

분수 연산을 위한 지능형 교수시스템의 설계 및 구현

서병태 · 한선관 · 조근식
인하대학교 전자계산공학과

요약

정적인 CAI 테크닉에 의해 개발된 기존의 프로그램들은 고정된 교육과정과 내용을 가지고 있기 때문에 다양한 수준의 학습자에게 적합한 학습을 하기에는 어려운 점이 매우 많다. 또한 학습자들 개개의 독특한 학습에 대해 적절한 교육방법을 제공하기 위한 보조 기능에 대해 그 유연성이 부족하다. 이러한 문제들을 해결하기 위하여 본 논문에서는 분수 연산을 위한 지식 기반의 교수 시스템의 설계 및 구현을 하였다. 분수 연산에 관련된 지식들을 분석하여 학습자의 이해 수준과 잘못 인식하고 있는 버그, 상호 관련된 오류를 진단하고 학습자에게 적합한 교수학습 모듈을 구현한다. 학습자의 이해 수준과 오류를 진단하기 위한 진단 모듈과 학습자 인터페이스 모듈을 실험을 하여 기존 분수학습 CAI보다 지능형 분수 교수 시스템의 학습 성취도가 높아짐을 살펴보았다.

Design and Implementation of Intelligent Tutoring System for Fractional Computation

Byeong-Tae Seo, Sun-Gwan Han, Geun-Sik Jo

ABSTRACT

The traditional programs developed by the existing CAI technique have the fixed curricular, which make it difficult to deliver various study materials that fit the learners of various levels. In addition, a lack of the flexibility prevents from helping to make their methodology in studying uniform open minded. In order to solve these problems, we have designed and implemented a learner interface that can exclude the limits in the learners active study in solving the fractional operation. In addition to the user interface, this study includes a diagnosis module that can intellectually extract the status of learners understanding, ostensible bugs, and the associated misconceptions through the interface. The experimentation based on the learner interface and the diagnosis module shows that this system correctly diagnoses the level of learners' understanding and the errors in learning, which greatly helps the individualized study.

1. 서론

기존의 CAI는 표현 방식이나 학습상의 절차는 학습하기에 앞서 프로그램에 명시되어야 하므로 학습자 개개인의 학습요구에 적용하는 능력이 부족한 문제점을 갖게 된다. 따라서 이러한 문제점을 보완하기 위해서 나온 기법으로 CAI에 인공지능(Artificial Intelligent)의 기법을 적용해서 학습자에게 더욱 지능적이고 적응력 있는 교육 시스템을 개발하려고 시도하였는데 이것이 지능형 교육 시스템(ITS : Intelligent Tutoring System)이다[8].

지능형 교수 시스템은 학습자의 지식 수준을 진단하며 문제 해결 과정을 추론하여 학습자 개개인의 수준에 적절한 학습 전략을 제공하므로 이러한 지능형 교수 시스템이 개발 보급되면 아동의 학습 능력에 따라서 개별화 수업이 가능하며 개개인의 학습 결손 부분을 선별적으로 교정해 줄 수 있어 많은 교육적 효과가 기대되고 있다[3].

ITS의 학습자 모델에서는 학습자의 지식에 대한 현재의 상태를 나타내고 영역 개념을 학습하는데 학습자가 만날 수 있는 오류를 찾고 분석하게 되며 이 때 학습자 모델내의 지식 수준을 추론하는 것을 진단이라고 한다[7]. 진단은 관찰 가능한 학습자의 질의-응답 행동을 통하여 숨겨진 지식 상태를 알아 내려는 것이다[2].

학습자 모델링 기법은 여러 가지 연구내용이 발표되었으나 한가지 모델링 기법만으로는 학습자의 지식 상태를 표현하는데 부족하여 두 가지의 모델링 기법을 상호 보완 관계를 유지하는데 사용하고 있으며, 진단 기법은 학습자 모델 및 문제 영역에 따라 다양하게 나타나고 있다[8].

CAI 기반의 초등학교 분수 연산에 대한 교육용 코스웨어를 구현할 때 진단 기법은 분수 연산의 특수성, 즉 개념과 계산조작이 매우 복잡하고, 논리적인 공리성으로 인하여 해결 과정이 너무 추상화되어 있기 때문에 학습자의 지식 상태를 정확히 추론하기 힘들다. 또한 학습자의 응답을 대부분의 시스템에서 정답과 오답으로만 분리함으로써 학습자의 지식 상태를 분석하는데 한계를 나타내고 있다[9].

