

공개 소프트웨어를 활용한 문제해결력 향상 교수학습 모형개발

김길용 · 노영욱

신라대학교 컴퓨터교육과

A Development of Teaching and Learning Model for Enhancing Problem Solving Ability Based on Open Source Software

GilYong Kim · YoungUhg Lho

Department of Computer Education, Silla University

요 약

7차 교육과정은 응용소프트웨어를 얼마나 잘 다루는지와 같은 컴퓨터를 도구적 활용을 중점으로 구성되었다. 컴퓨터 과학의 기본원리를 적용하여 문제해결능력을 신장시키고 이를 구현하기 위한 체계적인 교육의 필요성이 제기되어 2007년 개정 교육과정에서 문제해결방법과 절차라는 대영역이 포함되었다. 정보과목 문제해결방법과 절차 영역에서 다양한 문제를 이해 분석하여 알고리즘을 설계하고 구현하는데 있어 원활한 교수학습을 위한 다양한 방법과 도구들에 대한 연구가 요구되고 있다. 본 연구는 창의적 문제해결 향상에 효과적인 프로그래밍 언어들은 상용소프트웨어들의 비용부담과 언어적 문법, 에러발생에 대한 해결에 비중이 높아 학습자가 겪는 인지적 부담을 감소 할 수 있는 EPI(Educational Programming Language)인 Scratch 프로그래밍으로 창의적 문제해결력 향상을 위한 교수학습 모형을 개발하였다.

키워드

공개소프트웨어, 문제해결력향상, 교수학습 모형, EPL

I. 서 론

7차 교육과정에서는 중학교에서는 『컴퓨터』교과로 일상생활의 문제 해결력에 정보통신기술을 활용한 교육목표를 두어 ICT활용교육에 워드프로세서, 스프레드시트, 파워포인트, 인터넷 등 응용소프트웨어 활용 및 사용법 위주의 수업이 진행되었다. 이러한 교육과정이 컴퓨터과학의 학문적 중심으로 창의적 문제해결력을 위한 교육이 소외되고 응용 소프트웨어의 단순한 기능을 익히고 사용하는 위주의 내용으로 컴퓨터를 도구로써 활용되었다는 점은 분명한 사실이다. 또한 이러한 소프트웨어는 상용소프트웨어로써 소프트웨어 구입과 업그레이드(upgrade)로 인한 막대한 비용이 들었다. 7차 교육과정의 『컴퓨터』교과를 『정보』교과로 개정하여 개정 제7차 교육과정 발표하여 이러한 문제점을 개선되었다고 할 수 있다.

정보교육과정의 목표를 살펴보면 정보처리의 기본 원리와 올바른 정보 활용 지식을 습득하여 자신의 생각을 다양한 형태의 정보로 표현하고 실생활에서 일어나는 문제를 창의적이고 능동적인 방법으로 해결할 수 있는 능력과 태도를 기른

다. 따라서 문제해결방법 및 절차영역을 추가하여 일상생활에서 다양한 문제를 정보처리의 관점에서 이해하고 정보처리의 지식과 기능을 활용하여 창의적이고 능동적으로 문제를 해결할 수 있다 [1]. 실생활에서 일어나는 다양한 문제를 컴퓨터 과학의 학문적 중심으로 이해 분석하여 효율적인 문제해결 방법을 찾고, 알고리즘을 설계하여 결과를 구현해 보는 내용으로 중점을 두고 있다. 본 연구는 창의적 문제해결 향상에 효과적인 프로그래밍 언어들은 상용소프트웨어들의 비용부담과 언어적 문법, 에러발생에 대한 해결에 비중이 높아 학습자가 겪는 인지적 부담을 감소 할 수 있는 EPI(Educational Programming Language)인 Scratch 프로그래밍으로 창의적 문제해결력 향상을 위한 교수학습 모형을 개발하였다.

II. 관련 연구

1. 프로그래밍 교육의 특징

프로그래밍 교육은 창조적이고, 논리적 사고를 동반한 다양한 문제해결의 활동으로 창의적 문제

해결력을 기를 수 있는 유용한 도구이다. 문제 해결 과정 중 순서도작성을 통해 다양한 논리적 사고력을 키울 수 있으며 오류수정 등 문제해결 과정을 통해 반성적 사고와 문제해결 능력이 자연스럽게 키울 수 있다. 또한 문제해결 과정에서 과학 수학 등 다양한 문제를 해결함으로써 범교과적인 학습이 이루어 질수 있다.

