

프로그래밍 과정에서 나타나는 초보학습자들의 행동 및 사고과정 분석

김수환[†] · 한선관^{††} · 김현철^{†††}

요 약

본 연구의 목적은 CL(Computational Literacy) 교육에서 중요한 교육 내용인 프로그래밍 문제해결 과정을 규명하는데 있다. 프로그래밍 초보학습자들의 인지적 사고과정 분석을 통해 프로그래밍 문제해결 과정에서 나타나는 사고 특성을 조망하는 분석틀을 제시하고, 실제 CL 문제해결 과정에서의 초보학습자들의 사고과정을 규명하는 것이다. 이를 위해 사고발성법을 적용하여 프로그래밍 과정에서 나타나는 여러 상황을 분석하였다. 본 연구에서는 학생들의 사고 특성을 추출하는 분석코드와 프로그래밍 과정 코드를 이용한 코딩조직을 개발하였고, 프로그래밍 과정에서 겪는 어려움과 이를 해결하는 과정을 Nvivo를 사용하여 종합적으로 분석하였다. 특히 프로그래밍 과정에서 나타나는 사고력의 활용에 대한 분석과 시사점을 제시하였다. 본 연구는 프로그래밍 과정에서 나타나는 인지적 사고 특성과 흐름을 이해하는데 기여하며 프로그래밍에서의 사고 과정을 과학적으로 분석할 수 있는 준거를 제시한다.

주제어 : Computational Literacy 교육, 프로그래밍교육, 사고발성법, Think Aloud

Analysis of Programming Processes Through Novices' Thinking Aloud in Computational Literacy Education

SooHwan Kim[†] · SeonKwan Han^{††} · HyeonCheol Kim^{†††}

ABSTRACT

The purpose in this paper is to provide the theoretical framework of characteristics of programming thinking processes in computational literacy education. That is, we developed the theoretical framework through analyzing novices' cognitive thinking processes, applied it to the real situation about computational literacy problem-solving processes and defined characteristics of the processes. For this purpose, we tried to analyze characteristics of programming thinking processes of novices by using think-aloud method. Also we developed the programming process code about novices' cognitive processes and programming processes, and analyzed the process that novice faced and overcame programming barriers by using qualitative research tool, Nvivo. As a result, we found what characteristics of programming problem-solving processes were and how novices used the thinking skill in the process. This study contributes to understand programming problem-solving processes and provides the criterion to analyze the processes scientifically.

Keywords : Computational Literacy Education, Programming Education, Programming Processes, Thinking Aloud

† 정 회 원: 고려대학교 컴퓨터교육학과 박사과정
 †† 중신회원: 경인교육대학교 컴퓨터교육과 교수
 ††† 중신회원: 고려대학교 컴퓨터교육과 교수(교신저자)
 논문접수: 2010년 10월 01일, 심사완료: 2010년 12월 04일

1. 서론

정보교육과정이 개정되면서 알고리즘과 프로그래밍 영역이 확대되어 프로그래밍 교육이 중요하게 대두되고 있다[1]. 알고리즘과 프로그래밍 영역은 컴퓨터를 이해하고 조작하는데 핵심적인 부분이며, 학생들이 자신의 아이디어를 만들고 컴퓨터로 표현하는데 필요한 능력을 기를 수 있는 중요한 영역이다[2]. 이러한 능력은 결국 CL(Computational Literacy)를 기르는데 기본적인 교육내용 및 방법으로 부각되고 있다[2][3].

현재 프로그래밍 교육은 초보 학습자들에게 어려운 영역으로 인식되고 있으며, 많은 연구에서도 그 원인을 규명하고 이를 극복하기 위한 방안을 제시하고 있다[4]. 하지만 이러한 연구는 대부분 설문조사나 학습자들의 결과물, 관찰을 토대로 한 연구로 프로그래밍 과정에서 실질적인 사고과정을 조망하여 연구한 사례는 미흡한 실정이다.

학습자의 사고과정 자체를 연구하는 방법은 전통적으로 심리학에서 면담, 관찰을 통한 방법이나 최근에는 뇌과학 측면에서 뇌파를 측정하거나 뇌스캔을 통하여 사고과정을 규명하는 연구가 시도되기도 한다[5]. 하지만, 뇌스캔 방법은 아직 실험 단계이고, 면담이나 관찰 또한 프로그래밍 과정 자체를 이해하고 규명하기에는 부족하다. 따라서 본 연구에서는 CL 교육에서 프로그래밍 초보학습자들의 사고과정을 이해하고, 규명하기 위해 사고발성법(Thinking Aloud)기법[6]을 사용하여 학습자들이 프로그래밍 과정에서 일어나는 사고과정을 기록하고 Nvivo 8.0[7]을 사용하여 분석하였다. 사고발성법을 통하여 프로그래밍 과정에서 학습자들이 어떤 사고과정을 통해 문제를 해결하는지, 어떤 사고력을 사용하여 문제를 해결하는지에 대한 분석을 실시하였다.

