Journal of The Korean Association
of Information Education Vol. 18, No. 1,
March 2014, pp. 57-64

http://dx.doi.org/10.14352/jkaie.2014.18.1.57
© 2014 KAIE

Computational Thinking역량 계발을 위한 수업 설계 및 평가 루브릭 개발

최형신

춘천교육대학교 컴퓨터교육과

요 약

본 연구의 목적은 최근에 큰 주목을 받고 있는 Computational Thinking(CT) 개념과 세부 역량에 대해 문헌 연구를 수행하고 CT역량을 계발하기 위한 초등예비교사를 위한 수업 내용과 평가 방법을 고안하는 것이다. 이를 위해 CT관련 문헌, 프로그래밍 효과성 관련 문헌, 스크래치 교육용 프로그래밍 언어 2.0버전의 기능을 분석하여 예비초등교사를 위한 6개의 CT세부 요소(절차 및 알고리즘, 병행화 및 동기화, 자료 표현, 추상화, 문제 분해, 시뮬레이션)를 도출하였다. 도출된 6개의 CT세부 역량의 계발을 위해 교원양성기관에서 한 학기에 진행할 수 있는 수업을 설계하고 각 차시의 수업내용과 CT요소와의 관련성을 분석하여 제시하였다. 또한 PECT(Progression of Early Computational Thinking)모형과 CT평가 프레임워크를 분석하여 학습자의 산출물에 나타난 CT역량 습득 정도를 기초, 발달, 능숙 세 단계로 평가할 수 있는 루브릭을 고안하였다. 본 연구의 결과는 향후 실증적 효과성 검증을 통해 교원양성기관에서 CT역량 계발 프로그램으로 활용될 수 있을 것으로 기대한다.

키워드 : Computational Thinking, 프로그래밍, 구성주의(Constructionism), 루브릭, 디지털 콘텐츠

Developing Lessons and Rubrics to Promote Computational Thinking

Hyungshin Choi

Dept. of Computer Education, Chuncheon National University of Education

ABSTRACT

This study aims to suggest lesson plans and evaluation methods for primary pre-service teachers by reviewing the concept of computational thinking(CT) skills and its sub components. To pursue this goal, a literature review has been conducted in regards to CT and the effectiveness of programming courses. In addition, the Scratch educational programming functions were analyzed yielding six CT elements(data representation, problem decomposition, abstraction, algorithm & procedures, parallelization, simulation). With these six elements, one semester lesson plans for 15 weeks that represent the connections with six CT elements were designed. Based on the PECT(Progression of Early Computational Thinking) model and the CT framework a rubric to evaluate

논문투고 : 2014-01-28
논문심사 : 2014-02-03
심사완료 : 2014-03-03

learners' proficiency levels(basic, developing, proficient) revealed in their final projects was developed as well. Upon a follow-up empirical study, the lesson plans and the rubric suggested in the current study are expected to be utilized in teachers' colleges.

Keywords: Computational Thinking, Programming, Constructionism, Rubrics, Digital Content

1. 서론

미래부는 '과학기술과 ICT를 통한 창조경제와 국민행복 실현'이라는 비전을 제시하고 추진 전략으로 21세기 언어인 소프트웨어를 핵심 산업으로 육성하고 누구나 소프트웨어를 개발 및 활용할 수 있는 교육 기반을 조성하며 '한국스타일' 콘텐츠로 세계에 진출하는 과제를 제시하고 있다[9]. 또한 2013년 3월에는 미국 주요 IT 기업의 CEO들이 유튜브 동영상을 통해 프로그래밍 조기 교육의 필요성을 강조한바 있다. 한편 북유럽 소국 에스토니아는 초등학교 1학년부터 프로그래밍 교육을 필수화하기로 하였으며[18], 영국은 2013년 7월 개정된 교육과정에 2014학년도부터 시작하여 5세부터 프로그래밍 교육을 필수화하기로 하였다[1].