프로그래밍 과정에서 요구되는 문제 해결과 지속적인 오류 검증 및 수정 작업에서 요구되는 반성적 사고를 통해 고등인지 기술 향상을 강조한 것으로 프로그래밍 과정은 문제해결능력을 요구하므로 이 학습은 인간의 사고력을 향상시킬 수 있는 잠재적 특성을 지니고 있는 전제이다[2].

프로그래밍은 컴퓨터를 사용하는 학생들에게 인지적 사고력과 문제해결력을 향상시켜 줄 수 있는 효과적인 학습을 제공해 준다. 또한 프로그래밍은 자신의 생각을 구체적 행동으로 연결하여 쉽게 그 결과를 시각화할 수 있고, 프로그램이 활동에서 발생하는 오류를 학생 자신이 찾아내고 수정함으로써 문제해결력을 기르며 논리적 사고력과 창의력을 개발할 수 있다[3]. 이러한 교육의 학습자는 논리적 사고력증진, 문제해결력 향상, 문제해결의 결과에 대한 성취감 획득, 토론을 통한 상호작용, 협동심으로 높은 인지능력을 향상시킬 수 있다. 그러나 프로그래밍언어는 중등학생이 학습하기에 학습자에 따라 프로그래밍언어에 대한 문법과 구조를 익히는데 많은 어려움에 인지적 부담을 가지게 된다. 일반적으로 알려진 텍스트기반의 프로그래밍 언어를 사용하여 프로그래밍교육을 하고 있다. 텍스트 기반의 프로그래밍 언어는 프로그램 자체의 문법과 구조를 이해해야 하고 프로그램을 작성하는 방법을 익히는데 많은 시간과 노력을 들이고 나서야 본인이 원하는 것을 만들 수 있기 때문에 초보자에게 프로그래밍어 어려움이 있다.

개정된 제 7차 교육과정에서 교육용 프로그래밍언어를 신장하고 있다. 또한 상용 프로그래밍언어의 비용적 차원에서 고려하여 공개용 프로그래밍 언어를 고려 할 필요가 있다. 교육용 프로그래밍 언어의 시각적 환경을 제공하면서 Drag & Drop 형태로 쉽고 간단하게 프로그램을 구현할 수 있는 대표적인 프로그래밍언어는 Agentsheets, Squeak Etoys, Alice, Scratch 등이 있다[4]. 이러한 교육용 프로그래밍언어는 초보학습자가 다루기 쉬운 형태로 구성되어 있으므로 프로그래밍 학습과정에서 부가되는 학습자들의 인지적 부담을 감소시켜 주고 프로그래밍 학습에 흥미를 유발하고 몰입하게 한다. 또한 풍부한 멀티미디어 콘텐츠 제작을 위한 저작기능을 포함하고 있기 때문에 프로그래밍 학습 뿐 아니라 수학 및 과학적 개념 습득, 예술적 감각 습득 등에 이용한 교수 학습 환경을 제공할 수 있다[5].

Piaget는 인간의 인지 발달단계를 4단계로 구분했다.

- 1단계 : 감각운동 단계(0세~2세)

- 2단계 : 전 조작 단계(2세~7세)
- 3단계 : 구체적 조작 단계(7세~11세)
- 4단계 : 형식적 조작 단계(11세 이상)

각 단계가 시작되는 시기는 모든 아동들에게 각 단계를 거쳐서 발달하지만 문화나 개인의 차이에 따라 그 속도는 다르게 나타난다. 즉 형식적 조작기는 11세, 12세가 되어도 형식적 조작기 단계에 있을 수 있고 구체적 조작단계에 있을 수도 있다. 우리나라 중학교 1학년 기준으로 볼 때 13세가 됨으로 형식적 조작단계에 있다고 할 수 있다.