먼저 문헌연구를 통하여 프로그래밍에서 사용되는 코딩조직을 추출하고 이를 재조직하여 프로그래밍 과정에서 나타나는 코딩조직으로 구성하였다. 프로그래밍 학습과정에서 사고발성한 전사스크립트를, 구성한 코딩조직을 가지고 분석함으로써 프로그래밍에서 사고과정을 분석하였다. 사고과정 분석에는 Nvivo 8.0을 이용하여 코딩분석을 실시하였다. 결국 본 연구에서는 프로그래밍

문제해결 과정에서 나타나는 코딩조직과 이에 대한 분석을 통하여 프로그래밍에서의 사고과정을 규명하였다.

2. 이론적 배경

2.1 프로그래밍을 통한 CL 문제해결 과정

이옥화[8]에 의하면 프로그래밍 학습은 첫째 언어를 배우는 기능 습득을 목적으로 하는 경우와, 둘째 문제 해결능력 신장과 같은 고등정신 기능 함양을 위해 언어를 수단으로 하는 경우가 있다고 한다. 현대 교육에서는 정보교육[1]의 목표가 언어 습득 보다는 정보적 문제해결력을 기르는데 중점을 두고 있으므로, 본 연구에서도 프로그래밍을 통한 문제해결 과정에 초점을 맞추었다. 이러한 프로그래밍의 문제해결 과정을 살펴보면 첫째, 프로그래밍 문제의 목표를 인식하는 것, 둘째, 계획을 세우고, 셋째, 프로그래밍 언어로 코딩하고, 넷째, 오류수정을 통하여 목표와 현상태의 차이를 최소화 하는 과정을 거친다[1]. Mayer[9]의 경우에도 이와 비슷하게 만들기(creating), 이해하기(comprehending), 수정하기(modifying), 교정하기(debugging)의 문제해결 과정을 제시하고 있다. Dale과 Lewis[10]도 컴퓨터 문제해결과정을 분석과 명세화 과정, 알고리즘 개발 과정, 구현과정, 관리과정으로 나누어 제시하였다. CL을 이용한 문제해결 과정 또한 문제를 작은 단위로 분해하고, 이를 재조합, 재구성하는 과정, 새로운 의미로 완성하는 과정을 거친다[2]. 이러한 결과를 종합해 볼 때 프로그래밍을 활용한 CL 문제해결 과정은 문제를 정보단위로 쪼개고, 이를 해석하고 이해하여 재구성하며, 구성한 것을 점검, 수정하여 완성된 의미를 부여하는 과정으로 구성된다. 따라서 본 연구에서는 초보 프로그래밍 학습자들의 이러한 학습단계에서 어떤 행동과 사고활동을 보이는지 분석하였다.

2.2 사고발성법

사고발성법은 피실험자가 문제를 해결하는 과정에서 떠오르는 모든 생각이나 의문점 등을 말

로 소리 내어 표현하도록 하고, 숙련된 실험자가 이것을 오디오 테잎이나 비디오 테잎으로 녹음하는 방법이다[6]. 실험이 끝난 다음에 실험자는 녹음된 것을 다시 보고 들으면서 원고를 작성(Protocol)하고 각 사고과정을 코드화(coding)한다. 이 방법은 문제해결 과정에서 진행 중인 생각을 그대로 언어로 표현함으로써 관찰 불가능한 사고의 과정을 엿볼 수 있다는 장점을 가지고 있다. 다른 기법들과 달리 가급적 질문이나 실험자들을 재촉하는 방법을 사용하지 않기 때문에 과제해결 과정을 방해하지 않는다. 또한 자신의 사고활동에 대해서 해석하지 않고, 추론하지 않으므로 아주 직접적이고 시간의 지연 없이 동시적인 구두보고로 이루어지므로 기억오류로 인한 문제가 없다[11]. 국내연구에서는 국어 교육에서 주로 이루어지고 있으며, 수학, 과학 문제해결 과정에서의 연구도 이루어지고 있다[11][12]. 따라서 본 연구에서도 학습자들의 사고과정을 추출하여 분석하기 위해 사고발성법을 적용하였다.

