

블룸 분류 기반 초중등 프로그래밍교육의 평가 기준 탐색

신수범

공주교육대학교

요 약

정보과 교육이 정규교과과정에 편입됨에 따라서 학습자의 정보과 교육과정에 대한 학업 수준을 정확히 진단하고 그에 맞는 등급을 부여할 수 있도록 일종의 이론적 기반이 필요하다. 이에 본 연구에서는 개정 블룸의 교육목표 분류와 지식 유형 분류 이론을 통해 정보과 교육과정의 프로그래밍 분야에 대한 기준을 제시하고자 하였다. 그리고 블룸의 “기억”에서부터 “제작” 단계까지 프로그래밍교육 평가 기준을 제시하였다. 그리고 블룸의 지식 유형 기준에 따라서 사실적 지식에서부터 메타인지 지식까지 프로그래밍의 지식 유형 예제를 제시하였다. 또한, 본 연구에서 프로그래밍 활동을 위해 가장 중요한 블룸의 분류는 “적용”, “제작” 그리고 절차적 지식 유형으로 분석하였다. 이러한 블룸의 이론 기반 프로그래밍 평가 기준에 대하여 소프트웨어교육 전문가를 통하여 델파이 조사를 하였다. 조사 결과 목표 분류 기준 신뢰도가 0.90, 합의도가 0.85였으며 지식 유형 분류 기준으로는 신뢰도가 0.90, 합의도가 0.79로 나타났다. 이에 본 연구에서 제시한 기준에 대해서 전문가들은 동의하는 것으로 해석할 수 있었다.

키워드 : 정보과 교육 평가, 프로그래밍교육 평가, 블룸 분류체계

Research on the Assessment Criteria of Programming Education based on Bloom's Taxonomy in the Elementary and Secondary School

Soobum Shin

Gong-Ju University of Education

ABSTRACT

It needs theoretical assessment fundamental for informatics curriculum to judge appropriate grades and measure academic standard of an learner according to be included in the conventional curriculum. Thus this study tried to present an criteria on programming area of an informatics curriculum through bloom taxonomy and knowledge type. And it presented assessment criterion on each steps from “Remember” to “Create”. And we presented knowledge type examples of programming such as Factual to Metacognitive based on Bloom's knowledge types. Also we analysed that most important level or type is Apply Level, Create Level and Procedural Knowledge. We investigated for each criterion of programming assessment based on bloom's theory through Delphi method. And the result of this investigation was that area of bloom's taxonomy was CVR 0.90, Validity 0.85 and area of knowledge type was CVR 0.90, Validity 0.79. So it can decide to accept for our assessment criteria of programming education based on Bloom theory.

Keywords : Assessment of Informatics Education, Programming Education Assessment, Bloom Taxonomy

본 연구는 공주교육대학교 2017년도 자유연구과제 사업에 의해 지원되었음.

논문투고 : 2017-08-30

논문심사 : 2017-09-04

심사완료 : 2017-10-04

1. 서론

교육부는 2018년부터 중학교를 시작으로 정규교육과정에서 프로그래밍 교육을 실시할 예정이다. 그동안의 소프트웨어교육은 비교과과정에서 자유롭게 이루어져 왔다. 즉, 교육과정이나 평가방법에 있어서 교육 주관기관에 의해 자율적으로 운영을 해 왔다. 하지만 정규교육과정에서의 평가방법은 엄격하게 관리되고 명확한 기준이 필요하다. 비교과 과정에서의 평가는 학습자의 만족도 조사와 학업성취수준을 측정하는 것이 일반적이었지만 정규교과에서의 평가는 학습자의 학업성취 등급도 중요한 목적을 두고 있다. 학습자의 학업성취수준과 등급을 부여하기 위해서는 공정성과 객관성이 확보되어야 한다. 이를 위해 일반적으로 교육학에서는 학업 내용에 교육목표 분류를 기준으로 구분하여 등급을 차별화하고 학업성취수준을 측정하고 있다. 더구나 프로그래밍 활동은 블록의 교육목표의 분류 기준에서 비추어 볼 때 지식수준의 내용은 프로그래밍 역량과 직접적 관련성이 떨어진다. 이에 프로그래밍 교육내용에 대한 분류도 필요한 상황이다.

