

소프트웨어 교육에서 과정중심 평가를 위한 디버깅 과제 및 도구 개발

Development of Debugging Tasks and Tool for Process-centered Assessment on Software Education

류기곤[†] · 김수환^{††}

Kigon Lyu[†] · Soohwan Kim^{††}

요 약

본 연구의 목적은 소프트웨어(SW) 교육에서 과정중심 평가를 지원하기 위한 디버깅 과제를 개발하고 서비스하는 사이트를 개발하는 것이다. 초등학교와 중학교 SW교육과정 중 프로그래밍 영역의 성취기준을 살펴보면 반복, 조건과 같은 제어구조를 구현하는 것과 변수, 입출력과 같은 요소로 구성되어 있다. 교육부에서 제시하는 과정중심 평가를 적용하기 위해서는 실제 수업환경에서 활용할 수 있는 과제를 제시하고, 성취수준과의 연계를 고려해야 한다. 따라서 본 연구에서는 디버깅 과제로 초등학교용 12과제, 중학교용 15과제를 개발하고 그 타당도를 검증하였다. 타당도 검증을 위해서 델파이 검증을 실시한 결과, 디버깅 과제의 수준과 내용이 적합한 것으로 나타났다. 또한, 디버깅 챌린지 사이트의 사용성을 휴리스틱 평가법으로 검증한 결과 양호한 수준으로 나타나, 실제 교육 현장에서 활용하기에 적합한 것으로 나타났다.

주제어: 과정중심 평가, 디버깅 과제, 소프트웨어 교육, 수행평가

ABSTRACT

The purpose of this study is to develop a debugging task to support process-centered assessment on software(SW) education and to develop a service site. Achievement criteria for programming in elementary and middle school SW curriculum consists of control structures such as repetition and condition, and elements such as variables and input/output. In order to apply the process-centered assessment presented by the Ministry of Education, it is necessary to present tasks that can be used in the actual classroom environment and to consider connection with achievement levels. Therefore, in this study, 12 tasks for elementary school and 15 tasks for middle school were developed as debugging tasks and their validity was verified. As a result of conducting Delphi verification for validity, it was found that the level and content of the debugging task are appropriate. In addition, as a result of verifying the usability of the debugging challenge site by applying a heuristic evaluation method, it was found that it is suitable for use in the real educational field.

Keywords: Process-centered Assessment, Debugging Task, Software Education, Performance evaluation

1. 서론

초중등 소프트웨어(SW)교육의 목표는 컴퓨팅 사고력(Computational thinking)을 증진하는 것을 목표로 하고 있다[1]. 컴퓨팅 사고력은 문제나 현상을 컴퓨팅의 관점에서 바라보고 추상화 자동화를 거쳐 컴퓨팅 파워를 통해 문제를 해결하는 능력이다[2][3]. 컴퓨팅 사고력을 증

진하기 위해서는 학생들이 알고리즘 및 프로그래밍 활동을 통해 문제를 분해하고 추상화하고 자동화하는 경험을 통해 체득하는 것이 효과적이다[3][4]. 특히, 2015개정 교육과정은 학생들의 실제적인 역량을 강화하는 ‘역량중심 교육과정’으로 설계 되었으며, 여러 가지 교수학습 방법과 함께 ‘과정중심 평가’를 통한 학생의 성장을 돕도록 개발되었다[5].

[†]정 회 원: 고려대학교 연구교수

^{††}충신회원: 충신대학교 조교수(교신저자)

논문접수: 2020년 04월 07일, 심사완료: 2020년 06월 09일, 게재확정: 2020년 07월 01일

※ 이 논문은 2019년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임(NRF-2019S1A5A8034424)

2015개정 교육과정에서 제시된 과정중심 평가는 교육과정의 성취기준에 기반한 평가 계획에 따라 교수·학습 과정에서 학생의 변화와 성장에 대한 자료를 다각도로 수집하여 적절한 피드백을 제공하는 평가라고 정의하고 있다. 또한 과정중심 평가의 목적은 학생의 교육목표 도달도를 확인하고 교수·학습의 질을 개선하여 학생들의 학습과 성장을 지원하는 것이다. 정보 교과에서 적용할 수 있는 평가방법은 프로젝트, 포트폴리오, 자가·동료평가, 관찰법, 디버깅 과제 해결 등이 있다[6].