이러한 일련의 움직임이 시사하는 바는 프로그래밍 교육이 가지고 있는 본질적인 가치에 미국, 영국, 유럽의 국가들, 우리나라도 주목하게 되었다는 것이며, 그 본질적 가치는 바로 프로그래밍을 통한 사고력 향상이다. Papert[14]에 의해 처음 설명된 Computational Thinking(이하 CT)은 2006년 미국 카네기 멜론 대학교의 컴퓨터과학과 교수 Jeanette Wing의 논문을 통해 학계에 크게 부각되었다. Wing은 CT가 단순히 컴퓨터 과학자들만이 아니라 모든 사람들이 배우고 사용해야 하는 보편적인 태도와 기술을 제시하고 있다고 주장하였다. 따라서 읽기, 쓰기, 셈하기에 이어 모든 아동의 분석적 능력으로 CT를 추가해야 한다고 하였다[19][20].

미국을 비롯한 해외 여러 나라에서 CT에 대한 지원이 증가되고 있다. 미국에서는 국가 차원의 K-12학생, 교사, 대학교 1, 2학년의 CT능력 발전을 위해 노력하고 있으며 그 예로서 국립과학재단에서는 2010년 '21세기를 위한 컴퓨팅교육(CE21)' 프로젝트 착수를 들 수 있다. 또한 영국은 국립과학재단(Royal Society)에서 CT 교육 방법을 모색하기 위한 프로젝트에 착수하였다 [8].

미래 직업은 더욱 복잡한 문제를 효율적이고 창조적으로 해결하는 능력을 요구한다. 이를 대비하기 위해 CT역량을 제고시키는 교육의 필요성에 대한 인식이 우리사회 전반에서 필요하다.

본 연구에서는 먼저 국내·외 CT관련 문헌 연구를 통해 초등예비교사가 갖추어야 할 CT skills의 세부 역량은 무엇인지 도출하고 이러한 세부 역량을 프로그래밍 수업에 도입하기 위한 교수 학습 설계 및 평가 방안을 모색하고자 하였다.

2. Computational Thinking계발관련 선행연구

2.1 국내 선행 연구

최근 5년간 국내에서 수행된 스크래치 프로그래밍의 효과성 연구들을 살펴보면 주로 학습자 동기, 사고력, 문제해결력 및 자기주도성 등에 초점을 맞추고 있다. 송정범, 조성환, 이태욱[17]은 초등학교 6학년 67명을 대상으로 재량활동 시간을 통한 스크래치 프로그래밍 학습이 학생들의 내재적 동기와 문제해결력 향상에 효과가 있다는 결과를 보고하였다. 이 연구에서 학습 동기는 작업 선호도 검사로 측정하였으며 문제해결력은 PISA의 문제해결력 검사를 활용하였다. 문제해결력 검사는 2003년 OECD가 실시한 학업성취도 PISA의 문제해결 영역의 공개문항 중 문제 해결의 유형, 문제 상황, 학문 영역, 문제해결과정, 추론 기술 영역에서 12문항을 무작위 추출하여 사용하였다. 문제의 유형에는 의사 결정, 시스템분석과 설계, 문제점 해결이 포함되었다.

또한 조성환 외는 중학교 1-2학년 30명을 대상으로 12주간 방과후 특기적성 프로그램으로 진행한 스크래치 게임제작 프로그래밍 수업이 메타인지에 긍정적

효과가 있음을 밝혔다[5]. 이 연구에서 메타인지를 측정하기 위해 MSLQ를 번역 수정한 질문지를 활용하였다. 측정 내용에는 연습, 정교화, 조직화, 비판적 사고, 자기 규제 요인이 포함되었다. 이외에도 다른 연구자들은 스크래치 프로그래밍이 체제적 사고, 창의적 사고력, 자기 주도성의 향상에 긍정적인 영향이 있음을 보고한 바 있다[2][6].

