

백워드 설계 모형을 적용한 소프트웨어 교과의 교수설계에 관한 연구

이영호 · 구덕회

서울교육대학교 컴퓨터교육과

요 약

본 연구의 목적은 백워드 설계 모형을 적용하여 초등학교 소프트웨어 교육 운영 지침의 ‘알고리즘과 프로그래밍’ 단원의 교수학습계획을 개발함으로써, 백워드 설계 모형이 소프트웨어 교육과정 구성에 주는 시사점을 도출하는 것이다. 연구자는 Wiggins와 McTighe의 백워드 설계 템플릿에 기초하여, ‘알고리즘과 프로그래밍’ 단원의 교수설계를 개발하였다. 첫째, 교육부에서 제시한 소프트웨어 교육과정 운영 지침에 제시되어 있는 목표, 내용체계표, 성취기준, 단원목표 등을 확인함으로써, ‘알고리즘과 프로그래밍’ 단원에 대한 영속적 이해, 포괄적이고 본질적인 질문, 핵심 지식과 핵심 기능을 도출하였다. 둘째, GRASPS 기법을 활용하여 ‘알고리즘과 프로그래밍’ 단원의 수행 과제와 총체적인 루브릭을 개발하였다. 셋째, ‘알고리즘과 프로그래밍’ 단원을 효과적으로 수업하기 위하여, WHERETO 요소의 14개 활동으로 구성된 7차시 수업을 개발하였다. 마지막으로, 백워드 설계 모형을 이용한 교육과정 구성이 소프트웨어 교육에 미치는 효용성에 대하여 고찰하였다. 백워드 설계 모형은 소프트웨어 교육을 위한 단원 및 수업 설계 개선에 많은 시사점을 제공할 것이라 기대한다.

키워드 : 이해중심 교육과정, 백워드 설계 모형, 소프트웨어 교육, 교수설계, 알고리즘과 프로그래밍

A Study on the Instructional Design of Software Education Based on Backward Design Model

Youngoho Lee · Dukhoi Koo

Dept. of Computer Education, Seoul National University of Education

ABSTRACT

The purpose of this study is derived implications at software curriculum development utilizing the backward design model. In this study, we developed ‘Algorithms and Programming’ unit teaching plan based on backward design template. First, we have derived enduring understandings, essential questions, specific knowledge and skill on ‘Algorithms and Programming’ unit by considering the goal, content, achievement standard of Software education operating instructions. Second, we developed authentic tasks using GRASPS technic and holistic scoring rubrics. Third, we developed 7 lesson 14 WHERETO element for effective teaching in ‘Algorithms and

교신저자 : 구덕회(서울교육대학교 컴퓨터교육과)

논문투고 : 2015-09-29

논문심사 : 2015-09-30

심사완료 : 2015-12-09

Programming' unit. Fourth, we investigated about the effectiveness of the development unit based on backward design. Backward design could be useful of developing curriculum unit and lesson plan at software education.

Keywords : Understanding by Design, Backward Design Model, Software Education, Instructional Design, Algorithms and Programming

1. 서론

소프트웨어가 혁신과 성장, 가치 창출의 원천이 되며 개인·사회·국가 경쟁력의 기본이 되는 소프트웨어 중심 사회로 사회가 급격히 변화하고 있다. 세계의 주요 선진국은 소프트웨어에 대한 중요성을 인식하고 있으며 미래 세대들의 소프트웨어 역량을 강화하기 위해 소프트웨어 교육을 필수 교과로 지정하여 시행하고 있다. 우리나라도 문이과 통합 교육으로 창의융합인재 양성을 목표로 하는 '2015 개정 교육과정'에 소프트웨어 교육을 포함하였으며 2018년부터 초중등학교에서 소프트웨어 교육을 필수적으로 운영할 예정이다. 이에 교육부에서는 2015 개정 교육과정이 적용되기 전까지의 소프트웨어 교육과정을 운영하는 안내서로서 소프트웨어 교육 운영 지침을 개발하였다[8]. 교육과정을 질적으로 우수한 수업으로 구현하기 위해서는 교과의 교육내용이 학문 기반의 핵심 개념과 원리들로 구성되어하므로 교육과정을 핵심 성취기준 중심으로 개발하고 있다[12]. 하지만 2015 개정 교육과정을 위한 교과 교육과정 개발 방향 설정 연구에 따르면 우리나라 2009 개정 교육과정에서 교과의 성취기준은 학생의 수행으로 진술되기보다 알아야 할 것을 나열하여 제시하여 교과의 핵심 개념을 반영하지 못하고 있으며, 교과를 통해 학생들이 할 수 있어야 할 것을 제대로 제시해 주지 못하고 있다고 분석하였다[6]. 그러므로 성취기준 진술에 있어서 '성취기준 개발 이전에 교과의 핵심 개념과 고등 사고 기능 위주의 교육 내용 정선', '성취기준을 진술하기 위해 교과의 기능 혹은 과정에 대한 규명', '수행형의 성취기준의 진술' 등이 필요하다고 보았다[6]. 이는 브루너의 교과의 구조와 유사한 '교과의 이해'에 초점을 맞추고 있는 외국의 최근 교과 교육과정 개발 동향과 일치한다.