하지만 국내의 초중등 및 대학 수준의 소프트웨어교육에서 교육내용을 일정한 기준에 의해 분류하고 난이도를 구분한 연구는 매우 미흡한 상황이다. 반면에 해외에서는 초보 프로그래밍 과정에서의 평가방법에 대한 연구는 지속적으로 나타나고 있다. 이에 본 연구에서는 해외의 사례를 분석하여 국내의 소프트웨어교육 내용에 대한 평가 기준을 제시해 보고자 한다.

이를 위해 2018년도부터 실시되는 국내 소프트웨어의 목표와 주요 교육내용을 분석하여 평가 기준의 자료로 활용하고자 한다.

2. 초중등 프로그래밍교육 주요 내용

2.1 프로그래밍교육 목적

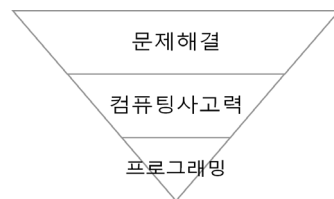
2018년도 정보교육의 핵심은 코딩교육이지만 코딩은 프로그래밍 언어 의존적인 용어라고 판단되기 때문에 본 연구에서는 프로그래밍 언어라는 용어로 제시하

고자 한다. 그리고 국내의 정보과 교육과정에서 프로그래밍교육의 도입 목적에 대해서 진술하면 다음과 같이 요약할 수 있다[8][10][11].

첫째 프로그래밍교육 목적은 사고력 향상에 있다.

교육부(2015)의 자료에 의하면 문제해결력을 갖춘 창의인재를 위해서 소프트웨어 교육을 강화하는 것으로 되어 있다[3]. 문제해결력은 컴퓨팅사고력의 일반화된 표현이며 프로그래밍을 통해 구현된다[10]. 즉 프로그래밍의 목적은 컴퓨팅사고력 향상이자 문제해결력 신장이라고 할 수 있다. 문제해결력, 컴퓨팅사고력을 요약적으로 표현하면 사고력이라고 할 수 있다. 즉 프로그래밍교육의 근본적인 목적은 사고력 향상이다.

둘째, 컴퓨팅사고력의 구체화 단계로 필요하다. 컴퓨팅사고력은 컴퓨터과학과 연계한 인간의 인지 활동에 대한 것이다. 컴퓨팅 사고력의 추상화, 알고리즘 활동을 통해서 정형화된 내용을 프로그래밍으로 구체화하게 된다. 이에 대해 Selby는 다음과 같이 문제해결, 컴퓨팅사고력, 프로그래밍의 관계를 그림으로 제시하였다[4].



(Fig. 1) Relation of Core Concepts

(Fig. 1)의 3개 개념은 상호 간 영향을 주는데 밑으로 내려올수록 컴퓨터과학과 연관된 개념이라고 할 수 있다. 그리고 프로그래밍은 컴퓨팅사고활동 일부를 구체화하는 활동으로 체계화한 모습이다. 국내의 중등 정보과 교육과정에서도 컴퓨팅사고력 향상을 위한 프로그래밍 교육과정을 제시하고 있다.

셋째, 기본적인 프로그래밍 언어 사용 능력 향상이다.

정규교육과정에서의 프로그래밍 활동이기 때문에 국민 누구나가 알고 있어야 하는 소양 수준에서 그 도구의 범위를 결정해야 할 것이다. 이와 같은 기본적인 프로그래밍 과정을 통해서 미래사회에서 필요로 하는 기본적인 역량을 습득하고 사회가 시스템적으로 어떻게 움직이는지 알 수 있다. 또한 ACM 자료에 의하면 컴퓨

터과학에 기반을 둔 산업사회의 인력 수요는 지속 증가할 것으로 예상된다[27]. 이에 기본적인 프로그래밍 언어 사용 능력은 미래사회의 기본 역량으로 자리 잡을 수 있을 것이다.

2.2 소프트웨어 주요 교육내용

국내 소프트웨어교육의 핵심 교육내용은 코딩교육 즉 프로그래밍교육이라고 할 수 있다.

다음의 <Table 1>은 초중등 소프트웨어교육과정에서 프로그래밍과 직접 관련이 있는 내용을 요약한 내용이다[10].