위에서 언급한 연구들은 스크래치 프로그래밍의 효과성을 검증함에 있어 구체적 CT 구성요소와의 직접적인 연결은 제시하지 않고 있다. 최근에 와서 CT에 대한 관심이 고조되면서 CT 개념 및 구성요인을 규명하는 이론적 연구들이 수행되었으며[7] 프로그래밍을 통한 CT역량 계발 효과성을 검증하는 실증적 연구들이 수행되기 시작했다. 김수환과 한선관[10]은 초등학교 4학년 영재학급 학생들 20명을 대상으로 CT를 향상시키기 위해 디자인기반 학습(design-based learning) 접근으로 스크래치 프로그래밍 교육을 실시하고 Computational Literacy(CL)능력을 평가하였다. CL능력에는 정보적 문제해결력, self-프로그래밍 능력, self-프로그래밍 흥미도가 포함되었는데 분석 결과 self-프로그래밍 능력과 self-프로그래밍 흥미도는 통계적으로 유의한 효과가 있는 것으로 나타났다.

가장 최근에 수행된 스크래치 프로그래밍을 CT개념에 연결시킨 연구로서 이은경은 중학교 1학년 34명을 대상으로 4주에 걸쳐 12차시의 스크래치를 활용한 프로그래밍 학습 활동을 진행하고 프로그래밍 학습이 학습자의 CT능력과 창의성 잠재력에 미치는 영향을 확인하였다[11]. 그 중 CT능력 평가를 위해 학습자의 프로젝트 결과물을 수집하고 CT 평가 프레임워크[3] 중 개념적 영역의 요소를 도입하여 순차, 반복, 조건, 이벤트, 자료, 연산의 총 6개의 기준을 설정하여 평가하였다. 그러나 이 연구에서는 창의적 잠재력 향상의 측정에 보다 초점이 맞춰져 있는 경향이 있으며 CT 평가 프레임워크의 세 개의 영역 중 하나의 영역인 개념 영역만을 평가 대상으로 활용하였다.

2.2 국외 선행 연구

국외에서 스크래치 프로그래밍 교육관련 연구들을 살펴보면 초기에는 디지털 콘텐츠를 통해 창의적인 자기 표현 능력 및 디지털 콘텐츠 생산 능력 향상에 초점을 두는 연구[12][15]에서 시작하여 최근에는 스크래치 프로그래밍 과정에서 향상되는 CT능력의 평가 방안 연구[3][13][16]로까지 진보하고 있는 것을 볼 수 있다.

Seiter와 Foreman[16]은 스크래치 프로그래밍을 통한 초등학생의 CT능력 평가를 위해 PECT (Progression of Early Computational Thinking)모형을 제안하였다. PECT모형은 증거 변수(Evidence Variable)와 설계 패턴 변수(Design Pattern Variable)를 포함하고 있다. 증거 변수들은 스크래치로 작성된 프로그램의 특성들(Looks, Sound, Motion, Variables, Sequence & Looping, Boolean Expressions, Operators, Conditionals, Coordination, User Interface Event, Parallelization, Initialize Location, Initialize Looks)로 구성되었으며, 설계 패턴 변수는 스크래치 프로그램의 맥락적 능숙도(contextual proficiencies)에 기초한 변수들(Animate looks, Animate Motion, Conversate, Collide, Maintain Score, User Interaction)로 구성된다. 학생들의 완성 프로그램을 두 개의 변수로 이루어진 매트릭스로 배치해서 학생의 수준을 3단계(기초, 발전, 능숙)로 평가한다. 또한 Computer Science Teachers Association[4]에서 개발한 각 CT개념(Procedure and Algorithms, Problem Decomposition, Parallelization and Synchronization, Abstraction, Data Representation)에 대해 설계 패턴 변수를 매핑하고 3단계 수준에서 증거로 삼을 수 있는 지표를 제안하였다.

가장 최근에 Brennan과 Resnick[3]은 CT 평가 프레임워크를 개발하였는데 그 주요 차원(dimension)으로 계산적 개념(computational concepts), 계산적 실행(computational practices), 계산적 관점(computational perspectives)을 제안하였다. 계산적 개념은 일곱 개의 개념 즉 시퀀스, 루프, 병행성, 이벤트, 조건, 연산자, 데이터로 구성되며, 계산적 실행은 4개의 실행 요소 즉 단계성과 반복성, 테스팅과 디버깅, 재사용과 리믹싱, 추상화와 모듈화로 구성된다. 계산적 관점으로는 3개의 요소가 제안되었는데 3요인은 표현하기, 연계하기, 질문하기이다. 이들 3차원의 요소들을 평가하기 위해서 프로젝트 분석, 산출물 기반 면담, 디자인 시

나리오 접근을 제안하고 각 접근에서 개념, 실행, 관점 영역에서 평가할 수 있는 내용을 제안하였다.