소프트웨어 교육의 목적은 컴퓨팅 사고력을 통한 문제 해결력의 향상이다[8]. 현재 다양하게 진행되고 있는

소프트웨어 교육은 컴퓨팅 사고보다 프로그래밍 언어의 문법 위주의 학습에 치중하고 있으며, 도구 중심 교육이 진행되고 있다[1]. 이는 곧 소프트웨어 교육의 목표인 컴퓨팅 사고를 통한 문제 해결에 도달할 수 있는 교육이 이루어지고 있지 않다는 것을 의미한다. 소프트웨어 교육의 목표에 도달하기 위해서는 교과의 핵심 개념과 고등 사고 기능을 이해하여, 이를 실제에 적용할 수 있고 전이를 높일 수 있는 방법상의 논의가 필요하다.

전통적으로 이루어지고 있는 교수설계 방법은 교육과정에 제시된 목표를 기준으로 수업 내용과 방법을 먼저 구상하고, 학생들이 목표에 도달하였는가를 판단하는 방법이다[7]. 이러한 교수설계 방법은 학생들이 학습 목표에 성공적으로 도달하였는지에 대한 학습증거를 확인할 수 있는 근거가 부족하며, 교사들이 단순히 교과서의 내용을 전달하거나 흥미위주의 수업을 조직하는 등의 문제가 나타난다[2].

Wiggins와 McTighe는 학생들이 교과의 핵심, 영속적 이해에 도달을 목표로 교수 학습을 체계적으로 설계하는 백워드 설계 모형을 제시하였다[14][15]. 백워드 설계 모형은 교과의 목표에 대한 학생들의 이해를 돕기 위한 단일 설계 및 수업 계획으로 학생들이 교과의 핵심에 대한 이해에 도달하였는지에 대한 평가를 먼저 수립하고 이를 기초로 수업활동을 구성하는 특징이 있다. 이에 다양한 교과에서도 학생들의 교과에 대한 핵심 개념과 고등 사고 기능의 이해를 우선하는 백워드 교수설계를 통한 단원의 교수 설계를 적용하고 있다. 백워드 교육과정 수업 설계와 관련된 우리나라의 연구를 살펴보면, 2015년 10월 현재 학위논문들을 포함하여 37편의 연구가 진행되었다. 이 중 14편의 연구는 백워드 설계 모형에 대한 이론적 탐구 및 고찰이며, 나머지 연구는 백워드 설계 모형을 적용한 국어, 수학, 사회, 과학, 체육, 미술의 교과의 학교 현장 적용 가능성에 대한 검증을 실시하였다. 이는 백워드 설계 모형에 대한 이론적 연구

및 다양한 교과에서의 교수 설계 및 현장 적용이 실시되고 있으며, 2015 개정 교육과정에서 필수로 진행되는 소프트웨어 교육에 대한 백워드 설계 모형의 이론적 연구 및 적용 연구의 필요성이 있음을 보여주고 있다.

이에 본 연구에서는 백워드 설계 모형의 소프트웨어 교육 적용 가능성을 탐색하고자 한다. 구체적으로 첫째, 백워드 설계 모형의 개관과 구조를 살펴보고 소프트웨어 교육의 배경 및 운영 지침을 살펴본다. 둘째, 소프트웨어 교육 지침의 알고리즘과 프로그래밍 영역의 단원의 교수설계를 개발한다. 끝으로 백워드 설계 모형이 소프트웨어 교육에 주는 시사점을 도출하여 본다.

2. 백워드 설계 모형의 이해

2.1. 백워드 설계 모형 개관

효과적으로 수업을 전개하기 위해 어떠한 교수방법과 전략을 적용할 것인가를 계획하고, 구상하는 과정을 교수설계라고 한다. 타일러는 교육과정 개발과 교수설계의 과정을 수업목표 설정, 학습 경험 선정과 조직, 평가 순으로 제시하였다[13]. 타일러의 교육과정 개발 모형은 구조적이고 체계적인 교수설계의 원형으로 널리 사용되었지만, 학생들이 학습 목표에 성공적으로 도달하였는지에 대한 학습 증거를 확인할 수 있는 근거가 부족하며[2], 교사들이 단순히 교과서에 있는 내용을 그대로 전달하거나 흥미 위주의 수업을 조직하는 것과 같이 목적성이 결여된 교수학습에 치중하고 있는 문제점이 나타났다[5].