디버깅 과제의 경우 프로그래밍 수업에서 적용할 수 있는 평가 방법으로 국내외 소프트웨어 교육에서 활용하는 방법이다[7][8]. 디버깅 과제는 프로그래밍의 요소를 학습하고 학생들의 형성평가나 과정중심 평가의 방법으로 적용할 수 있다. 초·중등 소프트웨어 교육의 성취수준에는 프로그래밍 수업에서 도달해야 할 목표를 명시하고 있어서 디버깅 과제로 학생들의 성취도를 확인할 수 있다. 따라서 본 연구에서는 초등학교, 중학교 소프트웨어 교육에서 적용할 수 있는 디버깅 과제를 개발하고 제시된 과제를 학생들의 과정중심 평가에 활용할 수 있는 웹사이트를 개발하였다. 개발한 웹 사이트에는 학생들의 디버깅 결과물을 판별할 수 있는 평가 모듈을 개발하고 적용하였다.

본 연구에서 개발한 디버깅 과제의 타당도를 검증하기 위해 전문가 그룹의 델파이 조사를 실시하였으며, 디버깅 과제 웹 사이트의 사용성을 휴리스틱 평가법을 적용하여 검증하였다.

2. 이론적 배경 및 선행연구 분석

2.1 과정중심 평가

2015 개정 교육과정에 따르면, ‘과정중심 평가는 성취기준에 근거하여 교수·학습과 평가의 일관성을 유지하여 배운 내용을 평가하되, 학습결과에 대한 평가뿐만 아니라 학습과정 상의 평가를 중요하게 포함하여 학생의 자기성찰과 성장을 지원하고자 하는 평가라는 특징을 가진다’고 제시되어 있다[5][6].

이는 기존의 형성평가나 수행평가, 실제평가와 비슷한 개념이지만 교수·학습과정에서 교사의 교육과정 재구성 및 평가 계획에 따라 진단, 형성, 총괄평가를 모두 포함할 수 있고, 현행 지침상의 지필·수행평가 모두를 포괄하는 개념으로 나타난다[5][7].

과정중심 평가는 학생들의 성장을 도모하는 것으로 테일러(Tyler)와 브루너(Bruner)의 지식의 구조를 기반으로

평가하고 가르치는 데에 주안점을 둔다[8]. 따라서 학생들의 진정한 이해에 대한 증거를 수집하여 평가하는 것이 중요하다. 소프트웨어 수업의 알고리즘과 프로그래밍 단원은 개념을 실제 프로그래밍을 통해서 체득하도록 되어 있다. 따라서 평가도 학생들의 학습환경과 비슷한 환경에서 실제 수업 내용을 반영한 평가문제로 개발하여 제시하는 것이 필요하다. 즉, 평가문제를 문제해결 상황이 포함된 디버깅 과제로 제시하면 학습과 평가가 수업에서 바로 이루어질 수 있다.

2.2 디버깅 과제

2015개정교육과정 정보과 직무연수에 의하면 서술, 논술, 구술, 토의·토론, 프로젝트, 실험·실습, 포트폴리오, 자기평가·동료평가, 관찰법 등이 있다[6].

소프트웨어 교육에서 디버깅 과제의 경우 학생들의 산출물로 개념 평가, 프로그래밍 능력을 평가하여 학생들의 컴퓨팅 사고력 증진을 도모할 수 있다. 미국 스크래치 팀의 평가방법[13]이나 SRI나 AP 코스에서도 평가과제로 디버깅 과제를 제시하고 있다[14][15]. 과정중심 평가에서 추구하는 교수·학습 흐름은 평가 과제를 학습활동으로 제시하고 실제 평가 과제를 해결하는 가운데 학습과 평가가 순환적으로 이루어지는 방향을 추구하고 있다[6]. 따라서 본 연구의 디버깅 과제는 학생들이 오류를 찾고 수정하는 과정에서 학습 목표 달성과 컴퓨팅 사고력 증진을 도모하는 방법으로 적용할 수 있다.

초·중등 교육현장에서 디버깅 과제로 과정중심 평가를 실시할 수 있는 세부적인 방법은 <표 1>과 같이 정리할 수 있다[9][10].