3. 연구의 방법 및 절차

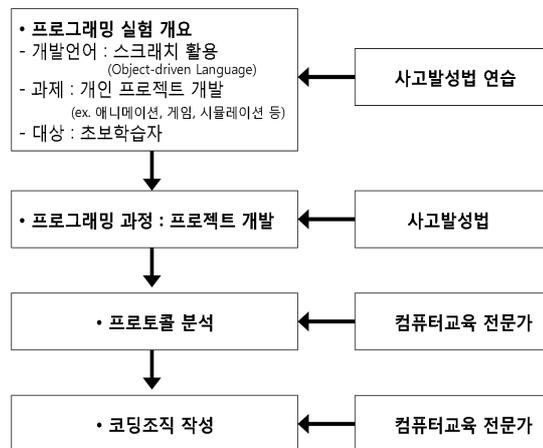
3.1 연구과정 및 교육내용

본 연구에서는 사고발성법을 통한 초보 프로그래밍 학습자의 사고과정을 분석하는 것이 목적이므로 연구의 절차는 크게 3단계로 이루어졌다. 먼저 2009년도에 1학기에 프로그래밍을 처음 배우는 수도권 소재 A대학교에서 스크래치[13]을 통한 CL 과제를 해결하게 하고 초기 코딩조직을 개발하였다. 2009년도 2학기과 2010년도 1학기에 걸쳐서 B대학교 학생 60여명을 대상으로 스크래치를 통한 프로그래밍 교육을 실시하면서 사고과정을 추출하였다. 사고발성법의 사용을 위해 학습자들에게 사전 설명을 실시했으며, 연습 시간을 제공하였다. 2차 본 실험에서 사고발성법을 잘 사용하는 학생들을 선별하여 7명의 학생의 샘플을 녹음하고 분석하였다.

<표 1> 연구의 과정

단계	활동 내용	비고
준비	다양한 전공의 대학생 선별	
기초 실험	스크래치 활용 CL 교육 -정보적 사고방법 -아이디어 구상 및 구현	사고발성법 및 관찰/ 초기 코딩조직 개발
실험 1	스크래치 활용 CL 교육 -정보적 사고방법 -컴퓨터 교육실습 -아이디어 구상 및 구현	2009년 2학기 사고발성법 및 관찰/ 코딩조직 개발
실험 2	스크래치 활용 CL 교육 -정보적 사고방법 -컴퓨터 교육실습 -아이디어 구상 및 구현	2010년 1학기 사고발성법 및 관찰, 비디오/ 코딩조직 완성

프로그래밍의 과정의 경우 사용하는 언어의 특성과 문제의 유형에 따라 달라질 수 있으므로, 본 연구에서는 CL의 목적에 적합한 스크래치를 개발 언어로 선택하였다. 스크래치는 초보자들이 자신의 아이디어를 다양한 멀티미디어 프로그램으로 쉽게 작성할 수 있는 언어로써, Object-based, Event-based 언어이고[14], 명령어를 블록형식으로 조립 가능한 Visual Programming Language의 특성을 가지며 SmallTalk를 기반으로 하고 있다. 또한 과제는 기존의 프로그래밍 과제들이 아닌 개인별로 자신이 구상한 애니메이션, 게임, 시뮬레이션 등을 개발하는 것으로 제시하였다.



<그림 1> 연구의 과정

각 실험별로의 사고발성법 적용과 코딩조직 개발 과정은 <그림 1>과 같다.

또한 CL교육으로써의 프로그래밍 교육 내용의 주별 과정은 <표 2>와 같다.

3.2 프로토콜(Protocol)의 작성 및 코딩화

학습자들이 사고발성한 프로토콜 자료를 그대로

<표 2> 교수-학습 활동 주제

차시	활동 주제	프로그래밍 요소
1	스크래치 인사하기, CL 문제해결법	인터페이스 설명
2~3	도형그리기, 애니메이션	절차, 반복, 분기
4~5	공튀기기, 물고기 게임	변수, 난수, 객체 통신
6~7	사칙연산 프로그램	데이터 입출력, 배열, 불린 로직, 연산자
8~9	음악 프로그램	불린 로직, 객체통신
10~11	미술 프로그램	데이터 입출력, 객체통신
12~13	프로젝트 만들기	알고리즘 설계 및 구현
14~15	프로젝트 만들기	알고리즘 설계 및 구현
16	평가	

로 기록하는 것에 중점을 두었으므로 말이 매끄럽지 못하거나 틀리게 단어를 사용한 것도 그대로 기록하였으며, 또한 학생들이 중간에 말을 멈추고 휴지시간을 갖는 것 등도 ‘...’를 사용하여 기록하였다. 학습자들이 사고발성한 과정은 컴퓨터 녹음 프로그램을 통하여 기록하고, 학생들이 어떤 활동과 사고를 하는지 주목하여 분석하였다.