학업 성취에 대한 평가가 새로운 화두로 떠오르면서 1990년대 중반부터 미국 학계에서 성취기준 중심의 교육 개혁이 꾸준히 제기되어 왔다[3]. 미국은 그동안의 학력 저하를 극복하고, 저학년 문맹 극복을 해결하기 위하여 2012년 까지 모든 학생들이 일정 수행 기준에 도달해야 하며 이를 객관식 평가 결과로 판단하는 것을 골자로 하는 Bush 정권의 NCLB(No Child Left Behind) 법안을 제정하였으며, 이 NCLB 법안 중심의 교육 개혁 운동을 시작하였다[16]. NCLB의 평가 프로그램은 교과 목적이 확연하게 드러나고, 학년에 따라 핵심적 학습 목표 및 활동 자원들이 체계적으로 제시된

성취기준에 의존하고 있다. 이에 그러므로 학교장과 교사들이 성취기준을 얼마만큼 체계적으로 분석하고 실행하느냐가 평가의 결과를 좌우하였다. 성취 기준 중심 교육에서의 ‘성취기준’은 한 학문의 전체적 범주, 큰 목적 혹은 지향점을 의미한다[4]. 하지만 성취기준에 의거한 평가 프로그램은 활동중심 수업, 진도 나가기식 수업으로 진행되는 폐해가 나타났다. 이러한 흐름 속에서 평가 전문가인 Wiggins와 McTighe는 이를 지적하며 학습자들의 심층적인 이해와 전이를 강조하는, 즉 ‘사실’에 대한 ‘기억’이 아니라 심층적인 지식의 구조에 대한 앎과 적용이 이루어졌는가를 평가하는 성취기준 중심의 교육과정 개발 방법인 백워드 설계 모형을 제안하였다. 백워드 설계 모형은 목적(바람직한 결과, 즉 목적 혹은 성취기준)에서 시작하여, 그 목적의 성취 결과로서 학습의 증거(수행)를 마련하고, 그리고 교육 활동을 적절하게 계획하는 설계 원리이다. 목표와 준거 설정을 교육과정 개발의 주요한 과업으로 본 타일러의 논거와 일치하지만 목표에 대한 성취 여부를 판단할 수 있는 평가를 우선적으로 고려하기 때문에 기존의 입장을 포워드(forward)로 보고 새로운 입장을 백워드(backward)라 명명하였다. 백워드 교육과정에서 강조하는 ‘심층적인 지식의 구조’는 부르너의 학문 중심 교육과정 이론과 그 궤를 같이 한다[2].

백워드 설계 모형은 교사들이 수업 계획 단계에서 단원의 기반이 되는 ‘중요한 내용’이 무엇인지를 학문의 핵심 개념과 원리들에 기초하여 끌어내고, 학습자들이 그 내용을 제대로 이해하였음을 드러내는 증거로서 평가과제를 먼저 개발하고 이를 적절히 수행할 수 있도록 교육활동을 계획하는 설계 모형이다. 그리고 이를 통해 학생들이 도달하여야 하는 목표와 활동을 일치시키는 데 초점을 두고 있다.

2.2. 백워드 설계 모형 구조

백워드 설계 모형에서는 교육목표가 아닌 평가가 수업의 중심이 된다. 이는 ‘평가되는 것만 가르치기(teaching to the test)’가 아닌, 학생들의 수업의 결과인 ‘이해’의 증거를 중심으로 수업을 조직한다는 의미이다. 백워드 설계 모형은 다음의 세 단계로 구성되어 있다.

2.2.1. 바라는 결과의 확인

첫 번째 단계는 바라는 결과의 확인으로 학생들이 '이해'해야 하는 것이 무엇인지를 밝혀 수업의 목적을 분명히 하는 단계이다. Wiggins와 McTighe는 이해의 의미를 유의미 추론, 전이 가능성의 관점으로 구분하였다[15]. 이해는 아는 것의 회상을 넘어 알고 있는 지식과 기능을 다른 맥락에서 적용, 분석, 종합, 평가할 수 있는 능력으로 학습자는 지식과 기능을 습득하고 지식의 추론 과정을 거쳐 이해에 도달하며 이해한 것을 전이할 수 있어야 한다[7]. 백워드 설계에서는 수업을 설계하기 위해 학문의 중심부에 있는 근본적이고 중요한 아이디어, 개념, 원리의 이해인 영속적인 이해(enduring understanding)를 추출하고, 이를 질문의 형태로 전환시켜 학습자들의 탐구와 사고를 촉진하며 학습 내용으로 안내하는 핵심 질문을 제시한다. 그리고 단원에서 학습자들이 알아야 하는 사실, 개념인 지식과 수행 능력인 기능을 제시한다[5].