계산적 개념은 학생들이 프로그래밍을 하면서 반복, 병행처리 등과 같은 개념을 접하는 것을 의미하며, 계산적 실행이란 그런 개념들을 가지고 프로그램을 개발하면서 디버깅을 하거나 다른 학생들의 작업 내용을 리믹싱하는 것을 말한다. 또한 계산적 관점은 학생들이 일련의 학습 활동을 하면서 자신에 대해 또한 다른 동료들에 대해 형성하게 되는 관점을 의미한다.

첫 번째 차원인 계산적 개념(computational concepts)은 많은 프로그래밍언어들에 공통적으로 내포되어 있는 개념이며 MIT 미디어 랩에서는 다양한 스크래치 프로젝트에서 가장 유용하면서도 비프로그래밍 맥락(non-programming contexts)에 전이되는 일곱 개의 개념을 규명하였다. 일곱 개의 계산적 개념은 시퀀스(sequences), 루프(loops), 병행성(parallelism), 이벤트(events), 조건(conditionals), 연산자(operators), 데이터(data)이다.

두번째 차원인 계산적 실행(computational practices)은 학생들이 창의적으로 자신의 프로젝트를 만들면서 형성하게 되는 CT개념 구성의 과정에 초점을 둔다. 이것은 학생이 무엇을 배우느냐를 넘어서서 어떻게 배우느냐로 옮겨가는 사고의 과정 및 학습의 과정을 의미한다. 이러한 과정은 상호작용적 미디어 콘텐츠를 생성하는데 중요한 부분이며 프로그래밍뿐만 아니라 다양한 디자인 활동에도 유용하다. 많은 관찰을 통해 제안된 4개의 실행 요소는 단계성과 반복성(being incremental & iterative), 테스팅과 디버깅(testing & debugging), 재사용과 리믹싱(reusing & remixing), 추상화와 모듈화(abstracting & modularizing)이다.

마지막 차원인 계산적 관점(computational perspectives)은 학생들이 창의적으로 자신의 프로젝트를 만들면서 자신에 대한 이해, 타인과의 관계에 대한 이해, 그들을 둘러싼 테크롤로지 세상에 대한 이해를 반영한다. 계산적 관점에는 3개의 요소가 제안되었는데 3요인은 표현하기(expressing), 연계하기(connecting), 질문하기(questioning)이다.

Seiter와 Foreman[16]의 PECT모형과 Brennan과 Resnick[3]의 CT평가 프레임워크는 모두 CT능력을 평가하기 위한 방법을 제공하고 있으나 전자는 학생 작품을 기반으로 여러 측면을 평가하는 반면 후자는 학생 작품 기반으로는 개념 영역만을 평가할 수 있고 다른 두 영역은 학생 면담 등을 수행하는 것이 요구된다는 점이다. 따라서 현실적으로 학교 현장에서의 CT평가를 위한 방안을 제공하기 위해서는 학생 산출물에서 다양한 측면의 CT능력의 증거를 분석하는 것이 효율적일 수 있다고 판단된다. 따라서 본 연구의 스크래치 CT계발 프로그램 설계에서는 현장 적용의 효율성을 고려한 평가 방안을 고안하고자 한다.

3. 연구 방법 및 절차

본 연구에서는 먼저 국내·외 CT관련 문헌 연구를 통해 CT skills의 세부 역량은 무엇인지 도출하였다. 또한 도출된 세부 역량 중에서 스크래치 프로그래밍 수업을 통해 계발할 수 있는 CT세부 역량을 선택하기 위해 2013년도 새롭게 업그레이드된 스크래치 2.0 버전의 새 기능들을 분석하였다. 이 분석을 통해 나온 선택된 CT세부 역량을 다루는 초등교원양성기관 프로그래밍 수업 내용을 설계하기 위해 스크래치 프로그래밍 언어가 보급되기 시작한 이후 2008년부터 국내에서 스크래치 프로그래밍 효과성 연구를 분석하고 수업 맥락과 평가 방법을 검토하였다. 또한 국외의 스크래치 프로그래밍 관련 연구들을 검토하여 추이를 파악하고 CT관련 평가에 초점을 두고 있는 연구들로 본 연구에서 설계하고자 하는 스크래치 프로그램 설계 및 평가 방안에 기초 자료를 확보하였다. 이를 통해 교원양성기관에서 예비교사 대상의 스크래치 프로그래밍 수업 내용과 평가 루브릭을 개발하였다.