4. 코딩조직의 개발 및 해설

4.1 코딩조직의 전체적인 모델

코딩조직을 개발하기 위해 <표 3>에서와 같이 먼저 프로그래밍 학습 과정에서 일어나는 학습자들의 반응, 행동에 대한 문헌을 분석하였다. 먼저 이옥화[8], Mayer[9], Dale & Lewis[10], Livert 외 [15]의 프로그래밍 과정연구에서 초기 코딩조직을 추출하였다. 예를 들면 Livert 외[15]가 제시한 planning의 경우 ‘과업예상’, ‘과업추측’으로, 이옥화가 제시한 계획세우기는 ‘과업추측’, ‘결정’ 등의

<표 3> 코딩조직 개발 분석

학자 과정	Dale & Lewis	Livert, et al.	Schneiderman & Mayer	Bennedsen & Caspersen	이옥화	기초 코딩조직	확정 코딩조직
프로 그래밍 과정	분석과 명세화	planning	skill and knowledge acquisition	Use of IDE	목표인식	문제읽기	과업예상
						알고리즘 개발	결정
	알고리즘 개발	developing	comprehension	incremental developing	계획세우기	구현	-
						절차과악	객체초르기
						객체정의	-
						필요기능과악	실행결과판단
						전반적요소결정	실행결과결정
						객체편집	객체편집
	구현	testing	composition	testing	프로그램 언어로 코딩	명령어찾기	명령어해석 명령어탐색 명령어판단
						절차확인	기능의문 절차막힘 절차확인 실행장면묘사
						에러발견	에러발견
						에러수정	해결방법인식 에러수정
관리	debugging	debugging	error controlling	online documentation	코드결정	코드결정	
					코드오류	코드확인실행 코딩시도 코딩작업 실행 및 코딩	
					에러발견	제안	
					에러수정	에러수정	
		modification	Model-driven programming	에러수정	의욕	코딩확장시도	

<표 4> 주요 코딩조직 설명

코딩조직	설 명
과업예상	프로그래밍 과업에 대한 어렴풋한 생각 예) 바에 닿으면 공이 튕기면 되지 않을까?
과업추측	프로그래밍 과업에 대한 확실한 예측 예) 공은 자유롭게 움직이고, 벽에 닿으면 튕기게 만들면 되겠구나.
결정	만들고자 하는 내용 및 과정 시 나타나는 결과에 대한 결정
객체고르기	사용될 스프라이트나 객체 고르기
명령어해석	명령어의 뜻과 사용에 방법에 대한 해석
명령어탐색	적절한 명령어를 찾기 위한 탐색
명령어판단	원하는 명령어인지 아닌지 판단 예) 그래, 조건에 따라 달라져야 하나까 만약 ~라면이야.
실행결과추측	코드 실행 후 나타나는 결과를 추측해 보는 것
실행결과결정	코드 실행 후 나타나는 결과에 대한 결정 예) 이 코드를 실행하면 변수에 1이 저장되도록 해야지
객체편집	객체의 크기, 모양, 색깔 등을 편집
기능의문	객체가 어떤 기능을 가져야 할지 의문이 들 때
절차막힘	다음 코드나 실행과정에 대한 막힘
절차확인	실행해 보지 않는 상황에서 코드를 보면서 절차를 확인
코드결정	명령어보다는 큰 단위의 코드 결정
코드확인실행	코드가 맞는지 확인해 보기 위해 실행하는 과정
코딩시도	확실하지 않은 코드를 여러 가지 시도해 보는 것
코딩작업	머릿속에서 결정된 코드로 코딩하는 작업
에러발견	코드나 실행과정에서의 에러 발견
해결방법인식	에러를 해결할 방법을 알게 된 것
에러수정	에러 인식 뒤에 에러를 수정하는 행위
의욕	프로그래밍과 결과에 향상에 대한 의욕
제안	새로운 기능이나 효율적인 방법에 대한 제안
코딩확장시도	만든 코드에 기능을 추가하거나 확장

코딩조직으로 만들었다. Schneiderman과 Mayer [16]의 연구에서는 ‘에러발견’, ‘해결방법인식’, ‘에러수정’ 등의 코딩조직을 추출하였다.