2.2.2. 수용 가능한 증거 결정하기

두 번째 단계는 학습자들이 1단계에서 설정한 목표를 성취하였다는 것을 확인할 수 있는 증거를 개발하는 단계이다. Wiggins와 McTighe는 교사가 수행과제를 제작할 때 이해의 여섯 가지 측면을 고려해야 하며 가능한 많은 측면을 포함시킬 것을 제안하였다. 학생들은 과제를 수행할 때 1) 설명하고, 2) 해석하고, 3) 적용하고, 4) 관점을 가지고 바라보고, 5) 통찰로서 감정이입을 하고, 6) 스스로의 사고과정을 반성할 수 있어야 한다. 이 이해의 측면은 Bloom의 분석, 적용, 종합, 평가의 고차적 사고수준에 해당한다[5]. 하나의 과제가 모든 이해의 측면을 통합하여 드러낼 수는 없지만, 적어도 하나 이상의 이해 측면을 포함할 필요가 있다. Wiggins와 McTighe는 수행과제를 쉽게 설계할 수 있도록 수행과제에 포함될 요소들 머리글자를 따서 GRASPS 모델을 제시하였다. 수행과제는 학습자들이 실생활에 적용할 수 있는 상황(Situation)에서 어떤 목표(Goal)를 가지고 구체적인 대상 혹은 관중(Audience)을 고려하면서 특정 역할(Role)을 맡아서 기준(Standard)에 따라 결과물(Product)을 만들어내는 형식으로 개발된다. 이와 함

께 중요한 것은 학생들이 사실과 개념을 습득했는지를 확인하는 것인데 이는 다양한 평가 방법으로 검증될 수 있다.

2.2.3. 학습 경험과 수업 계획

백워드 설계의 세 번째 단계에서는 단원의 개별 차시에 포함되는 학습활동을 계획한다. 개별 차시에 해당하는 학습활동들은 단원 내에서 체계적으로 제시될 필요가 있으며, Wiggins와 McTighe는 학습활동 계획을 위해 WHERETO 방법을 제시하였다[14]. 이는 수업활동들은 1) 학생들이 단원의 궁극적인 목표와 방향이 무엇인지, 왜 그것을 배우는지 알 수 있도록 안내해야 하며 (Where, Why), 2) 관심을 집중시키며(Hook), 3) 과제수행에 필요한 지식과 경험, 도구, 노하우 등을 갖추게 하고(Equip, Enable), 4) 핵심 아이디어들을 다시 생각해 보고 반성하고 재점검(Rethink, Reflect, Revise)하게 하고, 5) 스스로의 진보를 평가할 수 있는 기회를 제공하고(Evaluate), 6) 학생 개개인의 강점, 재능, 흥미에 적합한 방식으로 다양화하고(Tailored), 7) 깊이 있는 이해를 최적화 할 수 있도록 조직(Organize)하는 방법이다. 학습경험 계획하기 단계를 통해 학습활동에서 핵심개념과 원리들을 설명하고, 여러 다양한 맥락과 상황에 적용하도록 도와줌으로써 학습의 전이를 도모한다.

3. 소프트웨어 교육

3.1. 소프트웨어 교육 배경

최근 지식과 정보가 가치 창출의 중요한 도구이자 국가 경쟁력을 결정하는 핵심요소로 자리를 잡고 있으며, 소프트웨어가 사회, 문화, 경제 등 여러 요소에서 중요한 영향을 미치고 있다. 최근 OECD의 PISA나 IEA의 ICILS에서는 정보화 시대에 요구되는 역량을 측정요소로 추가하여 시행하고 있으며, 영국, 미국, 인도, 핀란드 등에서는 소프트웨어 교육을 필수 교과로 지정하여 학생들의 역량을 강화하고 있다[8]. 이에 우리나라에서도 학생들에게 소프트웨어 교육의 기회를 제공할 수 있도록 소프트웨어 교육 운영 지침을 개발

하였다. 운영 지침에 따르면 소프트웨어 교육은 건전한 정보 윤리의식을 바탕으로 컴퓨팅 사고력 기반의 문제 해결력을 기르는데 기본 목적을 두고 있으며, 이를 토대로 다양한 분야에서 소프트웨어 역량을 융합하여 창의적으로 문제를 해결할 수 있는 창의융합 인재 향성을 목적으로 한다[9]. 이는 문이과 통합 교육을 통해 학생들이 인문학적 상상력과 과학기술 창조력을 갖춘 창의융합 인재로 성장할 수 있도록 우리나라 교육을 근본적으로 개혁하고자 하는 2015 개정 교육과정의 개정 방향과 일맥상통한다.