<Table 1> Summarized Domestic SW curriculum of Informatics Related Subjects

	Ele. Practical Tech. Subject	Sec. informatics Subject
Abstraction		- Extract Core Elements from Understanding and Analysis of Problem
Algorithm	- Procedure Problem Solving	- Algorithm Understanding & Expression
Programming	- SW Understanding - Programming Elements and Structure	- Programming Language Dev. Environments - Variables and Calculation - Input & Output - Control Structure - Prg. Application
Physical Computing		- Implementation of Physical Computing

위 <Table 1>에 의하면 초등에서는 컴퓨팅사고력에 대한 내용은 생략되고 중등에서 추상화 영역이 삽입되었고 프로그래밍 영역이 초중등에 걸쳐서 가장 강조되고 있는 영역이라고 할 수 있다.

2.3 초보자 프로그래밍 활동 유형

위의 <Table 1>은 프로그래밍 영역별로 정리한 것이고 실제 프로그래밍 학습자의 활동은 코드 읽기, 코드 작성으로 구분될 수 있다. 물론 코드 작성을 위해서 문

제를 분석하고 세분화하는 활동을 할 수 있지만, 프로그래밍에 집중하여 초보 프로그래밍 학습자의 활동을 구분하자면 코드 읽기와 코드 작성 활동으로 구분할 수 있다. 코드읽기 활동은 완성된 프로그램을 해석하고 추적하는 것을 포함한다.

코드 읽기는 코드 제작을 위해서 거쳐야 하는 중요한 과정으로서 Soloway(1986)는 코드 읽기의 중요성을 제시하였다. 또한, 그는 전문프로그래머는 코드를 읽으면서 두뇌에서 시뮬레이션을 하면서 예측활동을 한다고 제시하였다(재인용, [12]). 이에 코드 읽기를 강화하기 위해 본 연구에서는 코드 읽기와 코드 추적활동으로 별도로 분리하여 제시하고자 한다.

코드 추적 활동은 일종의 디버깅 활동으로서 전체 프로그램 결과 또는 모듈 단위 프로그램 결과 예측과 흐름 파악 활동을 하는 것이다.

반면에 코드 제작 활동은 실제 프로그램을 제작하는 활동이다. 실제 코드 제작 활동 안에는 문제를 분석하고 알고리즘을 수립하는 과정 등 다양한 활동을 포함할 수 있지만, 초보 과정에서 해당 활동의 비중은 작을 것으로 판단하고 코드제작으로 압축하여 제시하고자 한다.

이에 본 연구에서는 초중등 단계 프로그래밍 활동 유형을 코드 읽기, 코드 추적, 코드 제작의 3단계로 제시하고자 하며 다음 <Table 2>와 같이 구체적으로 설명할 수 있다.

<Table 2> Categorized of the Beginner Programming Activities

Type of Programming Activities	Type Explanations
Code Reading	- A kind of code analysis such as program aim, reason why this module is written by others or instructor - Code Reading of Totally, Segments or a Statement unit
Code Tracing	- Prediction of Program Executing Sequence, Changing Value of Variables or Arguments - Activities of Error Fixing, Code Replacement
Code Writing	- Module or Total Code Writing to Solve Given Problem

3. Bloom 이론 기반 프로그래밍교육 평가 기준

전술하고 있는 바와 같이 교육 평가는 학습 진척도와 등급을 부여하기 위해서 진행된다. 이를 위해 소프트웨어교육의 목표와 주요 내용을 살펴보았으며 이를 바탕으로 평가 기준을 제시해 보고자 한다. 평가 기준은 개정 Bloom 이론을 적용하고자 한다. 그 이외에 다양한 평가 이론이 있지만, 실습 중심의 교육내용, 다양한 초보자 프로그래밍교육 평가 선행 연구에서 개정 Bloom의 기준을 많이 채택한 바 본 연구에서도 이를 이용하고자 한다.

3.1 Bloom 목표 기준에 의한 프로그래밍교육 목표 분류

Bloom은 인지, 정의, 태도 영역에 걸쳐서 각각의 교육 목표의 단계를 제시하였으며 인지 영역에서의 단계를 6 단계로 제시하였다[2]. 그리고 Bloom의 교육목표 분류는 기존 분류체계와 개정된 분류체계로 구분되는데 개정된 분류체계와 차이는 명사를 동사형으로 바꾸고 마지막 단계를 변경한 점이다[1]. 그리고 본 연구에서는 Bloom의

교육목표 분류를 표현할 때 동사형 종결어미는 생략해서 표현하였다.

<Table 3>은 개정 Bloom의 6단계 교육목표 분류 기준을 설명하고 초보 프로그래밍 선행 연구에서 제시한 Bloom 기준 프로그래밍교육 평가 기준을 요약 제시한 것이다[5][12][13][14]. 그리고 본 연구에서 제시한 초보자용 3가지 프로그래밍 학습 유형을 연계하였다.