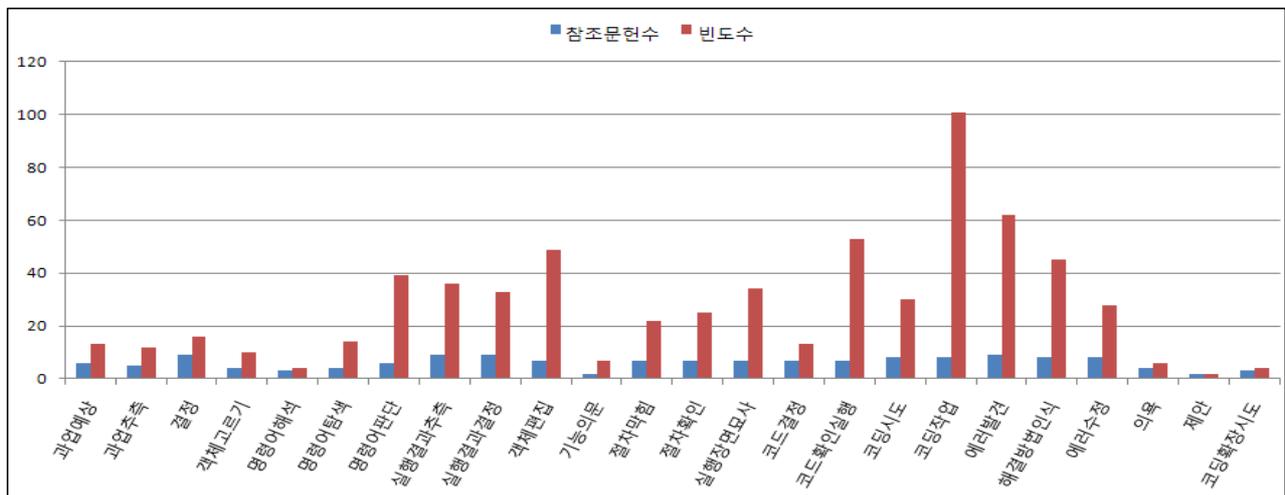
Bennedsen[17]의 경우 초보 학습자들이 프로그래밍 과정에서 필요한 요소를 제시하여 참조하였다. 학자에 따라 구분방법이 다르나 결국 프로그래밍의 과정은 이옥화[8]의 4단계를 재구성한 것이므로 본 연구에서는 프로그래밍 단계를 목표인식, 계획세우기, 코딩, 에러수정으로 구분하고 개발된 코딩조직을 각 단계별로 구분하여 제시하였다. 각 코드별 관계는 <표 3>에서 제시한 바와 같이 단계별로 분류하였으며, ‘에러수정’과 같이 코딩과 에러수정 단계에 동시에 포함되는 경우도 있다.

사고력에 대한 구분은 <표 5>에서와 같이 한국교육개발원에서 개발한 사고력측정도구에서 구분한 사고력의 정의를 이용하여 각 코딩조직별로 해당되는 사고력을 분석하여 제시하였다[18].

<표 5> 사고력의 범주

사고력	세부요소
분석적 사고	이해력, 분석력, 의미 파악능력, 번역 능력, 해석 능력, 외삽 능력
추론적 사고	연역추론, 귀납추론, 결론 도출
종합적 사고	부분을 전체로 묶는 능력
대안적 사고	발상전환적 접근, 대안생각
비판적 사고	해석, 추론, 평가

코딩조직은 크게 3단계에 걸쳐서 개발되었다. 먼저 위에서 밝힌바와 같이 문헌 연구를 통해 프로그래밍 과정에서 나타날 수 있는 코딩조직 44



<그림 3> 코딩조직 빈도수

개를 전부 나열한 후 의미가 같은 것을 재분류한 후 실제 프로그래밍 과정에서 나타나는지 기초실험을 통해 확인하였다. 기초실험을 통해 개발된 코드는 총 33개였다. 이 때 개발된 코드는 명확하지 않고 중복되는 부분이 있어서, 컴퓨터교육 전문가 3명의 회의를 거쳐 <표 4>와 <그림 3>과 같이 최종 24개의 코딩조직으로 확정하였다.

4.2 프로그래밍 과정의 분석

초보 프로그래밍 학습자들의 프로그래밍 과정을 분석하기 위해 학습자들로부터 추출한 프로토콜을 대상으로 본 연구자가 만든 코딩조직(coding system)을 이용하여 피실험자가 프로그래밍 과정에서 나타낸 사고과정을 프로토콜에 나타난 순서대로 코딩화 하였다. 코딩의 신뢰성과 타당성을 높이기 위해 프로토콜 3부를 작성하여 컴퓨터교육 전문가 3명의 분석과정을 거쳐 코드가 일치하는 것은 완성된 코드로 선택하였다. <그림 3>과 <표 6>에서 나타난 '참조문헌수'는 코딩조직이 발생한 문서수를 뜻하며, '빈도수'는 전체 문서(9개 프로토콜)에서 나타난 빈도를 나타낸다. 프로그래밍 과정에서 중요한 요소로 나타난 코딩조직