3.2. 소프트웨어 교육 운영 지침

소프트웨어 교육 운영 지침에는 초·중·고등학교에서 소프트웨어 교육을 실시하기 위한 전반적인 내용을 기술한다. 운영 지침에서는 소프트웨어 교육을 ‘생활과 소프트웨어’, ‘알고리즘과 프로그래밍’, ‘컴퓨팅과 문제해결’의 세 영역으로 구분하며, 초·중·고 학교급별 교육 목표와 내용 요소 및 성취 기준을 각각 제시하였다. 그리고 교수학습 방법 및 평가, 시수 확보 및 기존 교과와의 연계 운영 방안을 제시하여 2015 개정 교육과정이 적용되기 전까지 현장에서 소프트웨어 교육을 실행할 수 있는 안내서의 역할을 하고 있다.

4. 백워드 모형에 의한 소프트웨어 교육 운영 지침 교수설계 적용

4.1. 백워드 모형에 의한 교수설계 적용 방향

이에 본 연구에서는 앞서 분석한 백워드 설계 모형을 소프트웨어 교육 운영 지침의 교수설계에 적용함으로써 소프트웨어 교육의 효과적인 교수-학습 방법을 모색하고자 한다. <Table 1>에 제시된 소프트웨어 교육 운영 지침의 내용체계 중 초·중·고등학교의 ‘알고리즘과 프로그래밍’을 대상으로 백워드 교수설계안을 개발할 것이며, 백워드 설계 모형의 3단계 과정을 적용하고자 한다.

<Table 1> Software Education Content Framework

Domain	Elementary School	Middle School	High School
Life and software	<ul style="list-style-type: none"> • Me and software • Information Ethics 	<ul style="list-style-type: none"> • The importance of utilizing the software • Information Ethics • Configuration of the information device and Information exchange 	<ul style="list-style-type: none"> • Computing and Information Life • Information Ethics • Operation of the information device and Information Processing
Algorithms and Programming	<ul style="list-style-type: none"> • Experience in problem solving process • Experience algorithm • Experience programming 	<ul style="list-style-type: none"> • Type of information and Structured • Understanding of computational Thinking • Understanding algorithms • Understanding Programming 	<ul style="list-style-type: none"> • Representation and management of information • Practical computational thinking • Practical algorithm • Practical programming • Problem solving and programming
Computing and solving problems		<ul style="list-style-type: none"> • Computational Thinking-based problem solving thinking 	<ul style="list-style-type: none"> • Computational Thinking-based Convergence activities

본 연구에서는 Wiggins와 McTighe가 개발한 백워드 설계틀을 위한 템플릿에 따라 교수설계를 구성하였다.

4.2. 단계별 백워드 교수설계

4.2.1. 1단계(바라는 결과 확인하기)

1단계는 단원 수업을 통해 학생들에게 기대하는 학습 결과를 명확히 규정하는 단계이다. 이 단계는 추출된 일반화, 핵심 질문, 사실과 개념적 지식, 기능/과정적 지식 요소를 포함한다[5].

초·중·고등학교에서는 생활과 소프트웨어, 알고리즘과 프로그래밍의 2개 대영역으로 구성되어 있고, 각 영역별 교육목표가 있다. 세부적으로 알고리즘과 프로그래밍 영역에는 문제해결 과정의 체험, 알고리즘의 체험, 프로그래밍의 체험의 3개 중영역으로 구성되어 있으며, 각각의 중영역은 내용 요소와 성취 기준이 있다. 백워드 설계 모형에서는 ‘목표 풀기’ 전략을 이용하여 학생들이 알고 있어야 할 지식과 할 수 있는 기능을 개발한다.