Piaget의 지적 발달 단계 및 행동 특성은 형식적 조작기의 인지적 조작의 특징은 추상적 사고와 명제적 사고, 조합적 사고 가능하다. 다양한 가능성과 해결방안의 모색이 가능하다. 이론적 문제해결의 기본이 되는 가설을 세우고, 검증력이 발달된다. 추상적 명제들에 대해서 논리적으로 생각할 수 있고 가설들을 체계적으로 검증할 수 있다(가설적인 명제를 다루는 능력), 가상적 미래 그리고 이데올로기적 문제들에 관심을 갖게 된다 [6]. 결론적으로 Piaget의 인지구조의 발달로 볼 때 중학생 1학년생이면 형식적 조작기의 특성을 갖추고 있다고 할 수 있다. 형식적 조작기는 논리적 사고, 추상적인 개념을 이해할 수 있으므로 문제해결능력 향상에 프로그래밍 교육이라는 것은 이미 많은 논문에서 검증되고 있으며, 중학교에서 프로그래밍 교육과정이 다양한 문제 해결과정에서 논리적 창의적으로 문제를 해결할 수 있도록 프로그래밍 교육을 적용함이 적당하다고 할 수 있다.

2. Scratch

Scratch는 MIT Media Lab의 Lifelong Kindergarten 연구에서 개발된 교육용 프로그래밍언어로 풍부한 미디어기반 프로그래밍 환경을 제공한다. 학습자들은 이러한 Scratch의 다양한 멀티미디어 지원 기능을 통한 학습 경험은 학습자 스스로 프로그램을 직접 설계하고 생성 및 개발하는 과정을 통해 최적의 학습경험을 얻을 수 있다. 즉 학습자들은 컴퓨터 프로그램을 직접설계하고 개발하는 과정을 통해 최적의 학습경험을 하게 된다[7]. Scratch와 관련 연구는 대학의 프로그래밍 코스에서 프로그래밍과 관련된 사전 지식이 전혀 없는 초보 학습자들이 Scratch를 통해 미리 프로그래밍과 관련된 기본 개념을 익힘으로써 이후 진행된 Java 프로그래밍 학습에 긍정적인 영향을 끼쳤음을 보여주었다. 이는 프로그래밍의 기본개념 습득을 위해 Scratch가 유용한 도구로 활용될 수 있으며, 이러한 프로그래밍에 관한 기본 개념의 습득이 프로그래밍 능력 향상에 중요한 요인으로 작용한다는 것을 의미한다. 따라서 초중등학교에서 이루어지는 프로그래밍 교육은 특정 언어 사용 능력이 아니라, 프로그래밍에 관한 기본 개념을 습득하고 이를 활용할 수 있는

목표로 하여야 한다[5].

Scratch를 활용하여 작성해야 할 프로그래밍 문제는 명확한 목표 상태를 가지고 있으며, 프로그램 작성 직후, 무대에서 다양한 캐릭터들을 통해 즉각적이고 생동감 있는 시연이 가능하기 때문에 학습자에게 즉각적이고 구체적인 피드백 전달이 가능하다.

스크래치는 그림 1과 같이 프로그램 작성을 위한 명령 구문들은 블록 형태로 제공함으로써 블록을 차례대로 쌓는 과정을 통해 원하는 프로그램을 쉽게 구성할 수 있다. 또한 블록들을 쌓을 때 각 블록들은 구문규칙에 적합한 경우에 한해서만 서로 결합하므로, 오류를 줄일 수 있으며, 각 블록들은 실행문, 조건검사, 조건문, 반복문, 변수와 같은 기본적인 프로그래밍 관련 개념들을 포함하고 있다. 각 문법적 기능을 블록단위의 이미지 형태로 Drag & drop으로 쉽고 간단하게 프로그래밍을 구현할 수 있다. 따라서 본 연구는 Scratch 프로그래밍을 활용하고자 한다.