표 1. 디버깅 과제 적용 평가방법

평가 요소	세부 내용
개념 및 지식	변수, 연산자, 반복과 같은 CS개념 → 수도 코드, 디버깅 과제 가능
프로그래밍 능력	학생들의 코딩 산출물을 Drscratch와 같은 평가도구로 분석 및 평가 → 디버깅 과제 가능
프로그래밍 응용	프로젝트 네임, 퍼포먼스 정하기 → 예제 시나리오 만들거나 수정 보완 → 디버깅 과제 가능

미국의 스크래치 교육팀에서는 디자인 시나리오 방식에 따라 24개의 스크래치 디버깅 문제를 개발하여 제시하였다. 각 디버깅 문제는 순차, 반복, 선택구조와 같은 내용을 기반으로 하고 있으나, 우리나라 교육과정과 매

칭되지 않는다[11][12][13]. 미국 collage board에서도 AP 코스를 위한 평가 문항을 개발하였는데, 비버문항이나 정보올림피아드 문항과 비슷한 형태로 개발되어있다[14]. SRI 연구팀에서는 프로그래밍 과정에서 증거기반 설계 기법을 적용한 과제 예시를 개발하여 제공하고 있으나, 대표과제 예시 몇 개만 개발하여 제시하고 있는 실정이다[15].

이상에서 살펴본 국내외 사례에서는 2015개정교육과정의 성취수준과 연계되어 있지 않으며 국내에서 가장 많이 사용하는 엔트리에 기반한 평가도구로 개발되어 있지 않다. 따라서 본 연구에서는 소프트웨어 교육과정에서 제시한 성취수준에 근거한 평가요소를 반영하여 디버깅 과제를 개발하였다.

2.3 프로그래밍 산출물 평가 도구

온라인 프로그래밍 평가 방법은 전통적으로 알고리즘 저지 사이트들로 KOI Study, Coding is fun, Dovelet, Code Up 등 있다[16]. 대부분 이런 사이트들은 주로 영재교육이나 산업교육 등 수월성 교육에서 사용되고 있으며, 텍스트 언어용으로 개발되어 있다.

블록형 도구의 코드 평가 시스템에는 [그림 1]과 같이 ‘엔트리’의 평가 도구 개발 사례가 있다[17].



그림 1. 엔트리 코드 분석 시스템(선행연구)

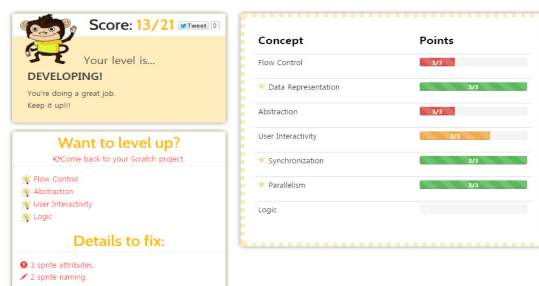


그림 2. Drscratch 분석 화면[20]

또한, 스크래치 산출물 평가 시스템은 [그림 2]와 같이 컴퓨팅 사고력을 학습자의 수준에 따라 결과를 보여주는 도구가 개발되어 있다[18][19].

이상에서 살펴본 프로그래밍 산출물 평가 도구는 특정 평가물에 의해서 컴퓨팅 사고력을 평가해주는 도구로 개발되었거나 텍스트 언어용으로 개발된 것으로, 수업환경에서 과정중심 평가를 지원하는 평가 도구로 개발되지는 않았다.

3. 연구방법 및 절차

본 연구에서는 우리나라 초등학교, 중학교 교육현장에서 많이 활용되는 엔트리에 적용할 수 있는 컴퓨팅 사고력 평가물과 수행형 디버깅 과제를 개발하였다. 디버깅 과제의 경우 단순한 문제풀이가 아닌 수행형 과제로 만들어 평가할 수 있도록 증거기반설계(ECD) 기법[21]을 활용하여 디버깅 과제를 개발하였다.

개발된 디버깅 과제의 타당성과 적합성을 검증하기 위해 초중등 소프트웨어 교육 전문가 10인을 대상으로 2회에 걸쳐 델파이 검증을 실시하였다. 전문가는 초등학교 교사 4명, 중학교 교사 3명, 컴퓨터교육 박사 2명이 참여하였다.