‘목표 풀기’ 전략은 목표에서 주요 아이디어와 이해를 밝혀내고 수행을 추출하는 과정이다[11]. 따라서 본 연구에서는 소프트웨어 교육 운영 지침의 성취 기준을 이용하여 바라는 결과를 추출하였다. 교과와 진정한 이해를 위해서는 대영역의 교육목표와 중영역의 성취 기준을 유기적으로 지도할 필요가 있다. 이에 본 연구에서는 대영역의 전체 성취기준을 중심으로 교육과정 풀기를 진행하였다. 교육과정 풀기에 따라 작성한 1단계 내용은 <Table 2>와 같다.

<Table 2> Identify Desired Results

Stage 1 - Identify Desired Results	
Established Goal Student can <ul style="list-style-type: none"> • Understand a given problem. • Simplify a given problem. • Describ in the order for solving the problem • Address the problem of the proposed problem solution and the method of improve. • Understand the concept of the algorithm. • Express problem solving procedure using sequence, repeat, select structure via picture or symbol. • Understand a simple algorithm by activities. • See the basic elements of the programming languages. • Make equal to a given program. • Create a program by modifying the given program. • Create simple program that I think. 	
Understanding(s) Student can <ul style="list-style-type: none"> • Be simplified by structuring a wide range of real-world problems. • Order for problem solving method. • Understand programming languages and basic elements. • Implement algorithm to programming. 	Essential Question(s) <ul style="list-style-type: none"> • How can structure the problem in real life? • What is the method of order is a problem solving method? • What are the basic elements of programming? • How to solve the problem by programming elements?
Students will know <ul style="list-style-type: none"> • Problem structure, simplify the problem, algorithms, programming basic elements, practical programming 	Students will be able to <ul style="list-style-type: none"> • Data analysis • Abstract • Automate

4.2.2. 2단계-다양한 이해의 증거 결정하기

2단계에서는 GRASPS 모델을 적용하여 <Table 3>의 수행과제를 제시하였다. 수행과제와 함께 1단계에서 설정한 이해에 대한 그 밖의 증거와 학생들이 어떠한 사고과정과, 학습자로서 얼마나 성장하였는지를 인식할 수 있는 반성의 기회인 자기 평가를 제시하였다.

<Table 3> Determine Assessment Evidence

Stage 2 - Determine Assessment Evidence	
Performance Task(s) <i>Please make automatic doors!</i> We try to create an automatic door to the school entrance(S). Doors are made, but the software does not exist, so automatic doors does not operate(S, R). Let’s design a automatic doors program for students to take advantage of convenience automatic door(G, A). The design have to show the automatic door operating procedures(S). And you should solve the problem that unfinished software programming about automatic doors made of educational programming languages(P).	Other Evidence <ul style="list-style-type: none"> • Descriptive Evaluation • Informal observation Evaluation Self-assessment <ul style="list-style-type: none"> • Record the learned information and questions • Presents criteria for the task performed in the form of a checklist.

수행과제에 대한 타당하고 신뢰로운 증거를 제시하기 위해 <Table 4>와 같은 분석적인 루브릭을 개발하였다.

<Table 4> Performance Task rubric

Domain	Excellent	Normal	Needs effort
Design	The problem was clearly structured and the problem solving method using the flow-chart was shown systematically.	The problem was structured and the problem solving method using the flowchart was shown.	Do not structure the problem and the problem solving method using the flowchart was not shown.
Programming	Modified in accordance with the problem solving	Modified in accordance with the problem	Modified in accordance with the problem solving

method, a un-
finished pro-
gram was to
be functioning
correctly.

solving
method, a un-
finished pro-
gram was to
be
functioning.

method, a un-
finished pro-
gram was not
to be
functioning.

4.2.3. 3단계-학습경험과 수업 계획하기

본 연구에서는 14개의 학습활동을 고안하였으며, 학습활동에 해당하는 WHERET의 요소를 표시하였다. 초등학교 40분 단위를 고려하여 총 7차시의 수업 기간을 고려하여 학습활동 'O'의 요소를 조직하였다. 계획된 학습활동 및 세부적인 차시 내용은 <Table 5>와 같다.

4.3. 백워드 설계 모형에 의한 소프트웨어 교육 교수설계의 효용성

본 연구에서는 교육과정의 성취기준에 기초하여 단원을 개발하는 백워드 설계 모형을 적용하여 소프트웨어 교육의 '알고리즘과 프로그래밍' 단원을 효과적으로 지도하기 위한 교수설계를 작성하였다. 작성된 내용에서 도출된 백워드 설계 모형의 소프트웨어 교육 교수설

계의 효용성은 다음과 같다.