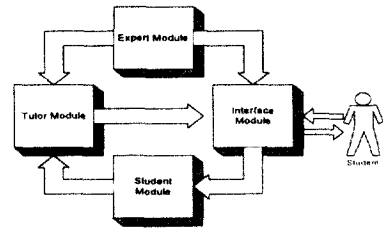
따라서 본 연구에서는 학습자의 지식 상태, 지식

의 결합, 표면적인 버그(Bug) 및 그와 관련되어 함축된 오류(Misconception)를 지능적으로 추출할 수 있는 지능형 분수 학습 시스템을 설계하고 구현하는데 연구 목적이 있다.

2. 학습자 중심 ITS 고찰

2.1. ITS의 일반적 구성

ITS는 크게 4가지의 기본적인 모듈로 구성되어 있다. 이 모듈들의 내용을 자세히 살펴보면 전문교사가 가르칠 내용을 포함하는 전문가 모듈(Expert Module), 학생의 반응에 따라 학생의 수준 정도를 장기적으로 파악하여 그 학생의 수준에 대한 정확한 정보를 제공하는 학습자 모듈(Student Module), 학습자 모듈에서 이루어진 질적 평가에 따라 전문교사가 행할 수 있는 교수 방법을 적용하는 학습 전략 모듈(Tutor Module), 컴퓨터와 학생간에 필요한 정보를 주고받는 학습자 인터페이스 모듈(Interface Module)로 나누어진다. ITS의 네 가지 구성 요소들은 [그림 1]과 같다[2].



[그림 1] ITS의 기본 구성도

2.2. 학습자 모델링 기법

학습자 모델로부터의 출력은 학습자의 지식 상태에 대한 진단 결과이므로 교과 과정의 선택을 통한 진도, 코치 또는 자발적인 조건, 새로운 문제 제시, 설명의 제안 등과 같은 용도를 위해 사용된다.

학습자 모듈에서 학습자에 대한 정보를 수집하는 방법은 오버레이 모델, 버그 모델, 차분 모델, 버그 부분 목록 등으로 구분된다[4].

오버레이 모델은 학습자 모델링 기법 중 가장

많이 사용되는 것으로서 아동이 성취해야 할 학습 목적을 그 학습 내용의 전문가가 갖고 있는 지식의 획득과 분석을 기초로 하여 세부적인 수준에까지 구분, 이를 기준으로 아동의 학업 성취 정도를 비교하는 방법이다.

버그 모델은 사전에 학습 내용에 따라 아동이 학습을 해 나가는데 저지를 수 있는 모든 가능한 오류를 경험적으로나 개념적 분석에 의하여 확인하고 아동이 학습을 진행하는 과정에서 오류를 범했거나 문제에 부딪칠 때 그것이 어떤 학습 인지상의 오류에 기인한 것인지를 추론하는 방법이다.

2.3. 진단 및 분석

현재 ITS에서는 학습자에게 주어진 문제와 그 대답을 분석하여 학습자에 대한 모든 교육 자료가 얻어진다. 학습자의 직접적인 대답의 맞고 틀림에 의한 정보는 명시적 증거이며, 학습자의 질문이나 도움 요청도 학습자의 지식 상태에 대한 간접적인 증거로서 이러한 경우는 학습자가 자신의 지식에 대해서 확신을 갖지 못한 것으로 부정적인 증거가 된다.

이와 같이 학습자의 지식, 지식의 결합, 표면적인 버그 및 그와 관련되어 함축된 오류 (Misconception)를 지능적으로 추출해 내는 것을 학습자 진단(Student Diagnosis)이라고 한다[5].

학습자 진단은 특정 학습 과정에 관한 학습자의 성향과 교수 영역에 관련된 학습자의 능력을 평가하기 위해서 여러 가지 방법으로 사용되는데 학습진도 결정, 의도적인 조언 제공, 문제의 선택, 적절한 설명의 채택 등으로 요약할 수 있다[7].