다음 단계로 개발된 디버깅 과제를 실제 수업에서 활용할 수 있도록 디버깅 챌린지 사이트를 개발하였다. 디버깅 사이트는 선행 연구를 통해 개발된 평가 모듈[19]을 개선하여 학생들이 수정하고 업로드 한 엔트리 소스를 사전에 개발된 정답 소스와 비교하여 판별할 수 있도록 개발하였다.

디버깅 챌린지 사이트는 교육현장 사용성을 테스트하기 위해 초등학교 교사 4명, 중학교 교사 4명을 대상으로 휴리스틱 평가를 실시하였다. 닐슨의 연구에 의하면 해당 분야의 전문가 5-6명의 피실험자를 통해 시스템의 85% 정도의 발견을 얻을 수 있다고 하다[22]. 즉, 적은 인원으로 효과적으로 사용성을 평가할 수 있는 방법으로 효율적이고 빠르게 수행할 수 있다는 장점이 있다. 본 연구는 초중등 교육현장에서의 사용성이 중요하므로, 교사를 전문가로 초빙하여 실제 수업환경에서의 사용성과 적합성을 검증하기 위한 방법으로 적용하였다. 닐슨이 제시한 휴리스틱 평가요소는 현재상태 시각화, 실세계 부합, 적절한 통제권, 일관성과 표준성, 실수방지, 적은 인지노력, 유연한 사용, 심미적이고 간결, 오류파악 수정, 충분한 도움말이다.

4. 과정중심 평가를 위한 디버깅 과제 개발

본 연구의 목적은 초등학교, 중학교 소프트웨어 교육에서 활용할 수 있는 과제중심 평가가 가능한 디버깅 과제를 개발하는 것이므로, 소프트웨어 교육과정에 제시된 성취수준을 평가할 수 있는 과제로 개발하였다. 성취수준의 평가요소를 도출하기 위해 증거기반설계(ECD) 기법을 적용하였으며 <표 2>와 같이 평가요소를 도출하였다.

<표 3>과 같이 초등학교, 중학교의 성취수준에 제시된 요소들을 평가할 수 있는 수행형 평가과제를 초등학교 12개 과제, 중학교 15개 과제를 개발하였다.

표 2. 성취기준에 따른 평가요소

구분	평가 요소	디버깅 과제
초등	• 순차 구조 • 반복 구조 • 조건 구조 • 종합문제	요소별 기초 2과제 중급 1과제 → 총 12과제
중등	• 순차, 반복 구조 • 순차, 조건 구조 • 변수와 연산 • 입출력 • 종합문제	요소별 기초 2과제 중급 1과제 → 총 15과제

도출된 평가요소를 반영한 디버깅 과제를 <표 3>과 같이 개발하였다.

표 3. 초등학교, 중학교 디버깅 과제

분류	디버깅 과제
초등학교	순차
	초급1
	초급2
	중급
	반복
	초급1
	초급2
	중급
	조건
	초급1
	초급2
	중급
	종합
	초급1
	초급2
	중급

분류	디버깅 과제
중학교	순차 / 반복
	초급1
	초급2
	중급
	순차, 조건
	초급1
	초급2
	중급
	변수와 연산
	초급1
	초급2
	중급
	입출력
	초급1
	초급2
	중급
	다양한 응용
	초급1
	초급2
	중급

개발된 디버깅 과제에 따라서 개발된 디버깅 챌린지 사이트는 [그림 3]과 같다. 학생들은 평가 요소별로 구분된 과제와 단계를 선택하여 도전할 수 있다.

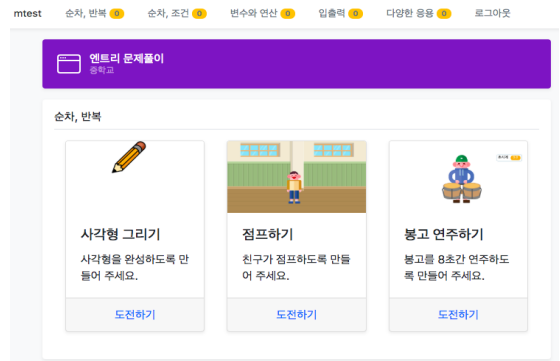


그림 3. 디버깅 챌린지 인터페이스

도전 과제를 선택하고 문제를 확인한 후 소스보기를 누르면 엔트리 사이트로 접속되고 디버깅 문제가 제시된다. 이때 디버깅 과제를 해결한 후, 소스를 다운받아 챌린지 사이트에 올리고 정답확인을 누르면 [그림 4]와 같이 내부 판별 모듈에서 정답을 체크하여 [그림 5]와 같이 피드백을 준다. 학생들은 자신이 어느 과제를 해결했는지 확인할 수 있다.