4. 연구 결과

4.1 CT 세부 역량

CT관련 문헌과 스크래치 프로그래밍 효과성 관련 문헌 분석과 스크래치 2.0 프로그램의 기능 분석을 통해 도출된 예비초등교사를 위한 CT세부 역량은 6개(절차 및 알고리즘, 병행화 및 동기화, 자료 표현, 추상화, 문제 분해, 시뮬레이션)로 나타났다. 이들 CT역량은 CSTA[4]에서 제시한 9가지 세부 CT역량, CT평

가 프레임워크, 스크래치의 제공 기능을 고려하여 추출되었다. <Table 1>은 도출된 CT역량과 스크래치 기능의 연관을 설명한 것이다. 6개의 세부 CT요인은 모두 CSTA 기준에 포함되는 요소로서 시뮬레이션 요소만을 제외하고는 CT평가 프레임워크[3]의 CT요소와도 일치를 보인다.

<Table 1> CT Competency

CT역량	스크래치 프로그래밍 학습 활동	CT 평가 프레임워크
절차 및 알고리즘	'동작', '형태', '소리', '반복' 등 다양한 블록들을 사용하여 순차, 반복, 조건 처리 등을 구현함.	시퀀스, 루프, 조건, 연산자, 단계성과 반복성
병행화 및 동기화	'이벤트' 블록들을 사용하여 동시에 다수의 블록들이 수행되도록 함. 객체(스프라이트)간에 방송 메시지를 교환하여 동기화를 구현함.	병행성, 이벤트
자료 표현	'데이타' 블록의 변수와 리스트를 활용하여 자료를 표현하고 관리함.	데이터
추상화	'블록 추가'를 통해 반복적으로 사용되는 블록의 셋트를 새로운 블록으로 만들고 필요할 때 사용함.	추상화와 모듈화
문제 분해	'자기 자신 복제 하기' 블록을 사용하여 복잡한 문제를 분해하여 해결함. 추상화와 모듈화와 연계하여 학습함.	재사용과 리믹싱
시뮬레이션	'블록 추가'를 통해 필요한 블록을 만들고 활용할 때마다 블록에 다른 파라미터를 전달함으로써 해당 파라미터에 대한 결과를 시뮬레이션함.	-

4.2 CT기반 프로그래밍 교수 · 학습 설계

본 연구에서는 도출된 6개의 CT세부역량의 계발을 위해 교원양성 기관에서 예비교사를 위해 한 학기에 진행할 수 있는 스크래치 프로그래밍 수업을 설계하였다. 분석과정에서 포함한 최근 업그레이드된 스크래치 2.0버전을 활용하여 진행할 수 있는 수업을 고안하였다. 수업 설계는 1주에 2시간 수업을 15주차 진행하는 일정을 기준으로 설정하였다. 수업 내용은 스크래치 관련 서적[21]의 내용을 분석하여 활용하거나 스크래치 사이트(http://scratch.mit.edu)에 공유된 콘텐츠 중 적절한 콘텐츠를 선택하여 활용하였다. 설계된 스크래치 프로그래밍 수업의 내용과 각 수업 내용이 CT세부 역량과 어떻게 연관이 되고 있는지 <Table 2>에 제시하였다.