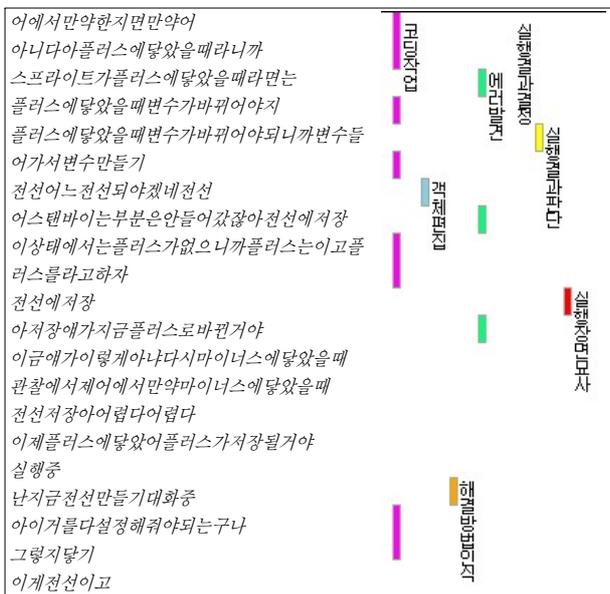
은 <그림 3>에서와 같이 객체편집, 코드확인 실행, 해결방법 인식, 에러발견, 실제 코딩작업과 같은 요소이다. 객체편집의 경우 초보학습자들은 자신의 생각을 표현하는 데에 디자인적인 요소나 화면배치, 적절한 객체선택 및 편집에 많은 시간을 투자하는 것으로 나타났으며, 이는 프로그래밍적인 요소와 더불어 디자인적인 요소가 중요한 요소로 작용한다는 점을 시사한다. 또한 해결방법 인식은 프로그래밍 전 단계에서 나타나는데 특히 에러발견과 더불어 이를 해결하는 방법을 인식하고 시도하는 과정 가운데서 많이 발생하여 전체 과정에서 중요한 요소임을 보여준다. 코딩작업은 프로그래밍 과정 중이므로 많은 비중을 차지하는 것으로 나타났다.

<그림 4>에서와 같이 초보학습자들은 프로그래밍 과정에서 코딩작업 중간 중간에 에러를 발견하고 해결방법을 인식하는 과정이 반복되는 양상을 보여준다. 이러한 과정은 스크래치가 코드 입력과 동시에 바로 실행하면서 프로그램을 확인할 수 있기 때문에 가능하다. 특히 코딩조직 간의 연계를 분석해보면, 코딩작업의 경우 명령어를 선택하거나 절차를 확인하는 등의 코딩조직을 포함

<표 6> 코딩조직 분석

번호	프로그래밍과정	코딩조직	세부요소	사고력	참조문헌수	빈도수	빈도수/전체빈도수(%)
1	목표인식	TE	과업예상	종합,추론	6	13	1.98%
2		TA	과업추측	분석,추론	5	12	1.83%
3		D	결정	분석,추론,종합	9	16	2.44%
4	계획세우기	SO	객체고르기	분석	4	10	1.52%
5		IC	명령어해석	분석,비판	3	4	0.61%
6		EC	명령어탐색	분석,비판	4	14	2.13%
7		JC	명령어판단	분석,비판	6	39	5.64%
8		RJ	실행결과추측	추론	9	36	5.49%
9		RD	실행결과결정	분석,추론,비판	9	33	5.03%
10	프로그래밍언어로 코딩	EO	객체편집	분석	7	49	7.47%
12		QF	기능의문	분석,종합,대안	2	7	1.07%
13		SP	절차막힘	추론,종합,분석,대안	7	22	3.35%
14		IP	절차확인	분석,추론,종합,비판	7	25	3.81%
15		DE	실행장면묘사	-	7	34	5.18%
16		DC	코드결정	종합	7	13	1.98%
17		EC	코드확인실행	분석,추론,비판	7	53	8.08%
18		TC	코딩시도	대안,분석,추론	8	30	4.57%
19		C	코딩작업	분석,종합	8	101	15.40%
20		에러수정	FE	에러발견	비판,분석	9	62
21	KS		해결방법인식	분석,종합,대안	8	45	6.86%
22	CE		에러수정	분석,추론,종합,대안	8	28	4.27%
23	DP		의욕	대안	4	6	0.91%
24	SF		제안	종합,대안,추론	2	2	0.30%
25	TE		코딩확장시도	대안,비판,종합	3	4	0.61%

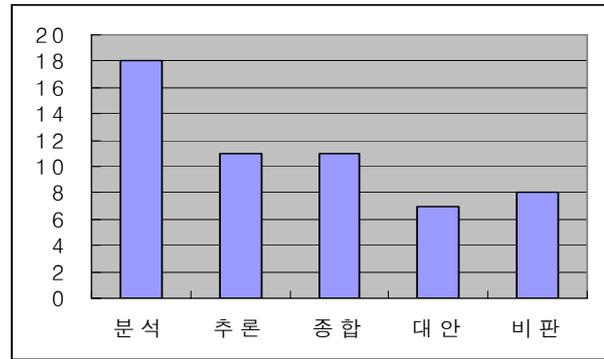
한다. 즉, 코딩작업이 기본적인 작업으로 진행되면서 그 중간 중간에 객체를 편집하는 행위와 에러를 발견하고 해결방법을 인식하는 사고과정이 반복적으로 나타나는 양상을 보인다. 특히 프로그래밍 과정에서 구분되어 있는 코딩과 에러수정의 단계가 복합적으로 일어나며, 코딩과정에서 문제를 잘게 쪼개어 해결하는 양상에서 명령어 선택, 코드확인, 에러수정 등의 코딩조직이 복합적으로 작용함을 알 수 있다.