조건 1단계



그림 4. 디버깅 과제 판별 인터페이스

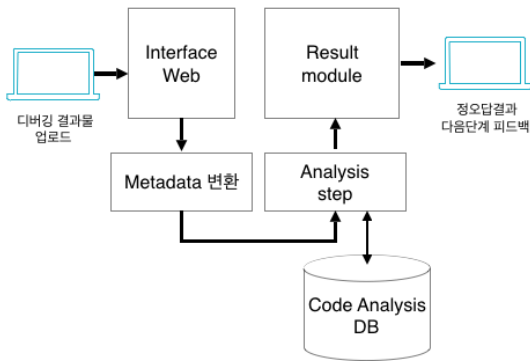


그림 5. 디버깅 챌린지 사이트 구조도

5. 연구 결과

5.1 델파이 전문가 검증

개발된 디버깅 과제에 대한 1차 전문가 타당도 검증 결과는 <표 4>와 같다. 평가요소와 수준에 대한 적합도를 매우적합부터 매우부적합의 5점 척도로 평가하였다. 델파이 기법에 따르면 전문가 10명 기준으로 CVR값이 0.64이상이 되어야 하므로 0.6보다 현저하게 낮은 문항들은 총 9문항으로 나타났다. 현저하게 낮은 문항의 특징은 문제가 모호하거나 평가요소가 제대로 반영되지 않은 의견이 많았다. 또한 초급, 중급, 고급의 구분이 모호하여 평가 점수가 낮게 나타난 것으로 분석되었다. 초등 과제에서는 반복에 관련된 평가 문항이 모두 적합하지 않은 것으로 나타났으며, 중학교 과제에서는 반복, 입출력, 응용 과제에서 낮은 적합도를 보였다.

표 4. 디버깅 과제 1차 델파이 결과

구분	내용	단계	평균 (5점척도)	CVR
초등문제	순차	초급	4.2	0.80
		중급	3.6	0.20
		고급	4	0.40
	반복	초급	3.1	-0.40
		중급	3.7	0.00
		고급	3.9	0.00
	조건	초급	3.6	0.00
		중급	4	0.40
		고급	3.7	0.40
	종합	초급	4.3	0.40
		중급	3.8	0.20
		고급	4.1	0.40
중등문제	순차, 반복	초급	4.6	1.00
		중급	4.4	0.80
		고급	3.4	-0.40
	순차, 조건	초급	4.5	1.00
		중급	4.7	1.00
		고급	4.4	0.60
	변수연산	초급	4.3	0.80
		중급	4.2	0.60
		고급	4.6	1.00
	입출력	초급	3.9	0.40
		중급	3.8	0.20
		고급	3.4	-0.20
	응용	초급	4.1	0.60
		중급	4	0.20
		고급	4.5	0.60

1차 델파이 결과에 따라 적합도가 낮게 나온 문항은 새로 개발하거나 수정하였고, 수준도 초급과 중급으로만 구분하였다. 2차 델파이 결과는 <표 5>와 같이 나타나 대부분의 과제가 적합한 것으로 평가되었다. 초등 과제에서는 반복에서 1개 문항이 부적합한 것으로 나타나 최종 수정하였다. 중학교 과제에서는 응용 중급 문항이 CVR 0.6으로 나타났으나, 기준에 근접한 적합도이므로 과제의 설명만 명확하게 수정하여 개발을 완료하였다.

