으로 함으로서 효율성을 평가하는 것도 가능하였다.

언플러그드 컴퓨팅 학습방법은 초등학생들의 수준에 맞는 구체물들을 이용함으로써 학생들이 자동화, 분할정복, 효율성, 재귀적 호출 추상화에 대해 쉽게 이해할 수 있도록 도와준다.

4. 언플러그드 컴퓨팅 적용시 고려할 점

학생들의 컴퓨터과학개념 이해와 계산적사고 개발을 위해 교사는 언플러그드 컴퓨팅 방법을 사용할 수 있다. 연구자는 초등학생을 대상으로 시범 적용 [2][3][4][10][12]한 결과와 예비교사를 대상으로 한 연구 [13] 및 참고문헌들 [1][15]을 바탕으로 언플러그드

컴퓨팅 교육방법을 현장으로 확산하는데 있어 고려해야 할 점들을 (그림 3)과 같이 교사준비, 교육내용, 교수자원의 측면으로 살펴보고자 한다.



(그림 3) 언플러그드 컴퓨팅 적용할 때 고려해야할 사항들

4.1 교사준비 측면

언플러그드 컴퓨팅 방법을 적용하는데 있어 교사준비 측면에서 고려해야할 사항을 교사의 인식전환, 내용의 이해, 학생의 사고 촉진의 측면에서 살펴보면 다음과 같다.

첫째, 교사의 인식의 전환이 필요하다. 컴퓨터교육을 위해서는 컴퓨터 실습실에서 수업해야 한다는 생각을 떨쳐버려야 한다. 컴퓨터과학에 핵심적인 내용을 통해 학생들의 사고능력을 길러 주는 것이 더 중요한 사안이라는 인식을 해야 한다. 초등 수준의 학생들을 위해 활동 중심의 수업, 실험 중심의 수업을 진행하기 위한 준비가 되어야 한다.

둘째, 교사는 컴퓨터과학에 대한 깊이 있는 이해를 가지고 있어야 한다. 학생들이 수업도중에 개발하게 될 방법들에 대해 이해 및 평가 할 수 있는 수준의 지식이 요구된다. 학생들이 자주 빠지는 오류의 종류에 대해 이해하고 수정할 수 있는 방법도 알고 있어야 한다[23].

셋째, 교사는 학생들의 사고를 촉진하기 위한 발문 및 활동 계획을 준비할 수 있어야 한다. 주어진 자료를 이용하여 주어진 문제를 해결하는 과정에서 학생들이

사고 할 기회를 얻는 것은 중요하다. 이를 위해서는 교사가 학습자들에게 사고를 촉진할 수 있는 다양한 실마리를 제공해야할 필요가 있다. 교사는 각 단계에서 학습자가 겪는 어려움에 대해 알고 있어야 한다. 단순히 컴퓨터과학을 이해하는 것보다는 자신이 직접 문제를 해결하기 위해 노력한 과정을 중요시 여겨야 한다.

4.2 컴퓨터과학 내용 측면

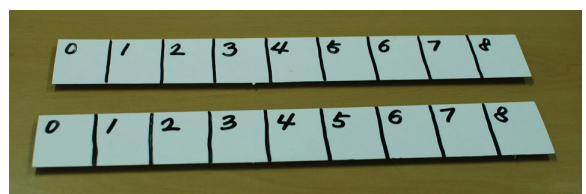
컴퓨터과학 교육내용측면에서 추상화 및 구체화 수준의 정도, 학습내용의 난이도, 심화내용 및 내용통합의 측면에서 살펴보면 다음과 같다.

첫째, 학습자를 고려한 추상화 수준을 설정하는 것이다. 초등학생들에 적용할 내용에 대한 추상화 수준의 교육학적 합의가 있어야 한다.

예를 들어, 접시저울을 이용한 정렬알고리즘 개발 활동에서 저장 공간에 대한 개념을 제외한 정렬방법을 개발[6][10][12]하게 해야 한다.

물리교과에서 이론과 실재를 구분하여 ‘계산의 편의를 위하여 마찰력은 없는 것으로 간주한다’와 같은 개념들이 적용될 필요가 있다. 사고중심의 학습을 위해서는 ‘개발된 알고리즘의 평가의 편의를 위해 메모리 접근에 대한 시간을 생략한다’와 같이 제한할 필요가 있다.