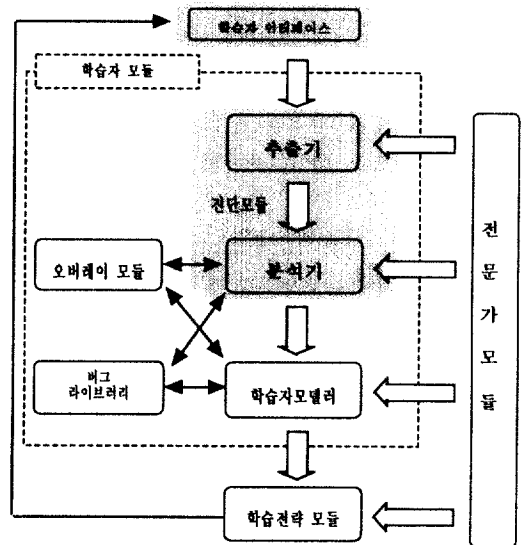
Kurt VanLehn은 학습자의 지식 인지상태를 인지적 상태(mental state), 중간상태(intermediate state), 최종상태(final state)의 세 가지 단계로 분류하고 이것을 대역폭(Bandwidth)이라 하였다[4].

진단기법은 각 상태에 따라 다음과 같이 9가지로 분류된다[2]. 즉, 모델추적, 경로결정, 조건유도, 계획인식, 주제추적, 전문가 시스템 접근방식, 결정트리, 생성과 검사, 상호진단 등이다.

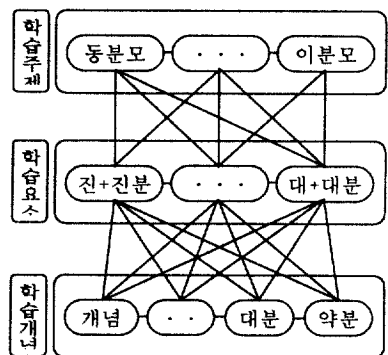
3. 지능형 교수시스템의 구조 및 설계

3.1. 설계 및 구성도

본 연구에서 설계한 시스템은 분수 연산의 특수 성인 위계성 측면, 결과보다는 풀이 과정의 중요도, 풀이 과정상의 오류 진단능력, 오류 분석과 선행 지식 및 개념의 복합성을 기초로 하여 다음 [그림 2]와 같이 설계하였다.



[그림 2] 시스템의 구성도



[그림 3] 학습 내용의 계층 구조도

3.2. 학습 개념 분석

분수의 덧셈을 대상으로 분석한 결과 [그림 3]과 같이 3개의 계층 구조로 세분화된다. 즉 최상위층

은 학습 주제에 관한 내용이고, 이는 여러 개의 학습 요소들로 이루어지며, 각 학습 요소들은 다시 여러 가지 선행 학습 개념들로 구성된다[9].

1) 분수 덧셈 연산의 유형

분수에서 분모간의 관계는 동분모, 서로소, 배수, 2보다 큰 최대공약수(GCM)가 있고 최소공배수(LCM)가 아닌 경우라는 4가지 유형이 있다. 따라서 통분의 비교는 이 4가지 유형을 모두 거쳐 통과했는지 여부로 결정할 수 있다.

<표 1> 분수 덧셈의 연산 유형

구분	진분수+진분수					
울림	x	x	x	o	o	o
약분	x	o	o	x	o	o
자연수	x	o	x	o	x	o
구분	대분수+진분수					
울림	x	x	x	o	o	o
약분	x	o	o	x	o	o
자연수	x	o	x	o	x	o
구분	대분수+대분수					
울림	x	x	x	o	o	o
약분	x	o	o	x	o	o
자연수	x	o	x	o	x	o

2) 분수의 덧셈에서 요구되는 능력

분자끼리만 더하기, 분모 분자의 최대공약수가 2 이상이면 약분하기, 계산 결과가 가분수이면 대분수로 고치기, 분모 분자의 최대공약수를 구하기, 두 분모의 최소공배수 구하기, 자연수와 분수를 더할 때는 '+' 부호를 생략해서 나타내기, 자연수는 자연수끼리 더하고 분자는 분자끼리 더하기 등이 있으며 이것을 지식베이스로 표현하면 다음과 같다.