표 5. 디버깅 과제 2차 델파이 결과

구분	내용	단계	평균 (5점척도)	CVR
초등문제	순차	초급1	4.6	1.00
		초급2	4.4	0.80
		중급	4.4	0.80
	반복	초급1	3.3	-0.40
		초급2	4.7	1.00
		중급	4.8	0.80
	조건	초급1	4.7	1.00
		초급2	4.6	1.00
		중급	4.4	1.00
	종합	초급1	4.8	1.00
		초급2	4.9	1.00
		중급	4.6	0.80

구분	내용	단계	평균 (5점척도)	CVR
중등문제	순차, 반복	초급1	4.8	1.00
		초급2	4.7	1.00
		중급	4.8	1.00
	순차, 조건	초급1	4.8	1.00
		초급2	4.9	1.00
		중급	4.8	1.00
	변수연산	초급1	4.5	1.00
		초급2	4.8	1.00
		중급	4.9	1.00
	입출력	초급1	4.4	0.80
		초급2	4.6	1.00
		중급	4.5	0.80
	응용	초급1	4.7	1.00
		초급2	4.7	1.00
		중급	4.4	0.60

5.2 휴리스틱 사용성 검증

디버깅 챌린지 사이트에 대한 사용성 검증을 위해서는 휴리스틱 사용성 평가 방법을 적용하였다. 휴리스틱 평가 문항 22중에서 본 연구와 연관이 적은 문항을 3개 삭제

제하고 적용 가능성은 초중등 교육현장 적용 가능성으로 수정하였으며, 만족도 문항도 추가하여 조사하였다. 전문가 8명의 평가 결과는 <표 6>과 같다.

사용성과 현장 적합성에 대한 부분은 전반적으로 양호한 것으로 나타났다. 전체 문항 중에서 사소한 문제는 11건, 심각한 문제 4건으로 나타났다. 사소한 문제의 대표 문제는 문제의 친절한 기술과 풀이방법에 대한 안내의 부족이었고, 심각한 문제는 학생들이 코드를 업로드한 상황을 알 수 있도록 해주어야 한다는 의견이 있었다. 4점 척도 사용성의 전체 평균은 3.7로 나타나 매우 양호한 것으로 나타났다.

사소한 문제의 해결을 위해 과제 해결 방법에 대한 매뉴얼을 제작하였고, 심각한 문제를 해결하기 위해 코드를 업로드하면 체크 아이콘이 나타나도록 수정하였다.

6. 결론 및 제언

본 연구에서는 초등학교, 중학교의 소프트웨어 교육에서 과정중심 평가 과제로 활용할 수 있는 디버깅 과제를

표 6. 휴리스틱 평가 결과

구분	번호	항목	평균 (4점 척도)	사소한 문제 (N)	심각한 문제 (N)
현재상태시각화	1.1	처음 들어왔을 때 앱의 목적에 가장 부합하는 콘텐츠가 우선적으로 보이는가?	4.0		
	1.2	Key를 누를 때마다 제대로 된 피드백이 느껴지는가?	3.9		
	1.3	적절한 텍스트와 적절한 이미지가 아름답게 배치되어 있는가?	3.4	1	1
실세계와부합	2.1	앱 아이콘 디자인이 구체적이고 친숙한가?	3.8	1	
	2.2	앱의 사용자가 원하는 정보를 정확히 제공하고 있는가?	3.3	2	
	2.3	사용시 불필요한 과정을 거치지 않는가?	3.9		1
통제권부여	3.1	각 메뉴마다 인터랙션이 동일하게 제공되는가?	3.9		
	3.2	각 기능을 사용하는 방법을 쉽게 기억할 수 있는가?	4.0		
일관성	4.1	똑같은 기능을 하는 버튼의 이름이 어느 화면에서나 항상 동일한가?	4.0		
	4.2	메뉴의 하위메뉴가 메뉴 상위 메뉴에 부합하도록 배치되어 있는가?	4.0		
에러방지	5.1	모든 메뉴가 논리적이고, 서로 겹치는 의미 없이 상호 구분적인가?	4.0		
	5.2	처음 접속하는 메뉴를 효과적으로 사용할 수 있도록 지시 방향이나 설명이 적절한가?	3.0	1	1
적은노력	6.1	데이터의 흐름이 위쪽-왼편에서부터 시작하는가?	4.0		
	6.2	각 화면마다 필요한 버튼들이 빠짐없이 배열되어 있는가?	3.9		
유연성	7.1	옵션 정보를 편리하게 볼 수 있는가?	4.0		
	7.2	뒤로 또는 앞으로의 이동이 간편하고 쉬운가?	4.0		
심미적간결성	8.1	가독성이 높은 텍스트와 배열을 적절하게 사용하고 있는가?	3.9		
	8.2	화면에 있는 모든 정보가 꼭 필요한 정보들로만 간결하게 나열되어 있는가?	3.4	2	
	8.3	모든 버튼들이 시각적으로나 의미적으로나 확실히 구분되는가?	4.0		
에러시도움	9.1	사용자가 에러를 스스로 해결할 수 있도록, 에러상황 발생시 일정기준의 가이드를 제공하는가?	3.3	1	1
	9.2	에러 메시지의 표현이 간결하고 명확한가?	3.8	1	
적합성	10.1	본 시스템은 수행평가 도구로 사용이 적합한가?	3.4	1	
	10.2	본 시스템은 초중등 교육현장에서 사용하기에 적합한가?	3.3	1	
만족도	11.1	본 시스템은 전반적으로 만족스러운가?	3.4		
	11.2	본 시스템의 기능은 만족스러운가?	3.5		
합계			3.7	11	4