<Table 2> Programming Lessons & CT Relatedness

주차	수업 내용	CT요소와의 연관성
1	스크래치 개요 스크래치 프로그래밍 개발 환경 익히기	수업에서 스크래치를 활용하여 프로젝트를 만드는 과정에서 어떤 CT역량들을 익히게 되는지 개괄적인 이해를 하게 됨.
2	바닷속 세상 표현 애니메이션 프로젝트	병행화(여러 물고기가 동시에 바닷속을 헤엄쳐 다님) 절차 및 알고리즘(물고기는 향하고 있는 방향으로 전진하다가 벽에 닿으면 다시 반대로 돌아감)
3	비보이 댄스 프로젝트	절차 및 알고리즘(비보이의 댄스 동작이 순차적으로 변경되어 애니메이션 효과를 냄) 병행화(댄스 동작, 음악 재생, 무대의 조명 변화가 모두 동시에 병행적으로 처리)
4	퀴즈 맞추기 프로젝트	자료 표현(변수의 사용을 통해 총점을 누적하고 합격/불합격 처리) 절차 및 알고리즘(문항의 순차적 진행과 정답 확인 과정) 동기화(보기 버튼이 클릭되었을 때 정답 확인 시작됨), 추상화(정답을 확인하는 부분을 새로운 블록으로 만들어 필요할 때마다 사용함)
5	공추적 게임 프로젝트	병행화(고양이가 비치볼을 추적하는 동시에 비치볼은 사용자의 마우스 움직임에 따라 다님) 동기화(30초 내에 고양이가 비치볼을 잡으면 패배 메시지를 표시하고 종료시킴)
6	두 사람의 대화 구현 프로젝트	절차 및 알고리즘(형준이와 제인이 각자 길을 가다가 대화를 나누고 다시 가던 길로 감)

주차	수업 내용	CT요소와의 연관성
		동기화(형준이의 말이 끝나면 제인이 바로 뒤이어 대화를 함)
7	피아노 건반 연주하기 프로젝트	동기화(해당키를 누르면 음계이름이 건반 위치에 나타났다 사라짐) 자료 표현(악기와 음량을 변수를 사용하여 처리) 병행화(건반이 동시에 눌려도 음계가 나오도록 함)
8	그림 패드 만들기 프로젝트	병행화(색의 선택과 크기 지정 키의 선택이 동시적으로 이루어짐)
9	합격자 발표 프로젝트	자료 표현(리스트에 합격자를 저장하고 합격자 명단에 있는 이름을 검색함)
10	대포 쏘기 게임 프로젝트	문제 분해(대포를 장전, 발사, 폭발 단계로 분해하여 처리할 내용을 각각의 모듈로 분해함) 추상화(분해한 문제를 하나의 모듈로 설계해서 모듈 당 하나의 블록을 만들어 사용함)
11	센서 보드활용 프로젝트	시뮬레이션(스위치의 세기, 빛의 밝기, 소리의 세기값을 파라미터로 블록에 전달하여 매번 다른 값에 대한 반응 결과를 관찰함)
12	팀프로젝트 설계 및 구현	각 프로젝트에서 CT요소가 어떻게 반영되었는지 기획서에 세부 사항을 작성함.
13 ~ 14	팀프로젝트 설계 및 구현	각 프로젝트에서 CT요소를 반영하여 구현함.
15	팀프로젝트 발표 및 피드백 제공	클래스 구성원들 간에 CT 요소 반영 여부에 대해 비판적인 관점으로 검토하고 피드백을 주고 받음.

4.3 CT역량 평가 방안

문헌연구를 통해 도출된 CT의 세부 역량을 수업 설계 내용으로 포함하면서 이를 일관성 있게 평가할 수 있는 방안을 고안하였다. Seiter와 Foreman[16]의 PECT모형은 디지털 스토리텔링이나 게임과 같은 디지털 콘텐츠 제작 맥락에서 나타나는 설계 패턴만을 포함하고 있어서 그 외의 디지털 유형을 포괄하지 못하는 제한점이 있고, Brennan과 Resnick[3]의 CT평가 프레임워크는 세 가지 영역을 제공하고 있지만 개념 영역외의 실행 영역은 학생 관찰 자료를 수집해야 가능하다는 점과 관점 영역은 인터뷰를 통해 직접 끌어내기 어렵다는 평가의 제한점을 가지고 있다. 따라서 현실적으로 교육 현장에서의 CT평가를 위한 방안을 제공하기 위해서는 학생 산출물에서 최대한 다양한 측면의 CT능력의 증거를 분석하는 것이 효율적일 수 있다는 판단하에 학생의 프로젝트에서 CT세부 요소를 보이는 증거들을 기초, 발달, 능숙 3단계의 수준으로 평가할 수 있는 루브릭을 고안하였다(<Table 3> 참조).