그리고 <Table 3>의 내용을 다음과 같이 부연 설명을 하고자 한다.

첫째 기억, 이해, 적용 단계는 유일한 정답이 존재하는 수준의 코드 추적과 코드 해석 능력을 평가한다.

지식 단계는 재해석, 연계, 적용, 비판 활동 없이 회상해 내는 것으로서 가장 낮은 수준의 평가라고 할 수 있다.

이해단계는 사실, 원리에 대해 자신의 언어로 재해석하고 표현하는 능력으로서 프로그램 문법, 이미 알려진 기본 알고리즘에 대해 자신의 언어로 설명하거나 다른 방법으로 대체 표현하는 수준이다. 예를 들면 결괏값 예상하기, 프로그램 목적 설명하기 등이 있다.

적용단계는 알려진 규칙, 기능을 새로운 문제 상황을 적용하는 수준이다. 적용 단계에서 읽기 활동은 프로그래밍 원리를 적용한 코드에 대한 학습자의 의견을 측정

<Table 3> Evaluation Criterion and Characteristic on Programming Education Based on Bloom's Taxonomy

Bloom Taxonomy	Criterion Explanation	Evaluation Criterion of Programming Education	Programming Type	Evaluation Focus
Create	Put pieces together to form something new or recognize components of a new structure.	- Program Writing Completely range from Low Level to high level	Writing	Problem Solving
Evaluate	Make Judgements based on criteria and standards	- Criticize or Effectiveness Analysis about Completed Code, Module	Reading	
Analyze	Break a concept down into its parts and describe how the parts relate to the whole	- Classify each meaningful module from the whole code Compare Modules or Find Difference		
Apply	Carry out or Use a Procedure in a given situation	- Make Code Segments, Part of Algorithm using rule or principle of Programming Language or Algorithm - Interpret Program or Algorithm using Known or Learned Rule or Principle of Programming Language or Algorithm	Writing, Reading	Understanding of Rule, Principle
Understand	Construct meaning from Instructional Message, including oral, written and graphic information	- Expect Output of Program - Trace Execution of Program Sequence or Algorithm - Explain Goal, of Program or Module - Transfer Algorithm such as Pseudo Code, Flowchart to Code or Reverse	Reading Tracing	
Remember	Retrieve the right information from memory	- Remember Language Grammar		

하는 경우에 나타날 수 있다.

또한 이미 알려진 내용은 규칙 자체에 논란이 없기 때문에 적용하는 절차와 결과가 명확할 수밖에 없다.

이와 같이 3개의 단계에서 정답이 유일할 수 있는 것은 언어의 문법, 구조 등 프로그래밍 언어 규칙에 초점을 맞추고 있기 때문이다. 언어의 규칙에는 예외가 있을 수 없으며 정답은 유일하기 때문에 선다형 방식의 표준화 평가가 가능하다. 그리고, 코드 이해, 추적 등의 활동에서 예외가 없는 평가 문항을 통해 평가할 수 있다.

또한, 코드 제작 활동이 표면적으로는 문제해결 활동의 일환으로 볼 수도 있지만 코드세그먼트 쓰기 등 언어의 규칙에 초점을 맞추어 코드 제작 활동을 할 경우에는 적용단계라고 할 수 있다.

블록 분석, 평가의 단계는 다양한 답안이 존재하거나 답안의 질적 수준을 평가하는 코드 읽기 활동을 평가한다.

블록의 분석단계부터는 고등정신능력을 측정하는 영역으로서 대안평가 영역으로써 적합한 것으로 제시하였다[15].

분석 단계는 전체 프로그램에서 프로그램의 목적에 맞게 모듈 단위로 나누고 타 모듈과의 차별성을 파악하거나 전체 프로그램에서 모듈의 경중을 파악하는 내용으로 평가 문항을 출제할 수 있다[5].

학습자의 기준에 의해서 비교 분석, 조사 활동을 하는 것이기 때문에 언어 규칙, 원리를 분석하기보다는 문제 분석, 알고리즘 수립 등의 활동이 적합한 것으로 판단된다. 그리고 Cynthia(2015)의 경우도 블록의 분석단계는 프로그래밍 전 단계로 컴퓨터사고력의 분해, 추상화를 적용하였다[3].