정렬알고리즘을 프로그래밍으로 구체화하는 경우 학생들에게 기억장소관리에 대한 알고리즘 이해가 필요해진다. 접시저울을 이용한 정렬의 경우에는 무게 추를 책상 위 아무 곳에 올려 둘 수 있다[10][12]. 그러나 실제 프로그래밍의 기억공간관리 측면을 고려한다면 교사는 기억공간을 의미하는 종이 띠(그림 4)를 이용하여 기억공간을 만든 후 “종이 띠 위에 한 칸 당 한 개의 무게 추를 놓아야 한다”라는 조건을 추가할 수 있다.



(그림 4) 정렬학습을 위한 메모리 구조

둘째, 컴퓨터과학 내용의 다양한 난이도 수준이 준비되어야 한다. 내용의 깊이는 위계와 깊은 관계를 가진다. 아동의 발달과정 및 학년에 따른 컴퓨터과학의 위계 및 난이도의 수준에 대해서는 더 많은 연구 결과들이 제시되어야 한다. 난이도는 컴퓨터과학 내용의 개념을 학생들이 어렵게 느끼는 정도이다. 이는 사전 지식의 양, 발달단계, 추상화 및 구체화 정도에 따라 학생들이 느끼는 어려움의 정도이다.

예를 들면, 알고리즘의 경우 단순한 알고리즘에서 복잡한 알고리즘의 순서로 준비되어야 하고, 반복 및 비교 분기의 측면에서는 단순 반복, 단순 분기에서 복잡한 반복과 복잡한 분기로의 수준으로 다양한 수준의 난이도가 준비되어야 한다. 또한 문제의 복잡도 및 문제해결 조건사항의 종류도 함께 고려되어야 한다.

셋째, 컴퓨터과학 내용의 심화 및 통합으로 이끄는 측면이다. 개발된 알고리즘, 해결책, 방법 등은 개선의 가능성이 많다. 이를 위해서는 문제의 확장에 다른 범용해결책으로의 확장 및 개발된 알고리즘의 평가를 통한 선택 등의 심화활동이 이루어 질 수 있다.

알고리즘의 전달 및 이해시키기 학습[2][3][4][9]은 심화학습에 대한 고려사항이 많지 않을 수 있으나 창의성을 요구하는 개발형태의 수업[10][12]은 더 많은 교사의 준비가 필요하다.

예를 들어 정렬알고리즘의 개발이후 평가의 측면에 있어서 볼 때, 개발된 알고리즘의 효율성에 대한 평가가 이루어져야 한다[10][12]. 효율성은 추상화와 더불어 컴퓨터과학에 있어 중요한 개념이다. 알고리즘 개발과 평가라는 개념이 통합될 수 있다. 비트맵 이미지표현 학습[4]의 경우 이미지 표현과 숫자로 상대방에 전달하는 것을 통해 통신프로토콜 개발과 통합될 수 있다.

또한 킷 정렬에 있어 접시저울을 이용할 경우 분할정복에 대한 개념 및 피봇에 대해서는 학생들이 쉽게 이해하지만, 실제의 프로그래밍에 있어서는 메모리 관리에 대한 개념이 추가로 필요하다. 부가적인 알고리즘은 킷정렬 프로그래밍 학습을 어렵게 하는 요소로 작용한다. 구체적 알고리즘에 대한 아동의 이해 수준을 고려하여 심화활동으로 이끌 수 있다.

4.3 교수자원의 측면

교육을 위한 교수자원 준비 측면에서는 주어진 시간을 관리하는 측면, 자료준비 측면에서 살펴보면 아래와 같다.

첫째, 주어진 시간의 관리 측면이다. 아동이 학습 활동을 하고, 새로운 방법들을 개발하는 데에는 더 많은 수업시간 필요하다. 언플러그드 방식의 목적은 학습내용을 전달하는 것 보다는 학생들이 새로운 방법을 개발하는 경험을 하는 것에 있다. 새로운 방법을 개발하는 과정에서 깊이 있는 사고의 기회를 가지기 때문이다. 사고를 촉진하기 위해서는 최소한의 시간이 확보되어야 하고, 팀으로 역할활동과 구성원 토론 및 학급 발표가 필수적이다.

수업 전개는 문제과약, 시험 해결책개발, 정교화, 해결책 발표, 질문 및 토론의 과정으로 진행된다[10][12]. 한 교실에 20-30명의 학생을 대상으로 접시저울 정렬알고리즘 학습을 실시한 결과 2차시(80분)를 넘기는 경우가 간혹 있었다[10][12].