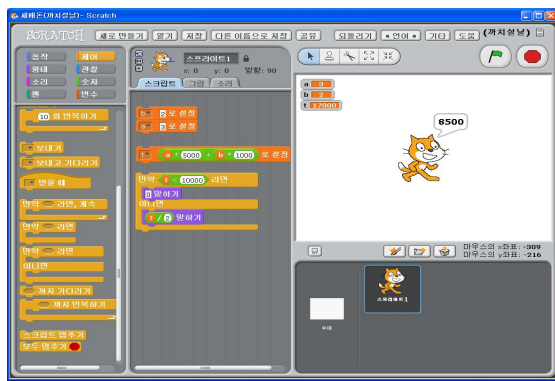


그림 1. Scratch 프로그램 화면

III. 문제해결력 향상과 프로그래밍 교육

1. 문제해결력 기준 연구

문제해결력의 정의가 학자에 따라 차이가 있지만 주어진 문제를 기술하고 해결하고 해결책을 보고하는 등의 일련의 학생들의 인지적 활동이라는 점에서 의견의 일치하고 있다. PISA 2003 검사 문제해결 영역을 규정짓는 가장 중요한 특징 세 가지는 인지적 과정이며, 범교과적이고, 실생활 문제에 기초로 하고 있다. PISA 2003에 새로이 도입된 문제해결 영역은 해결과정이 명료하지 않으면서 적용 가능한 소양 영역이나 교과 영역이 수학, 과학, 및 읽기 중에서 어느 하나의 영역 내에 국한되지 않는 실제적이고 범교과적 문제 사항을 직면했을 때 이를 해결하기 위하여 인지적 과정들을 활용할 수 있는 개인의 능력을 말한다. 문제해결 과정에 대한 기존 연구로는 Wallas의

4단계설, Dewey의 탐구 과정모형, Osborn-Parnes의 창의적 문제해결 모형, polya의 모형, Schoenfeld의 모형, Krulik과 Rudnick의 모형, 한국교육개발원의 모형 등이 있다.

2. 제안된 문제해결력 향상 교수 학습모형

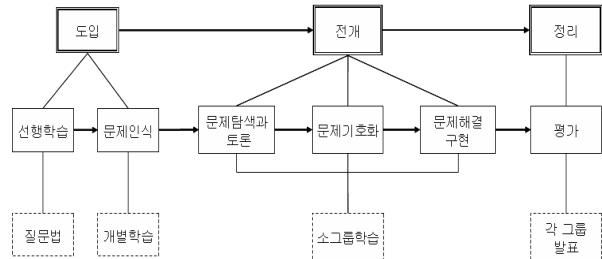


그림 2. 제안된 수업모형

제안된 수업 모형은 그림 2와 같이 크게 3 단계로 구성되며 세부 단계별 활동은 다음과 같다.

- 선행학습 : 전시 학습한 개념과 원리를 상기시키고 중요한 원리를 다시 재 적용하여 새로운 개념과 원리에 적용할 수 있도록 선행학습을 통해 이해정도를 파악하고 문제를 해결할 수 있도록 개념을 인식시킨다.
- 문제인식 : 선행학습으로 파악한 이해정도를 고려하여 흥미롭고 호기심을 가질 수 있는 문제를 제시하여 문제점을 파악하고 개인이 인지할 수 있는 범교과적으로 파악하여 문제를 단순화 조직화 기호화한다. 주어진 문제와 관련된 타 교과와 내용과 관련하여 생각해본다. 문제를 이해하고 주어진 문제에 대한 조건을 파악한다. 문제를 단순화 조직화 기호화하여 문제와 관련시켜 본다.
- 문제탐색과 토론 : 문제를 해결하기 위한 개념, 원리를 정리하는 단계이다. 학습자는 문제에 대한 인식이 다르므로 소집단의 학습자끼리 토론하여 문제해결 계획을 세우기도 하고, 소집단의 학습자가 다양한 생각을 창안해 낼 수 있도록 교수자의 발문형태로 힌트만 제공하여 많은 문제해결방안을 도출하고 문제를 해결하기 위한 새로운 아이디어를 만들어 낸다.
- 문제 기호화 : 문제해결의 가장 중요한 단계로 지금까지의 활동을 통해 범교과적인 차원에서 인지한 개념, 원리를 순서도를 통해 표현하여 문제를 해결할 수 있는 여러 사항을 최선의 방법을 선택하여 순서도화 한다.
- 문제해결 구현 : 작성된 순서도를 스크래치프로그래밍을 선택하여 구현한다. 학습자는 스크래치프로그램 실행방법을 생각해보고 채택한 구현방법으로 구현해 본다. 문제해결에 대한 결과를 바로 도출하여 확인할 수 있으므로 학습

자의 생각이 옳고 그름을 판단할 수 있다. 문제해결에 대한 자료를 입력하여 올바른 결과를 확인함으로써 문제해결에 능동적으로 확신을 가지고 창의적으로 문제를 해결하고자하는 마음, 흥미, 결과에 대한 만족감, 성취감을 느끼게 한다. 점차적으로 사고력을 요하는 문제를 제시 구현함으로써 학습자들이 문제해결에 능동적으로 확신한다.