이러한 Cynthia의 판단은 프로그래밍교육 영역 평가에서 분석, 평가의 단계는 정답이 하나이거나 기준에 제시된 답을 제시하는 것이 아니라 새로운 문제해결 방법 등을 독창적으로 제시하는 것을 의미한다. 이와 같은 기준에서 코드 읽기와 코드 추적 활동은 이전 단계보다 복잡적이고 다양한 답안이 존재할 수 있어야 한다.

그러기 위해서는 분석해야 하는 프로그램이 다양한 함수를 포함하는 등 복잡하거나 학습자가 이전에 배우지 않은 원리나 규칙에 대한 내용을 포함해야 한다.

그리고 현실의 복잡한 문제를 해결하기 위한 프로그래밍 과정은 문법적인 오류가 나타날 수도 있지만, 초점은 문제 해결에 있다. 즉 블록의 분석, 평가 단계는 현

실의 문제를 모델화해서 적합한 명령어와 제어구조를 선택하는 과정이다. 이것은 문제를 분석하고 적합한 명령어를 찾는 활동까지 내포하고 있기 때문에 보다 수준이 높다고 할 수 있다[9].

셋째 제작은 다양한 답안이 존재하는 코드 제작 활동을 평가하는 단계이다.

컴퓨터 프로그래밍은 기본적으로 다양한 방법이 존재한다. 문제에서 조건을 제한적으로 제시할 경우에는 도달 방법이 제한적일 수 있지만, 기본적으로 다양한 경로와 도구가 존재한다. 또한, 코드 제작은 학습자가 스스로 문제를 분석하고 최적의 알고리즘 설계와 프로그램 명령어의 선택 등이 필요하고 이를 위해 다양한 부가 능력들 역시 필요하기 때문에 가장 어려운 수준의 평가라고 할 수 있다. 하지만 전술하고 있듯이 조건에 따라서 제작은 상대적으로 용이할 수도 있으며 어려울 수도 있는 복잡도를 포함하고 있다. 즉 <Table 3>에서 제시한 낮고 높은 수준에 대한 다양한 기준이 존재할 수 있다. 이에 대해 Judy Sheard et al.(2013)는 블록의 이론 이외에 다양한 복잡도를 조절할 수 있는 변수가 있음을 제시하고 있다[7]. 그는 문제의 용어 난이도, 요구하는 조건과 요구하는 코딩 분량 등 6가지의 복잡도 변수를 제시하였다. 특히 코딩 길이에 대해서 10줄은 낮은 수준 10-20줄은 중간 수준 그리고 그 이상은 상위 수준으로 기준을 제시하였다. 이는 블록의 제작하다 수준에서도 다양한 복잡도가 있음을 보여주는 연구라고 판단된다.

이와 같은 개정 블록 기반 교육목표 분류에 의한 프로그래밍교육 평가 기준은 학습자가 어떤 수준에서 행동하는지에 대한 지표로 활용 가능한 점에 의의가 있다. 즉 기억하다, 이해하다 등의 동사는 학습자가 행동하는 수준을 나타내는 지표라고 할 수 있다.

3.2 블록 지식 유형 기준에 의한 프로그래밍 교육 내용 분류

블록은 교육목표 분류와는 별도로 지식의 유형을 별도로 제시하였다.

블록과 Anderson은 지식의 종류를 사실적 지식, 개념적 지식, 절차적 지식, 메타 인지적 지식 등의 4가지 유형으로 분류하고 6단계의 교육목표 유형과 상호작용이 필요하다고 제시하였다. 사실, 개념, 절차적 지식은 블록

<Table 4> Example of Programming Knowledge Type based on Bloom Knowledge Type

Knowledge Type	Explanation	Example of Knowledge Type in the Programming Education
Factual Knowledge	- Knowledge of terminology - Knowledge of specific details and elements	- Term itself Related Programming such as Variable, Compiler - Logical or Calculation Operator Type
Conceptual Knowledge	- The relationships among pieces of a larger structure that make them function together	- Understandable Knowledge of each function role that is constituted nested Function - Understandable Knowledge of Queue Structure's Principle - Understandable Knowledge of basic Principle of OOP
Procedural Knowledge	- Knowledge of subject-specific skills and algorithms Knowledge of criteria for determining when to use appropriate procedures	- Critical Knowledge for Specific Algorithm - Insight or Criterion to Select for Data Structure appropriated to a Problem in the Adding up Algorithm
Metacognitive Knowledge	- Knowledge of thinking in general and your thinking in particular	- Trace Method for Variable Value with its own way in the Event Driven Programming - Coding Style, Algorithm using usually with its own unique way to solve specific problem

에 의해 제시된 것이며 메타인지 지식은 Anderson 등이 개정된 블룸 목표 분류를 제시하면서 추가된 지식의 유형이다. 본 연구에서는 지식의 종류에 대한 일반적 설명을 기반으로 <Table 4>와 같이 프로그래밍교육의 지식 유형을 4가지로 구분해 보았다. 그리고 Milla(2016)는 프로그래밍 분야의 지식에 대한 분류도 블룸과 유사하게 제시한 바 있다[9].