<그림 4> 초보자들의 프로그래밍 패턴 예

4.3 사고력 사용 분석

프로그래밍 과정에서 보이는 사고력을 분석하기 위해 <표 5>와 같이 사고력의 범주를 구분하고, 분석, 추론, 종합, 대안, 비판적 사고를 각 코딩조직과 연결하였다. 각 코딩조직과의 연결을 위해서 컴퓨터교육 전문가 5명에게 각 코딩조직과 연결되는 사고력을 연결하도록 하였다. 전문가 3명 이상이 동의하는 사고력의 경우 각 코딩조직과 연결되는 것으로 판단하였다.



<그림 5> 사용된 사고력

가장 많이 나타나는 사고력은 분석적 사고력으로 대부분의 코딩조직에 연결되는 사고력으로 나타났다. 특히 중요한 코딩조직으로 나타난 코드확인 실행에는 분석적, 추론적, 비판적 사고력이 사용된다고 보았고, 코딩시도에는 대안적, 분석적, 추론적 사고력이 사용된다고 보았다. 특히 에러를 발견하고 이를 해결하는 패턴에서는 비판적, 분석적, 종합적, 대안적 사고력이 사용되는 것으로 나타났다. 에러를 발견하는 데에는 비판적인 사고력이 사용된다고 5명 모두 일치된 의견을 보였고, 해결방법인식에서는 종합적 사고력에 5명 모두 일치된 의견을 보였다. 이는 에러를 발견하는 것이 흐름을 추론하고 평가하는 과정에서 일어나게 되면 또한 이를 해결할 때는 부분적인 것을 종합적으로 생각하는 종합적 사고가 중요한 사고력으로 사용됨을 볼 수 있다.

5. 결론 및 제언

본 연구에서는 CL교육을 위한 모든 학생들에게 자신의 아이디어를 정보적으로 표현할 수 있는 방법으로 프로그래밍을 가르치고 이 과정에서 나타나는 문제해결과정과 사고력의 발현 과정을 분석하였다.

본 연구를 통해 도출된 시사점은 다음과 같다.

첫째, 초보 프로그래밍 학습자들의 프로그래밍 과정에서 개발된 코딩조직은 25개 요소이다. 개발된 코딩조직은 초보학습자들의 프로그래밍 사고과정의 추적을 가능하게하며, 이를 활용하여 프로그래밍의 효율적인 지도와 평가에 대한 기초자료로 활용할 수 있다.

둘째, 가장 빈번하게 나타나고 중요한 요소로 밝혀진 코딩조직은 객체편집, 코드확인 실행, 해결방법 인식, 에러발견이다.

셋째, 초보학습자들은 코딩과 더불어 객체 편집에 많은 시간을 할애하며, 이는 자신의 아이디어를 표현하는데 필요한 디자인적인 요소에 많은 시간을 사용함을 알 수 있다. 또한 스크래치의 특성상 실행결과를 바로 확인할 수 있는 언어의 경우, 코딩과 더불어 실행이 바로 확인 가능함으로 초보자일지라도 코딩을 하면서 바로 실행결과를 확인하고, 실행결과가 자신의 생각과 다르게 나타나면 바로 에러를 발견하고 이를 해결하기 위한 방안을 모색하는 패턴을 보였다.

넷째, 스크래치를 활용한 아이디어 구현 프로젝트 만들기 활동은 초보학습자들의 문제해결과정이 전문 프로그래머의 문제해결과정과 비슷한 양상을 나타낸다. Bennedsen과 Caspersen[17]가 밝힌 프로그래밍 과정에 따르면 문제를 작은 단위로 쪼개서 해결하는 패턴은 초보학습자들이 아닌 전문 프로그래머가 보여주는 양상인데, 이러한 양상이 초보학습자들에게 가능하다는 것은 언어의 특성과 더불어 산출결과물이 정해진 프로그래밍 문제가 아닌 아이디어 산출을 통한 과제를 해결하는 과정에서 나타난 결과이다.

따라서 CL교육의 방향은 기존의 프로그래밍 문제가 아닌 자신의 아이디어를 구체화하고 이를 컴퓨터를 통해 구현하여 문제를 해결하는 과제를 제시하는 것으로 나아가야 한다.

본 연구의 결과로써 초보학습자들의 프로그래밍 과정에 대한 분석을 위한 코딩조직이 개발되었으며, 이는 향후 모든 학생들에게 프로그래밍 교육을 실시할 때, 교육 방법과 평가 방법에 대한 기초자료로 활용될 수 있다. 또한 프로그래밍 과정에서 사용된 사고력 분석과 사용언어 및 과제 제시 내용은 CL교육에서의 프로그래밍 교육에 대한 방향을 제시해준다. 향후 연구로는 본 연구에서 개발된 코딩조직을 토대로 사고력의 과정과 논리추론에서 발생하는 사고력 과정을 분석하는 연구가 필요하다.