<Table 3> CT Competency Evaluation Rubric

CT 세부 요소	기초 (Basic)	발달 (Developing)	능숙 (Proficient)
절차 및 알고리즘	프로그램의 절차 및 알고리즘의 효율성에 개선할 부분이 많이 보임.	프로그램의 절차 및 알고리즘의 효율성에 개선할 부분이 다소 보임.	프로그램의 절차 및 알고리즘이 효율적으로 설계됨.
병행화 및 동기화	동시적 처리나 동기화 처리가 이루어지지 않거나 적절하지 않음.	동시적 처리는 적절히 이루어졌으나 객체간의 메시지 교환 처리가 없거나 적절히 이루어지지 않음.	동시적 처리가 적절히 이루어지고 객체간의 메시지 교환을 통해 프로그램 흐름의 동기화가 적절히 이루어짐.
자료 표현	변수나 리스트가 전혀 사용되지 않음.	변수 또는 리스트를 적절히 활용함.	변수와 리스트를 적절히 활용함.
추상화	사용자 자신이 만든 블록으로 구현된 것이 없음.	전체 프로그램에서 분해된 논리적 단위가 부분적으로 새 블록으로 구현됨.	전체 프로그램에서 분해된 모든 논리적 단위가 새 블록으로 구현됨.
문제 분해	알고리즘의 논리적 부분이 혼재되어 처리됨.	알고리즘의 일부를 논리적인 단위로 나누어 처리함.	알고리즘의 주요 부분 모두를 논리적인 단위로 나누어 처리함.

CT 세부 요소	기초 (Basic)	발달 (Developing)	능숙 (Proficient)
시뮬레이션	새 블록을 만들어 사용하였으나 파라미터(parameter)는 활용하지 않음.	새 블록에 파라미터(parameter)를 사용하여 파라미터 값에 따라 다른 결과를 도출하는 과정에서 부분적으로 부정확함이 보임.	새 블록에 파라미터(parameter)를 사용하여 파라미터 값에 따라 다른 결과를 도출하여 적절히 활용함.

5. 결론 및 제언

본 연구에서는 국내·외 CT관련 문헌 연구를 통해 CT skills의 세부 역량을 도출하고 초등교원양성기관에서 예비교사들의 프로그래밍 수업을 통해 CT세부 역량을 계발할 수 있는 수업 내용을 설계하고 평가 방안을 제안하였다. 본 연구에서 도출된 예비초등교사를 위한 CT세부 역량은 6개(절차 및 알고리즘, 병행화 및 동기화, 자료 표현, 추상화, 문제 분해, 시뮬레이션)로 나타났으며 이들 6개의 세부 요소들을 다룰 수 있는 구체적인 수업 내용과 함께 각 차시의 내용과 CT요소와의 구체적인 연관성에 초점을 두고 분석하였다. 또한 CT요소의 학습 여부를 평가하기 위해 교육 현장에서 학습자의 산출물에 나타난 CT요소의 수준(기초, 발달, 능숙)을 평가할 수 있는 루브릭을 제안하였다.

본 연구의 초점은 자칫 기초 프로그래밍 기능 습득과 콘텐츠 생성에만 머무를 수 있는 수업을 CT 세부 역량과 체계적으로 연결시켜 봄으로써 CT역량 강화를 위한 방안을 제시하였다는 것이다. 또한 학생 콘텐츠의 외현적 기능 구현에 치중할 수 있는 평가 영역에서도 학습자가 CT역량을 습득하였다는 것을 평가할 수 있는 방안을 고안하였다는 점이다. 본 논문에서 제안된 수업과 루브릭은 향후 실제 초등교원양성 기관에서 적용되고 검증되는 절차를 거쳐 정교화되어야 할 것이다. 실증적 검증 및 정교화를 위한 후속 연구를 통해 예비교원들의 CT역량 계발을 위해 활용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] Adams, R. (2013). New national curriculum to introduce fraction to five-year-olds. theguardian. URL:http://www.theguardian.com/politices/2013/jul/08/michael-gove-education-curriculum-fractions

[2] Bae, H. J., Lee, E. K., & Lee, Y. J. (2008). A Problem Based Teaching and Learning Model for Scratch Programming Education. *Journal of Korean association of Computer Education,* 12-3, 11-22.