오렌지게임[6][16]의 경우 게임에 참가하는 사람이 4명일 경우는 참가자들은 쉽게 문제를 해결하지만, 8명 정도 되면 오랜 시간이 걸린다. 다양한 모양의 네트워크 구성을 경험하기 위해서는 7명 이상의 많은 인원이 필요하다.

둘째, 수업 자료 준비 측면이다. 교사는 컴퓨터과학 수업을 위해 개발된 교과 내용에 대해 이하고 있어야 하고, 수업에 필요한 자료를 준비할 수 있어야 한다. 초등학생들이 쉽게 접할 수 있는 종이접, 접시저울, 무게 추, 학습지, 실험보고서, 활동지, 플래시카드 등의 수업자료들이 학생들의 생각을 촉진시키기 때문에 많은 자료의 준비는 필수적이다. 또한 동일한 자료라도 협동적으로 활동을 하기 위해서는 대형의 자료들이 필요하였다. 정렬망[6]을 통한 수업의 경우 바닥에 정렬망을 그리는 것은 많은 시간을 요구한다. 정렬망을 천에 인쇄하여 펼쳐 사용할 경우 시간을 줄일 수 있다[12]. 이진카드의 경우 개인용 및 집단용의 크기는 달라질 수 있다. 수업 자료가 대량으로 사용된다면 교구제작사를 통한 상업용 교구 구입도 가능할 것이다.

5. 결론 및 제언

컴퓨터과학의 발전에 따라 정보교육의 모습은 조금씩 달라졌다. 전산교육, 컴퓨터교육, ICT활용 교육, 정보교육으로 명칭이 변화되면서 각 교육에서 강조점들도 변화되었다. 최근에는 컴퓨터과학의 내용에 기반한 계산적사고 교육이 관심을 받고 있다. 계산적사고가 특히 초등교육에서 중요한 이유는 전이가 가능하고, 통합적 적용이 가능하고, 현실에서 다양하게 사용될 수 있고, 미래 기술을 접근하는 방식을 제공한다는 측면에서 중요하다.

본 논문에서는 컴퓨터과학의 계산적사고의 특성을 알아보고, 계산적사고를 교육하는 방법으로서의 언플러그드 컴퓨팅 방법의 적합성에 대해 살펴보았다. 언플러그드 컴퓨팅 연구는 시범적으로 연구되었고, 컴퓨터과학 전문가를 포함하여 학습내용이 개발되고, 초등교사에 의해 적용되었다. 그러나 확산을 위해서는 컴퓨터과학 전문가의 도움 없이 초등교사에 의해서 이루어 질 수 있어야 한다.

본 연구에서는 언플러그드 컴퓨팅을 적용할 때 고려해야할 점들을 교사준비, 교육내용, 교수자원 측면에서 살펴보았다. 교사준비 측면에 있어 인식의 전환, 내용에 대한 이해, 사고촉진 방법들의 준비가 필요하였고, 교육내용의 측면에서는 추상화 수준의 설정, 내용의 난이도 조정, 내용 심화 및 내용 통합이 필요하였다. 교수자원의 측면에서는 수업에 할당된 시간의 관리와 자료준비가 필요함을 살펴보았다.

계산적사고를 길러주기 위해 언플러그드 컴퓨팅 방법을 적용하는 것은 바람직하지만 더 다양한 측면에서 문제점 및 고려사항들에 대한 연구가 필요하다. 향후 교사양성 및 현직교사 연수에 대한 연구도 필요할 것이다.

참 고 문 헌

[1] 권은정(2008). 놀이를 통한 알고리즘 개념 학습이 학습 동기 및 학업 성취도에 미치는 영향. 한국교원대학교 교육대학원 석사학위논문. 한국교원대학교.
[2] 박연, 김지나, 한병래(2007). 초등학교 컴퓨터교육에서 라우팅알고리즘 학습가능성에 관한 연구. 한