첫째, 백워드 설계 모형은 소프트웨어 교육의 지식/기능과 학생의 경험을 연결한다. 백워드 설계 모형은 이해를 위한 교수와 평가를 진행하므로 학습자들이 핵심개념을 이해하고 이를 다양한 상황에 적용할 수 있어 학습의 전이를 높일 수 있다. 그리고 실제 현실과 연결된 도전적인 수행과제를 제시하기 때문에 학생들의 흥미를 높일 수 있다. 이는 알고리즘과 프로그래밍 단원의 목표인 '알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 컴퓨팅 사고로 이해할 수 있다.'와 연결된다.

둘째, 백워드 설계 모형으로 소프트웨어 교육 운영 지침의 내용을 보다 유의미하고 효과적인 수업으로 구성할 수 있다. 백워드 설계 모형은 교육과정 개발자로서의 교사의 역할을 강조한다. 교사가 백워드 설계 모형을 이용하여 단원을 개발할 때에는 가르칠 내용의 영속적 이해를 중심으로 수업을 재조직한다. 그 다음 학생의 이해를 점검할 수 있는 증거를 결정 한 후 이를 성공적으로 수행할 수 있도록 학습활동을 설계한다. 이러한 과정 속에서 교사는 교육과정의 분권화 및 자율화를 실현하여, 국가수준 교육과정의 실행자이면서 동시에

<Table 5> Plan Learning Experiences and Instruction

Stage 3 - Plan Learning Experiences and Instruction						
Learning Activities	WHERET Element					
	W	H	E	R	E	T
1. Causing the students' interest by showing the case of problem solving via computational thinking.						
2. Introducing essential question, and thinking together about performance task.						
3. Discussing about the method of abstract essential contents in daily life(1. include a computing, 2. not include a computing).						
4. Demonstrating the representation process of problem solving via flowchart.						
5. Representing various process of problem solving via flowchart and improving that process.						
6. Learning conditional structure using programming example.						
7. Learning loop structure using programming example.						
8. Demonstrating the process of convert flowchart to programming language.						
9. Programming the various flowcharts suitable for learner's level.						
10. Guiding the performance task.						
11. Structuring the performance task by pairs.						
12. Ordering the method of solving performance tasks.						
13. Programming the various unfinished project that suitable for learners level.						
14. Announcing the results and rethinking what they learned.						
Organize						
lesson 1 : 1, 2, 3	lesson 2 : 4, 5	lesson 3 : 6, 7	lesson 4 : 8, 9			
lesson 5 : 10, 11, 12	lesson 6 : 13	lesson 7 : 14				

학교와 학생 실정에 맞는 교육과정을 개발하는 개발자로서의 역할을 수행할 수 있다.

셋째, 백워드 설계 모형은 소프트웨어 교과가 갖는 궁극적 목적과 단위별 목표 혹은 수업 단위 학습 목표를 유기적으로 연결할 수 있다. 교육과정의 내용에서 브루너의 지식의 구조, 즉 어떤 상상이나 현상의 심층부에 있는 핵심적 아이디어, 개념 혹은 원리를 이해할 수 있는 교과의 추출된 일반화와 지식, 기능을 기초로 단원을 설계하므로 학생들의 영속적 이해(enduring understanding)에 한걸음 더 접근할 수 있다. 그리고 학생들이 영속적 이해에 도달하였는지를 판단할 수 있는 수행과제를 제시하고 학생들이 이에 도달할 수 있도록 교수학습을 설계하므로 학습 내용의 구조적이며 체계적인 설계가 가능하다.

5. 결론

본 연구에서는 백워드 설계 모형을 적용하여 초등학교 소프트웨어 교육의 알고리즘과 프로그래밍 단원을 개발하고, 소프트웨어 교육 교수학습설계에 대한 시사점을 도출하였다. 연구목적 달성을 위하여 문헌연구 방법을 활용하여, 소프트웨어 교육 운영 지침, 백워드 설계 모형, 이해중심 교육과정에 관한 문헌을 검토하였다. 연구자는 백워드 설계 모형의 3단계에 따라 초등학교 소프트웨어 교육의 '알고리즘과 프로그래밍' 단원을 설계하였다. 먼저 소프트웨어 교육 운영 지침에 제시된 단위 목표와 내용 체계표, 성취기준을 이용하여 추출된 일반화, 핵심질문, 사실과 개념, 기능/과정적 지식을 도출하였다. 다음으로 GRASPS 방법을 적용하여 알고리즘과 프로그래밍 단원의 수행과제와 루브릭을 개발하였다. 마지막으로 알고리즘과 프로그래밍 단원의 구체적인 수업내용을 WHERETO 방법을 적용하여 개발하였다.