[3] 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.

[4] Computer Science Teachers Association. Computational Thinking. 2012. http://csta.acm.org/Curriculum/sub/CompThinking.html.

[5] Cho, S. H., Song, J. B., Kim, S. S., & Baek, S. H. (2008). The Effect of a Programming Class Using Scratch. *Journal of The Korean Association of information Education,* 12-4, 375-384.

[6] Cho, S. H., Song, J. B., Kim, S. S., & Lee, K. H. (2008). The Effect of CPS-based Scratch EPL on Problem Solving Ability and Programming Attitude. *Journal of The Korean Association of information Education,* 12-1, 77-88.

[7] Choi, H. S. (2013). Study on Approaches of Teaching and Measuring Computational Thinking. *Korea Association of Information Education,* 4-1, 283-288.

[8] Choi, S. Y. (2011). An Analysis of “Informatics” Curriculum from the Perspective of 21st Century Skills and Computational Thinking. *Journal of Korean association of*

Computer Education, 14-6, 19-30.

[9] Digital Daily (2013.4.18.). MISP, Pursuing Computer Programming Education from Elementary and Middle Schools. URL:http://www.ddaily.co.kr/news/news_view.php?uid=103537

[10] Kim, S. H., & Han, S. K.(2012). Design-Based Learning for Computational Thinking. Journal of *The Korean Association of information Education,* 16-3, 319-326.

[11] Lee, E. K. (2013). Computer Education Curriculum and Instruction : Creative Programming Learning with Scratch for Enhancing Computational Thinking. *Journal of Korean association of Computer Eduation,* 16-1, 1-9.

[12] Maloney, J., Peppler, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with scratch. SIGCSE'08, March 12-15. 2008. Portland, Oregon, USA. 367-371. Retrieved Dec. 18, 2013 from http://web.media.mit.edu/~mres/papers/sigcse-08.pdf

[13] Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M.(2010). Learning computer science concepts with Scratch. ICER 2010. August 9-10, Arhus, Denmark. 69~76.

[14] Papert, S.(1993). Mindstorms: Children, computers, and powerful ideas. Cambrideg, MA: Perseus Publishing.

[15] Peppler, K. A. & Kafai, Y. B.(2005) Creative coding: Programming for personal expression. Retrieved Dec. 18, 2013 from http://download.scratch.mit.edu/CreativeCoding.pdf

[16] Seiter, L., & Foreman, B.(2013). Modeling the learning progressions of computational thinking of primary grade students. ICER'13. August 12-14, 2013. San Diego, California, USA. 59-66.

[17] Song, J. B., Cho, S. H., & Lee, T. W. (2008). The Effect of Learning Scratch Programming on Students' Motivation and Problem Solving Ability. Journal of *The Korean Association of information Education,* 12-3, 323-332.

[18] Wikipedia(2013.6.20.). Estonia. URL:http//ko.wikipedia.org/%EC%97%90%EC%8A%A4%ED%86%A0%EB%8B%88%EC%95%84

[19] Wing, J. M.(2006). Computational thinking. Communications of the ACM, 19-3, 33~35.

[20] _____(2011). Computational thinking－what and why?. CMU Research Notebook. Retrieved from http://link.cs.cmu.edu/article.php?a=600

[21] Yim, B. C., & Jung, I. K. (2010). Scratch Programming. Seoul: Human Science.

저 자 소 개

최 형 신

1988 이화여자대학교 (전자계산학 학사)
1993 (미) NJIT(컴퓨터과학 석사)
2007 이화여자대학교 (교육공학 박사)
2008 이화여자대학교 BK연구교수
2009~현재 춘천교육대학교 컴퓨터교육과 교수
관심분야: Computational Thinking, 스크래치 프로그래밍, 로봇활용교육
e-mail: hschoi@cnue.ac.kr