참 고 문 헌

- [1] 교육과학기술부 (2007). **중학교 교육과정 해설(V)**. 서울: 교육과학기술부.
- [2] 김수환, 한선관, 김현철(2010). Computational Literacy 교육에서 프로그래밍 능력과 학습자 특성에 관한 연구 -학습스타일과 다중지능을 중심으로-. **컴퓨터교육학회논문지**, 13(2), 15-23.
- [3] Wing, J. M.(2008). "Computational Thinking and Thinking About Computing." 2009. 8. 20 검색 <http://www.cs.cmu.edu/afscs/usr/wing/www/ct-and-tc-long.pdf>
- [4] 이정모, 이건호 (1998). **초보자의 C언어 학습과정에 대한 인지심리학적 분석 연구**. 한국인지과학회 연구보고서. 1-20.
- [5] 이조옥 (2003). **초·중등학생의 과학적 사고과정에서 나타난 뇌파 특성**. 석사학위논문, 한국교원대학교.
- [6] Newell, A. & Simon, H. A. (1972). *Human Problem Solving*. Englewood Cliffs, NJ., Prentice-Hall.
- [7] 박종원 (2009). **NVivo 8 프로그램의 활용 : 현장 연구자를 위한 질적데이터의 과학적 관리와 분석**. 서울: 형설출판사.
- [8] 이옥화 (1993). 로고 프로그래밍의 교육적 의의와 실천 방안 모색. **교육공학연구**, 8(1), 81-102.
- [9] Mayer, R. E. (1992). *Thinking, problem solving*. New York: W. E. Freeman and Company.
- [10] Dale, N. and Lewis, J. (2007). *Computer Science Illuminated* Jones and Bartlett Publishers, 3rd edition.
- [11] 홍진곤, 강은주 (2009). 사고구술법을 이용한 수학 영재의 사고 특성 연구. **수학교육연구 논문지**, 19(4), 565-584.
- [12] 김태선, 김범기 (2005). 과학관련 선 그래프를 해석하는 고등학생들의 발생사고 과정 분석. **한국과학교육학회지**, 25(2), 122-132.
- [13] Scratch. 2010. 9. 11 검색 <http://scratch.mit.edu>
- [14] 김수환, 김교준, 김승범, 이원규, 김현철 (2008). **교육용 프로그래밍 언어의 동향과 교**

육적 적용 방안. **한국정보과학회 학술발표논문집**, 35(1). 375-376.

- [15] Livet, M., Courser, M. & Wandersman, A. (2008). The Prevention Delivery System: Organizational Context and Use of Comprehensive Programming Frameworks. *American Journal of Community Psychology*, 41(3), 361-378.
- [16] Schneiderman, B. & Mayer, J. E. (1979). Syntactic/semantic interaction in programmer behavior: A model and experimental results. *International Journal of Computer and Information Science*, 8(3), 219-238.
- [17] Bennedsen, J. & Caspersen, M. E. (2008). Exposing the Programming Process. *Reflections of the Teaching of Programming, LNCS 4821*, 6-16.
- [18] 김명숙, 박정, 김영정, 민찬홍 (2002). **사고력 검사 개발 연구(II) -비판적 사고력 검사 제작편-**. 한국교육과정평가원. 연구보고서.



김 수 환

1999 인천교육대학교(교육학학사)
2006 경인교육대학교
컴퓨터교육과(교육학석사)

2007~현재 고려대학교 컴퓨터교육과 박사과정
관심분야: 컴퓨터교육, Computational Literacy, EPL, Unplugged, CSCL
E-Mail: lovejx@korea.ac.kr



한 선 관

1991 경인교육대학교(교육학학사)
1995 인하대학교 교육대학원
(컴퓨터교육학석사)
2001 인하대학교 전자계산공학과
(전산학 박사)

2002~현재 경인교육대학교 컴퓨터교육과 교수
관심분야: 인공지능, 지능형교수시스템, 초등정보교육, e-Learning, u-Learning
E-mail: han@gin.ac.kr



김 현 철

1988 고려대학교 전산학과 학사
1990 Univ. of Missouri - Rolla
(전산학석사)
1998 Univ. of Florida(전산학 박사)
1998 GTE Data Services, Inc.
시스템 분석가

1998~1999 삼성 SDS 책임컨설턴트
2005~2006 Univ. of Florida 대우교수
2010~2011 일본 홋카이도대학 정보기반센터
특임교수
1999~현재 고려대학교 컴퓨터교육과 교수
관심분야: 컴퓨터교육, 기계학습알고리즘
E-Mail: harrykim@korea.ac.kr