4가지 유형별로 프로그래밍교육의 지식을 분류해 본 결과 초중등 프로그래밍교육에서 중점적으로 평가해야 하는 영역은 개념적 지식과 절차적 지식이라고 판단되며 그중에서도 문제해결과 관련된 지식은 절차적 지식이다. 사실적 지식은 프로그래밍 분야의 가장 기본적인 지식이라고 할 수는 있지만, 문제해결력 신장이라는 초중등 프로그래밍교육 목표에 직접적으로는 거리가 있는 지식 유형이라고 할 수 있다.

또한, 메타인지적 지식은 학습자 자신의 독특한 프로그래밍 전략, 암묵적 지식 등을 의미하고 있어서 초보 프로그래머에게 난이도가 높은 수준이라고 할 수 있다. 그리고 메타인지 교육내용으로서의 삽입과 정규교육과정에서의 평가는 공정성이 확보되어야 하는데 이 부분에서도 기준 설정이 모호한 점이 있다.

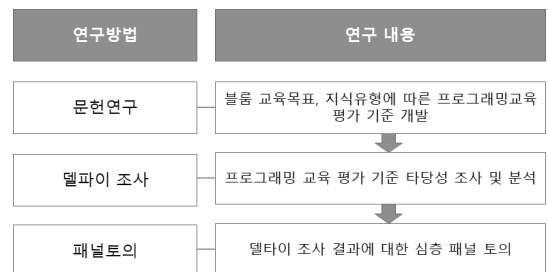
목표 분류에 의한 평가 기준은 학습자의 행동 수준의 지표를 나타내지만 지식 유형 분류는 평가 대상의 지식이 어떤 종류의 지식인지를 참고할 수 있는 지표로 활용 가능하다. 즉 지식 유형은 평가하고자 하는 대상이

어떤 종류의 지식인지를 검토할 수 있는 지표라고 할 수 있다.

4. 연구 방법과 결과

4.1 연구 절차

문헌 고찰과 선행연구 분석 내용을 바탕으로 조사 연구는 2단계에 걸쳐 실시하였다. 먼저 선행연구와 사례 분석에 기초하여 연구자가 개발한 프로그래밍교육 평가 기준의 초안에 대해서 다년간 프로그래밍 교육 경험이 있는 현직 소프트웨어 교육 전문가를 대상으로 1차 검토를 하였으며 설문결과에 대한 패널토의를 4명이 진행하였다. 이를 요약하면 다음 (Fig. 2)와 같다.



(Fig. 2) Process of This Research

4.2 자료 수집 및 분석

문헌 연구를 바탕으로 개발된 블룸 기반 프로그래밍 교육 평가 기준에 대하여 전문가 검토를 위한 설문지를 개발하였다. 설문지는 블룸의 교육목표 분류와 지식 분류의 유형으로 나누어서 5단계 리커트 척도로 타당한 정도를 표시하도록 구성하였다.

델파이 조사지는 블룸의 교육목표 분류 6개 단계, 지식 유형 분류 4개 총 10개 문항에 대해서 5단계 리커트 척도에 따른 타당성의 정도를 표시하는 것이었다.

델파이 조사 결과는 5점 척도의 평균과 내용 타당도 비율(Content Validity Ratio: CVR), 합의도를 활용하여 분석하였다. 평균은 전문가 패널의 응답에 대한 평균으로서 5점 척도의 중앙값인 3을 기준으로 어느 정도에 위치하고 있는지를 판단하였다. 내용 타당도는 Lawshe(1975)가 제시한 내용타당도 비율을 바탕으로 분석하며 CVR은 델파이 조사에 참여한 패널의 수에 따라 최솟값이 결정된다[6].

이 연구에서는 평균 4.0 이상, CVR 0.47 이상, 합의도 0.75 이상을 타당성 판단의 기준으로 설정하였다.