- 분수(나눗셈) : $\frac{C}{B}$
 $C \div B (B >= 1 \ \& \ C >= 1)$
- 진분수(common) : $\frac{C}{B}$
 IF $B > C$ THEN Proper_Fraction
- 가분수(common) : $\frac{C}{B}$
 IF $B < C$ THEN Improper_Fraction
- 대분수(common) : $A \frac{C}{B}$
 IF $A > 1 \ \& \ B > C$ THEN Complex_Fraction
- 받아들림(가분수 \Rightarrow 대분수) : $\frac{C}{B}$
 IF $B < C$ THEN $A = \text{int}(C/B), B = B, C = \text{mod}(C/B)$

· 약분 : $\frac{C}{B}$

IF $\text{GCM}(B,C) >= 2$ THEN $B = B/\text{GCM}(B,C) \ \& \ C = C/\text{GCM}(B,C)$

· 통분 : $\frac{C}{B}, \frac{F}{E}$

IF $\text{LCM}(B,E) >= B \ \& \ \text{LCM}(B,E) >= E$
 THEN $B = \text{LCM}(B,E) \ \& \ E = \text{LCM}(B,E) \ \& \ C = \text{LCM}(B,E)/B * C \ \& \ F = \text{LCM}(B,E)/E * F$

· 동분모 : $\frac{B}{A}, \frac{D}{C}$

IF $A = C$ THEN Common_Fraction

· 이분모 : $\frac{B}{A}, \frac{D}{C}$

IF $A \neq B$ THEN Decommon_Fraction

· 동분모분수의 덧셈 : $\frac{B}{A} + \frac{C}{A} = \frac{B+C}{A}$

IF Common_Fraction THEN $\frac{B+C}{A}$

· 이분모분수의 덧셈 : $\frac{B}{A} + \frac{D}{C} = \frac{B \times C + A \times D}{A \times C}$

IF Decommon_Fraction

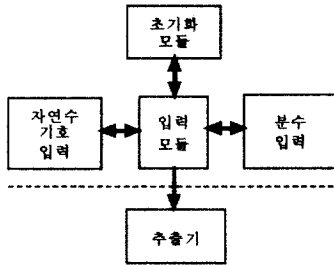
THEN $\frac{B \times C + A \times D}{A \times C}$

3.3. 학습자 중심의 인터페이스 모듈

인터페이스 모듈은 컴퓨터와 의사소통을 하는 모듈이다. 학습자는 이 모듈을 통하여 컴퓨터에 자료를 입력시키면 컴퓨터는 시스템 안에서 진단 모듈을 통하여 처리되고 그 결과나 산출된 자료를 사용자에게 제공하여 준다.

고정화된 인터페이스는 정답과 오답만을 검사할 수 있기 때문에 여러 문제를 해결하여야만 아동에 대한 지식 수준의 파악이 불가능하며, 학습자의 반응이 수동적일 수밖에 없기 때문에 아동의 사고를 왜곡시킬 가능성이 있다. 본 연구에서는 인터페이스 모듈이 진단 모듈과 직접적인 관련성을 가지므로 먼저 학습자 중심의 인터페이스 모듈을 [그림 4]와 같이 설계 및 구현하고자 한다.

이러한 학습자 인터페이스 모듈은 학습자의 의도에 따라 풀이 과정을 자유롭게 입력할 수 있으므로 확산적인 사고가 가능하고, 풀이 과정에서 나타나는 다양한 오류를 분석할 수 있는 환경을 제공할 수 있다.



[그림 4] 학습자 중심의 인터페이스 모듈 구성도

3.4. 추출기

추출 모듈은 학습자 반응으로부터 학습자의 지식 상태를 분석하는데 필요한 여러 가지 정보를 수집한다. 분수 연산의 풀이 과정에 있어서 오류를 진단할 수 있는 공통적인 인자들로는 풀이 과정의 단계를 구분 짓는 연산자와 값을 표현하는 자연수, 분자, 분모 등이 있다.

분수 연산의 풀이 과정은 학습자마다 다르므로 다음과 같이 여러 가지 형태로 나타낼 수 있다.

- 문제 : $\frac{2}{3} + \frac{1}{2} =$
- 유형 1 : $1\frac{1}{6}$ (통분-진분수덧셈과정 생략)
- 유형 2 : $\frac{7}{6}$ (통분-올림과정 생략)
- 유형 3 : $\frac{4}{6} + \frac{3}{6} = 1\frac{1}{6}$ (진분수덧셈과정 생략)
- 유형 4 : $\frac{4}{6} + \frac{3}{6} = \frac{7}{6} = 1\frac{1}{6}$

입력 형식이 $\frac{B}{A} + \frac{D}{C} = \frac{F}{E} = G\frac{I}{H}$ (유형 4)일 경우의 문자열 변환 규칙은 다음과 같다.