국정보교육학회논문지 11(3), pp. 267-279.
[3] 박윤성, 한병래(2009). 초등학교 컴퓨터교육에서 언플러그드 학습 방법을 활용한 정보표현 영역 교수-학습에 관한 연구. 한국정보교육학회 논문지 13(4), pp. 479-487.
[4] 이미영, 구정모, 한병래(2008). 초등학생의 특성을 고려한 비트맵이미지 저장원리 수업을 통한 초등 정보과학의 교수학습에 관한 연구. 한국정보교육학회논문지 12(4), pp. 405-415.
[5] 이용배, 이영미(2009). 놀이 활동 중심과 애니메이션 기반의 정렬 알고리즘 교수-학습 방법 비교. 한국정보교육학회논문지 13(2), pp. 225-236.
[6] 이원규 외 역(2006). 놀이로 배우는 컴퓨터 과학, 원저 Tim Bell, et al(1998) Computer Science Unplugged. 흥릉과학출판사.
[7] 이태욱(1999). 컴퓨터교육론. 좋은 소프트.
[8] 이태욱(2001). ICT 교육. 형설출판사.
[9] 임민영, 정상목, 한병래(2006). 초등학교 컴퓨터교육에서 검색과 정렬 알고리즘 학습가능성에 관한 연구. 한국정보교육학회논문지 10(3), pp. 289-298.
[10] 한병래(2010). 놀이를 통한 검색과 정렬 방법 찾기. 진주교육대학교 초등교육연구원 2010 창의성 통합캠프 자료집, pp. 55-68.
[11] 한병래a(2012). 정보교육과 계산적사고. 한국정보교육학회 학술논문집 3(1), pp. 57-62.
[12] 한병래b(2012). 창의적 방법으로 자료 찾기와 정렬하기. 진주교육대학교 초등교육연구원 2012 창의성 통합캠프 자료집, pp. 73-92.
[13] 한병래c(2012). 초등 예비교사 교육을 위한 활동 중심 정보과학교육. 한국정보교육학회 학술논문집 3(3), pp. 47-51.
[14] 한선관, 김정신(2007). 초등학생을 위한 컴퓨터과학의 언플러그드 학습 방법 연구. 한국정보교육학회논문지 11(4), pp. 497-504.
[15] 한희섭, 한선관(2008). 언플러그드 컴퓨팅을 이용한 예비교사의 정보교육 사례 연구. 한국정보교육학회논문지 13(1), pp. 23-30.
[16] Bell et. al.(2009). Computer Science Unplugged : School Students Doing Real Computing Without Computers. *New Zealand J of Applied Computing*

and Information Technology, 13(1), pp. 20-29.

[17] Denning(2001). *The invisible future*. NY : McGraw-Hill.

[18] Denning(2008). The computing field:structure. <http://cs.gmu.edu/cne/pjd/pubs/enc/cs08.pdf>

[19] Denning(2009). Beyond Computational thinking. *communications of the ACM*, vol 50(7), pp. 13-18.

[20] Denning(2010). *What is computation, Ubiquity vol 2010*. ACM.

[21] Dijkstra(1986) On a cultural gap. *The math intell. vol 8(1)*, pp. 48-52.

[22] Hromkovic(2006). Contributing to general education by teaching informatics. ISSEP 2006, *LNCS 4226*, pp. 25-37.

[23] O. Hazzan, T. Lapidot, N. Ragonis(2011). *Guide to teaching computer science : An activity-based approach*. Springer-Verlag London.

[24] Rosenbloom, P. S(2004). A new framework for computer science and engineering. *IEEE computer*(Nov, 2004), pp. 31-36.

[25] Wing, J. M.(2006). Computational Thinking. *CACM*, vol 49(3), pp. 33-35.

[26] Wing, J. M.(2010). Computational Thinking What and Why. *Phil. Trans. R. Soc. A*, vol 366, pp. 3717-3725.

[27] LOGO 프로그래밍 언어 LOGO Foundation. <http://el.media.mit.edu/logo-foundation/logo/programming.html>

[28] Squeak e-Toy. <http://www.squeak.org/>

[29] 두리틀Dolittle. <http://dolittle.eplang.jp>

[30] 스크래치scratch. <http://scratch.mit.edu>

[31] Google의 Computational thinking 페이지. <http://www.google.com/edu/computational-thinking/what-is-ct.html>

저 자 소 개

한 병 래



1992 대구교육대학교(학사)
 1998 한국교원대학교 컴퓨터교육과 (교육학석사)
 2002 한국교원대학교 컴퓨터교육과 (교육학박사)
 2003 세종대학교 초빙교수
 2004- 진주교육대학교 교수
 관심분야 : 컴퓨터교육, 정보교육, 컴퓨터교육과정 및 방법, e-learning, 정보과학교육, 스마트교육
 e-mail : raehan@cue.ac.kr