4.3 블룸 목표 분류 기반 평가 기준 타당성 조사 결과

<Table 2>의 내용을 설문으로 구성하여 델파이 조사한 결과를 <Table 5>와 같이 통계 결과를 요약하여 제시하였다.

<Table 5> The Delphi Result of Evaluation Criterion of Programming Education Based on Bloom Taxonomy

Bloom Taxonomy	Average	CVR	Validity
Remember	4.65	0.9	0.8
Understand	4.65	0.8	0.95
Apply	4.65	0.9	0.8
Analysis	4.65	0.9	0.8
Evaluate	4.55	0.9	0.8
Create	4.75	1	0.95
Average	4.65	0.90	0.85

<Table 5>에 의하면 전체적으로 타당성 있게 조사 결과가 나타났다. 특히 “제작하다” 단계의 신뢰도가 거의 “매우 타당하다”로 응답하였다. 이는 블룸의 제작하

다의 기준에 프로그래밍의 제작과 가장 적합한 것으로 판단한 것이라 할 수 있다. 이해하다의 신뢰도는 0.8로서 비교적 높은 편이지만 상대적으로 가장 낮게 나타났다. 이것은 추가적인 패널 토의에서 확인해 본 결과 “이해하다”라는 블룸 분류가 다소 애매하며 이것을 프로그래밍 요소로 표현할 때 자료 추적과 읽기 활동으로 표현되는 점에 대해서 다소 의문을 갖고 있는 전문가들이 있는 것으로 조사되었다.

4.4 블룸 지식 유형 기반 교육 내용 분류의 타당성 조사 결과

블룸의 지식 유형 기준은 <Table 3>의 내용을 설문화한 것으로서 <Table 6>과 같이 요약할 수 있다.

<Table 6> The Delphi Result of Evaluation Criterion of Programming Education Based on Type of Bloom Knowledges

Bloom Knowledge	Average	CVR	Validity
Factual	4.70	1	0.8
Conceptual	4.55	0.8	0.8
Procedural	4.40	0.9	0.75
Metacognitive	4.50	0.9	0.8
Average	4.54	0.90	0.79

전체적으로 타당도(CVR)와 합의도(Validity)가 기준치 이상을 상회하고 있어서 블룸의 지식 분류에 의한 프로그래밍교육 분야의 지식 분류에 대하여 타당성이 있는 것으로 조사되었다.

특히 사실적 지식 즉 용어에 대한 응답자의 신뢰도는 확고한 것으로 나타났다. 그것은 사실적 지식에 대한 설명이 상대적으로 명확하기 때문에 혼란이 적었던 것으로 패널토의를 통해 나타났다.

그럼에도 메타인지적 지식의 신뢰도가 0.7로 나타나서 최저치를 기록했으며 자신만이 가지고 있는 독특한 전략적 지식에 대한 타당성에 대해서 소프트웨어 현장 교육 전문가들은 상대적으로 낮은 수준의 신뢰도를 표현하였다.

하지만 실제 프로그래밍 과정에서 유용한 지식은 절차적 지식이 가장 유용한 것으로 판단되며 패널토의에

서도 초보 프로그래밍과정에서 가장 현실성 있는 지식 유형으로 제시되었다.

5. 결론

본 연구는 블룸 기반 프로그래밍교육 평가 기준에 대한 것이다. 블룸이 교육평가 기준 이론에 대한 진부는 아니다. 블룸 이외에도 SOLO(Structure of the Observed Learning Outcome) 모형 등 몇 개의 평가 기준에 대한 연구가 있다.

하지만 블룸은 교육평가 영역에서 가장 선도적인 모델을 제시했으며 해외의 초보자 프로그래밍평가 연구에서 다양하게 활용되고 있다. 그럼에도 국내에서 초중등 뿐만 아니라 대학 수준에서도 초보자 평가 기준에 대한 연구는 매우 미흡한 상태이다. 이에 본 연구에서는 블룸의 개정된 평가 기준을 이용하여 평가 기준을 제시하고 블룸의 지식 유형에 따라서 프로그래밍 지식을 유형별로 분류하였다.