만들고, 이를 서비스하는 디버깅 챌린지 사이트를 개발하였다. 본 연구를 통해 나타난 결과는 다음과 같다.

첫째, 초등학교 12개, 중학교 15개 디버깅 과제는 각 학교급의 교육과정 성취기준에 근거하여 개발하였으며, 개발된 디버깅 과제의 타당도를 검증하기 위해 2차례 델파이 검증을 실시한 결과 초등 반복 1개 문항을 제외하고 모든 문항에서 CVR 0.6 이상으로 나타나 적합한 것으로 나타났다. 이는 초등학교, 중학교 교육현장에서 과정중심 평가 과제로 활용이 가능하다는 것을 보여준다.

둘째, 디버깅 과제를 교실수업에 적용하기 위해 개발한 디버깅 챌린지 사이트는 전문가 휴리스틱 평가 결과 전반적으로 사용성이 우수한 것으로 나타났으며, 실제 교육현장에 적용 가능하다는 의견도 긍정적이었다.

셋째, 개발된 시스템의 사용성 테스트에서 나타난 문제점은 문제풀이에 대한 친절하고 상세한 안내가 필요하고, 학생들이 소스를 업로드 한 상황을 명확하게 인지할 수 있도록 피드백을 주어야 한다는 것으로 나타났다. 따라서 소스 업로드 후 정보 제시 방법의 수정이 필요한 것으로 확인되었다.

따라서 본 연구에서 개발한 디버깅 과제는 초등학교, 중학교에서 과정중심 평가 도구로 사용하기에 적합하며, 디버깅 챌린지 사이트도 안내 및 정보 제공 일부를 수정하면 교실 수업에 적용 가능한 것이 확인되었다.

본 연구에서 개발한 디버깅 챌린지 사이트를 교육현장에 실제 적용하기 위해서 다음과 같은 후속 연구가 필요하다.

첫째, 본 시스템의 디버깅 과제는 각 평가요소에 따라 한정적인 문항만 제공하므로, 후속 개발을 통하여 더 많은 평가문항을 확보할 필요가 있다. 특히, 교사들의 자발적인 참여를 통한 문제은행과 같은 체계를 구축하는 것이 필요하다.

둘째, 디버깅 챌린지 사이트를 과정중심 평가에 직접 활용하려면 교수자의 관리 기능이 필요하다. 본 시스템은 1단계의 산출물로 후속 연구를 통해 LMS 추가와 관리자 기능을 추가 개발할 예정이다.

셋째, 현재 학습자의 작품 판별은 정해진 정답에 의해 체크하도록 되어 있지만, 다양한 정답 코드의 유연성을 반영할 수 있는 정답 코드의 다양화가 필요하다. 특히, 정답 코드의 특정 패턴에 대한 분석을 통한 평가 모듈의 개선이 필요하다.

참고문헌

[1] 교육부(2015). **실과(기술·가정)/정보과 교육과정**. 교육부 고시 제2015-7호 [별책 10].

[2] 교육부 (2017). **초·중등학교 교육과정 총론**. 교육부 고시 제2017-131호 [별책 1].