백워드 설계 모형을 이용하여 단원을 설계한 결과 다음의 시사점을 도출할 수 있었다. 먼저 백워드 설계 모형은 소프트웨어적 지식/기능과 실제 현실을 연결한다. 그리고 백워드 설계 모형으로 소프트웨어 교육 운영 지침의 내용을 보다 의미미하고 효과적인 수업으로 구성할 수 있으며, 소프트웨어 교과가 갖는 궁극적 목적과

단위별 목표 혹은 수업 단위 학습 목표를 유기적으로 연결할 수 있다.

본 연구에서 도출한 소프트웨어 교육에서의 백워드 교수 설계는 소프트웨어 교육 운영지침, 백워드 설계모형, 이해중심 교육과정의 문헌 분석을 통해 이루어졌으므로 교육의 효과성 및 만족도를 검증하기 위해 도출된 교수 설계를 기반으로 교사와 학생 대상의 현장 적용 가능성에 대한 후속 연구가 필요하다.

소프트웨어 교육에 대한 관심이 높아지고, 수업 방법에 대한 논의가 진행되고 있는 이 시점에서 백워드 설계 모형을 적용한 단위 교수학습 설계는 교과 목표의 효율적인 달성 방법의 일환으로 긍정적으로 고려될 수 있을 것으로 기대한다.

참고문헌

- [1] Cha, S., Kim, J., Kim, J., & Lee, W. (2009). Study on recognition changes regarding programming education and necessity. *Journal of Korean Association of Computer Education*, 12(1), 1-13.
- [2] Cho, J. (2005). Thinking about backward curriculum design. *Journal of Korean Society for Curriculum Studies*, 23(1), 63-94.
- [3] Darling-Hammond, L. (1995). Setting standards for students: The case for authentic assessment. *The Educational Forum*, 59(1), 14-21.
- [4] Kang, H., & Ryu, J. (2010). In search of the authentic understanding of subject knowledge: Looking at the effect of backward design. *Philosophy of Education Society of Korea*, 40, 1-37.
- [5] Kim, K., Ohn, j. (2011). Transfer of learning understanding creativity. Seoul: EducationAcademi.
- [6] Lee, K., Jung, Y., Lee, K., Kim, S., Paik, N., & Ohn, J. (2014). A study on direction of subject curriculum development(CRC 2014-7). Korea Institute for Curriculum and Evaluation.
- [7] Lee, B. (2011). A study on the instructional design of 'Library and information life' subject based on backward design model. *Journal of the Korean*

Biblia Society for Library and Information Science, 22(3), 5-24.

[8] Ministry of Education (2014). Software education standard curriculum overview.

[9] Ministry of Education (2015). 2015 Revised curriculum. [Http://www.Moe.Go.kr/web/45859/ko/board/view.do](http://www.Moe.Go.kr/web/45859/ko/board/view.do)

[10] Oh, I., Yang, J., & Kim, S. (2007). Status of domestic computing education, problems, and improvement. *Journal of Communications of The Korea Information Science Society*, 25(7), 14-20.

[11] Park, I. (2012). A study on an application of backward design for Mathematics. *The Journal of Curriculum Studies*, 30(4), 109-137.

[12] So, K. (2015). Issues in the general guideline draft for the 2015 national curriculum: Remaining tasks for subject matter curriculum development. *Journal of Korean Society for Curriculum Studies*, 33(1), 195-214.

[13] Tyler, R. W. (2013). Basic principles of curriculum and instruction University of Chicago press.

[14] Wiggins, G. P., & McTighe, J. (2005). Understanding by design Ascd.

[15] Wiggins, G. P., McTighe, J., Kiernan, L. J., & Frost, F. (1998). Understanding by design Association for Supervision and Curriculum Development Alexandria, VA.

[16] Wright, P. W., Wright, P. D., & Heath, S. W. (2006). No child left behind Harbor House Law Press.

관심분야: 소프트웨어 교육, 교육용 프로그래밍 언어, 소프트웨어 교육과정
e-mail: yhbest12@naver.com



구 덕 회

2000 한국교육원대학교 대학원 박사
2000~2003 한국교육학술정보원 선임연구원
2003~2009 대구교육대학교 컴퓨터교육과 교수
2009~현재 서울교육대학교 컴퓨터교육과 교수
관심분야: 컴퓨터교육이론, 프로그래밍 교육, 원격교육, 융합교육, 디지털스토리텔링
e-mail: dhk@snue.ac.kr

저자소개



이 영 호

2013 서울교육대학교 교육대학원 컴퓨터교육과 졸업
2015 서울교육대학교 생활과학·컴퓨터교육과 박사과정
2010~현재 서울강덕초등학교 교사