프로그래밍교육에 대한 블룸의 평가 기준에 의하면 프로그래밍 용어에 대한 암기는 가장 낮은 수준 단계에 해당하며 이미 알려진 수준의 원리를 이용하는 간단한 프로그래밍 제작은 적용 수준에 해당하는 것으로 제시하였다. 또한 프로그래밍 도구를 이용하여 복잡한 문제를 해결하는 것은 제작으로서 가장 난이도가 높은 수준으로 제시하였다. 그리고 블룸의 지식유형 기준으로 개념적 지식과 절차적 지식이 가장 중요한 것으로 제시하였다. 그리고 이와 같은 연구결과의 타당성에 대한 델파이 조사를 실시하였다.

블룸의 6가지 교육목표 분류와 4가지 지식 유형에 따른 프로그래밍교육 평가 기준 초안을 대상으로 조사연구를 통해 설정하였다. 그리고 소프트웨어교육 전문가 20명에게 델파이 조사를 실시하였다. 조사 결과는 신뢰도 값이 0.9로 나타났고 합의도도 기준치로 초과하고 있어 타당성이 있는 것으로 조사되었다.

그리고 진술하고 있는 바와 같이 소프트웨어교육의 목적이 문제해결과 컴퓨팅사고력에 초점을 두고 있다. 이에 블룸 기반 평가 기준에 의하면 평가의 초점은 문제해결을 위한 “제작하다” 수준의 활동에 두어야 한다. 그럼에도 초보자를 위한 과정이기 때문에 블룸의 이해, 적용 수준의 평가 문항을 간과할 수는 없다.

또한 본 연구는 블룸 이론을 기초로 하여 프로그래밍 분야에 한정하여 평가에 대한 이론적 기반에 대한 연구이다. 이에 정보과 교육의 다양한 분야에 대해서도 평가 기준에 대한 지속적인 연구가 요구된다.

참고문헌

- [1] Anderson, Lorin W., and Krathwohl, David R.,(2001). A Taxonomy for Learning, Teaching, and Assessment: A Revision of Bloom's Taxonomy of Educational Objectives. New York: Longman.
- [2] Bloom, B. S. et. al.(1956). Taxonomy of Educational Objectives : Handbook I , Cognitive Domain. New York : David Mckay.
- [3] Cynthia(2015). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. The 10th Workshop in Primary and Secondary Computing Education. <http://dl.acm.org/citation.cfm?id=2818314>.
- [4] Cynthia Collins Selby(2014). How Can the Teaching of Programming Be Used to Enhance Computational Thinking Skills? <http://eprints.soton.ac.uk>
- [5] Errol Thompson et al.(2008). Bloom's Taxonomy for CS Assessment. Tenth Australasian Computing Education Conference (ACE 2008), Australia, p. 158.
- [6] Fred A. Materson (1984). Evaluating programming language for use in education. Computers and Education.
- [7] Judy Sheard et al.(2013). How difficult are exams? A framework for assessing the complexity of introductory programming exams. 15th Australasian Computing Education Conference (ACE 2013). pp 145-154.
- [8] KyungHun, et al.(2015). Prototype Development of Informatics Curriculum. KICE.
- [9] Mila Kwiatkowska(2016). Measuring the Difficulty of Test Items in Computing Science Education. WCCCE '16, May 06-07, 2016, Kamloops, BC, Canada.

- [10] MOE(2015). Informatics Curriculum. Vol 2015-74[Separate Volume 10].
- [11] MOE(2015). Promotion Plan to Train Competent Learner for SW Oriented Society. <http://www.moe.go.kr>
- [12] Otto (2012). Advances in Assessment of Programming Skills. Aalto University Publication Series. p. 34.
- [13] Shuhaida Shuhidan et al.(2009). A Taxonomic Study of Novice Programming Summative Assessment. Eleventh Australasian Computing Education Conference (ACE2009), Wellington, New Zealand.
- [14] Terry Scott(2003). BLOOM'S TAXONOMY APPLIED TO TESTING IN COMPUTER SCIENCE CLASSES. Journal of Computing Sciences in Colleges archive Volume 19 Issue 1, October 2003. Pages 267-274.
- [15] Wiggins, G. P., & McTighe, J. (2005). Understanding by design(2nd ed.). Alexandria, VA : Association for Supervision and Curriculum Development.

저자소개

신 수 범



1991 인천교육대학교 (교육학학사)
1998 한국교원대학교 (교육학석사)
2002 한국교원대학교 (교육학박사)
2002~2005 KERIS 연구원
2005~현재 공주교육대학교 컴퓨터교육과 교수
관심분야: 컴퓨터교육
e-mail: ssb@gjue.ac.kr