[3] Denning, P. J. & Tedre, M. (2019). *Computational Thinking*. Cambridge, MA; The MIT Press.

[4] 김수환·김찬웅·김혜영 (역) (2017). **code.org를 활용한 컴퓨팅 사고력과 코딩 교육. Computational Thinking and Coding for Every Student**. 서울: 한빛아카데미.

[5] 교육부·한국교육과정평가원 (2017). **과정을 중시하는 수행평가 어떻게 할까요?: 중등**. 연구자료 ORM 2017-19-2.

[6] 교육부·충청남도교육청·한국과학창의재단(2019). **교사별 과정 중심 평가 강사요원 양성 직무 연구 자료집**. 중학교 정보과.

[7] 반재천·김선·박정·김희경·이해선·김수진·신미경·김한승·유명한·정상명·여인경 (2018). **2015 개정 교육과정에 따른 교사별 과정중심 평가 활성화를 위한 학생 평가 모형 개발 연구**. 교육부 11-1342000-000300-01.

[8] 강현석·이지은 (2016). **이해중심 교육과정을 위한 백워드 설계의 이론과 실천: 교실 혁명**. 서울: 학지사.

[9] Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.

[10] 미래인재연구소 (2015). **창의컴퓨팅 이슈리포트 2015-1**. 인천: 경인교육대학교.

[11] Harvard Graduate School of Education. (2011). **CREATIVE COMPUTING**. 2020. 04. 07. 검색. <http://creativecomputing.gse.harvard.edu/guide/curriculum.html>

[12] Getting Unstuck. 2020. 04. 07. 검색. <https://gettingunstuck.gse.harvard.edu/>

[13] Scratch in Practice. 2020. 4. 7. 검색. <https://sip.scratch.mit.edu/>

[14] CollegeBoard (2016). *AP Computer Science Principles Courses and Exam Description*. 2020. 04. 07. 검색. <https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf>

[15] Snow, E., Tate, C., Rutstein, D., & Bienkowski, M. (2017). *Assessment Design Patterns for Computational Thinking Practices in Exploring Computer Science*. Menlo Park, CA: SRI International.

[16] 김성식 (2016). **프로그래밍(알고리즘) 자동평가 시스템의 국내의 동향 및 활용 방안**. 2016이슈리포트, 한국정보과학연합회, 한국과학창의재단.

[17] 김수환·송상수·임선명·김지현 (2016). **Computational Thinking 평가를 위한 엔트리 자동 평가 시스템 개발. 한국컴퓨터교육학회학술대회 발표집**.

[18] Martin, T., & Fields, D. (2014). *Macro data for micro learning: Developing FUN! for Automated Assessment of Computational Thinking in Scratch*. Poster presented at the 2014 Cyberlearning Summit, Madison, WI.

- [19] 김수환 (2015). Computational Thinking 개념 평가를 위한 스크래치 코드 분석 시스템 개발. **컴퓨터교육학회 논문지**, 18(6). 13-22.
- [20] Drscratch. 2015. 8. 14일 검색.
<http://drscratch.programamos.es/>
- [21] Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). *A brief introduction to evidence-centered. Research report*, RR-03-16. ETS,
- [22] Nielsen J. (1994). "how to conduct a Heuristic Evaluation" USEIT.com, 2020. 04. 07. 검색.
http://www.useit.com/papers/heuristic/heuristic_list.html



김 수 환

1999년 경인교육대학교(교육학 학사)
2006년 경인교육대학교
컴퓨터교육과(교육학석사)
2011년 고려대학교 컴퓨터교육과
(이학박사)
2014년~현재 충신대학교 조교수

관심분야: 컴퓨터교육, AI교육, 컴퓨팅사고력, CT평가
E-Mail: skim@csu.ac.kr



류 기 곤

2006년 백석대학교 정보보호학전공
(공학사)
2008년 한신대학교 컴퓨터학과
(공학석사)
2017년 고려대학교 컴퓨터학과
(공학박사)
2018년~현재 (주) 이어주다 대표

2017년~2020년 고려대학교 정보창의교육연구소 연구교수
관심분야: 컴퓨팅사고력, CT평가, 텍스트마이닝
E-Mail: gon0121@korea.ac.kr