- 평가 : 문제해결방법 및 절차와 결과를 발표함으로써 학습자가 구현한 문제해결절차 및 방법이 다른 학습자의 문제에 대한 문제해결 과정을 비교하여 더 나은 방법을 모색해본다

제안된 수업모형을 적용할 개정된 7차 교육과정의 정보교과는 단계별 4개의 대영역으로 나누어져있다. 단계별 연간 34시간 각 영역별 수업시수를 나누었다. 정보교과의 컴퓨터과학중심으로 개정된 취지에 맞추어 문제해결 방법과 절차 영역에 중요도를 고려하여 다른 영역보다 비중을 두었다. 교수학습과정은 표 1과 같다.

표 1. 대영역에 대한 수업시간

대영역명	중학교 단계별 수업시수			수업시수
	1단계	2단계	3단계	
정보기기의 구성과 동작	8	8	8	24
정보의 표현과 관리	8	8	8	24
문제 해결 방법과 절차	10	10	10	30
정보사회와 정보기술	8	8	8	24
수업시수의 합계	34	34	34	102

중학교 1단계의 프로그래밍 기초의 영역과, 3단계 자료의 정렬영역의 자료의 탐색영역으로 선택하여 수업시수를 나누어 수업을 진행한다. 각각 소영역의 개념을 인지시키고, 문제중심학습으로 수업을 진행하였다. 문제해결방법과 절차 영역의 1단계의 중영역인 프로그래밍 기초영역의 중요도를 고려 8시간으로 배정하여 수업을 진행할 예정이다.

중학교 3단계의 자료의 정렬과 자료의 탐색영역에는 각각 5시간씩 나누어 진행할 예정이다. 총 프로그래밍 기초 8시간, 자료의 정렬 5시간, 자료의 탐색 5시간으로 23시간에 대해 문제중심학습으로 수업이 진행할 계획이다.

V. 결론

본 연구와 관련하여 향후 연구과제로는 Scratch 프로그램기반으로 창의적 문제해결력 향상과 수업 후 학습흥미를 측정하기 위해 경남 김해시 거

주하는 중학생 40명을 선정하여 실험집단 20명, 통제집단 20명으로 구성하고, 각각 20명은 문제해결력에 대한 사전 및 사후 검사와, 문제를 해결하는 과정의 흥미와 성취감에 대한 검사를 실시할 예정이다. 실험대상의 문제해결력과 프로그래밍능력 알아보기 위해 한국정보올림피아드 2004년~2006년도의 예선 문제지로 하여 실험 통제집단을 사전검사를 하고 문제해결향상을 위한 순서도, 프로그래밍에 대한 사전에 경험해 본적이 있는지 알아볼 예정이다. 개정된 정보교과 문제해결 방법 및 절차영역의 프로그램기초 교육과정을 본 연구대로 진행한 결과 창의적 문제해결력 향상되어 있는지 알아보기 위해 OECD/PISA 2003에서는 문제해결 영역에 대한 평가가 새로이 도입되어 학생들의 범교과적 문제해결능력을 평가할 계획이다. PISA 2003의 문항을 중학교 1학년(13세) 정도의 인지적 능력을 고려하여 문항을 수정하여 문제해결능력을 평가하기 위해 검사지로 선택할 것이다.

참고문헌

- [1] 이옥화, 허희옥, 조미현, 김민정, 김미량, 강신천, 강오한, 이준희, 최재혁, 추병완 『정보교육의 ABC』 교육과학사.
- [2] 박성진, 박관우, 웹 기반 프로그래밍 튜터 시스템, 한국정보교육학회, 제 5권 2호, 2001.
- [3] 고일석, “웹기반 교육용 프로그래밍 언어 JAVA MAL의 설계 및 활용”, 서울대학교 대학원, 석사학위논문, 1999.
- [4] East, P. & Hutton. M(2007). Point/counter-point : On using graphical languages to teach programming. National Educational Computing Conferencd, Georgia World Congress Center, 2007.
- [5] 이은경, 이영준, Scratch 활용 프로그래밍 교육이 중학생의 몰입수준과 프로그래밍 능력에 미치는 영향, 중등교육연구, 56(2), 359-382, 2008.
- [6] 김정휘, 주영숙 공저, 교육심리학 탐구, 형설출판사, 1998.
- [7] Kafal, Y.B. Construction. In sawyer, R.K(eds.). The cambridge handbook of the learnig sciences. New York: Cambridge University Press. 2006.