- ' $\frac{B}{A} + \frac{D}{C} =$ ' 문자열 변환 : 'ⓐA,B,+ⓑC,D,='
- ' $\frac{F}{E} =$ '의 문자열 변환 : 'ⓒE,F,='
- ' $G\frac{I}{H}$ '의 문자열 변환 : 'GⓓH,I,'

결국 학습자가 입력한

' $\frac{B}{A} + \frac{D}{C} = \frac{F}{E} = G\frac{I}{H}$ ' (유형 4)'은 다음과 같이 문자열로 표현할 수 있다.

ⓐA,B,+ⓑC,D,=ⓒE,F,=GⓓH,I,

<표 2> 추출 모듈의 실행 결과

구분	구분	구분					
		1	2	3	4	5	6
기호	1		+	=	+		
자연수	2						1
분자	3	2	1	4	3	7	1
분모	4	3	2	6	6	6	6
결과		$\frac{2}{3}$	$+$ $\frac{1}{2}$	$=$ $\frac{4}{6}$	$+$ $\frac{3}{6}$	$=$ $\frac{7}{6}$	$1\frac{1}{6}$

이와 같이 변환된 문자열은 연산자, 기호, 분모, 분자 등의 인자들을 추출하기 위해 추출기로 보내어지며, 이를 실행하여 주어진 문자열을 추출기에 의해 추출된 결과값을 테이블로 나타내면 <표 2>와 같다.

3.5. 분석기

분석 모듈은 추출 모듈에서 추출한 인자들을 이용하여 학습 오류 상태를 분석하게 된다. 학습자의 응답을 분석하는 능력은 학습자의 질을 결정하며, 응답은 명시적 또는 암시적일 수 있는 많은 사실을 이용한 내부 추론의 결과이다. 정확한 또는 부정확한 답이 몇 개의 다른 방법에서 해석될 수 있다.

분석 기법에는 Model Tracing, Path Finding 등 여러 가지가 있으나[4] 본 연구에서는 학습자 중심의 인터페이스를 통하여 얻어진 분수 연산 풀이 과정을 Model Tracing 기법에 의해 분석하여 함축된 오류를 지능적으로 추출할 수 있도록 설계하였다.

분수 덧셈의 학습자 응답에 대한 진단 규칙은 다음과 같다.

- 학습자 응답 : $\frac{2}{3} + \frac{1}{2} = \frac{4}{5} + \frac{3}{5} = \frac{7}{5}$
 ① ② ③
- Common_String := 'ⓓ3,2,+ⓔ2,1,=ⓕ5,4,+ⓖ 5,3,=ⓗ5,7,'

<표 3> 학습자 응답 Table

수 식	①	②	③
구분자		+	=
자연수			
분자	2	1	4
분모	3	2	5

Common_String의 문자열을 추출기에 의해 <표

3>과 같이 구분자를 중심으로 분류한다.

- Rule 1 : 수식과 수식의 구분은 '='로 한다.
- Rule 2 : 비교식은 ①번식과 ②번식의 두 수식으로 비교하되 수식의 해석이 진행된다면 ②번식을 ①번식으로 ③번식을 ②번식으로 Swap한다.
- Rule 3 : 수식내의 구분자는 '+'기호만을 인정하고 그 외에 존재하는 모든 기호는 무시한다.
- Rule 4 : '+'기호를 구분되는 것은 '수'로 표현한다.
- Rule 5 : 대분수를 분리할 경우 반드시 자연수를 먼저, 진분수를 그 다음에 표현한다.
- Rule 6 : 모든 수식은 1회에 같은 Tracing Rule을 적용한다.

' $\frac{2}{3} + \frac{1}{2} = \frac{4}{5} + \frac{3}{5} = \frac{7}{5}$ '라는 학습자의

오답을 Tracing Rule을 적용하여 풀이 과정에서 나타나는 오류를 단계별로 누적시켜 다음과 같이 오류를 추적할 수 있다.

□ Comparison 1 :

①과 ② ($\frac{2}{3} + \frac{1}{2} = \frac{4}{5} + \frac{3}{5}$)

While (Step<=10)

{ IF Step=Error THEN Count++
ELSE Skip }

- Step 1~3 : skip
- Step 4 : 통분오류 (Step 4의 계수기 1 증가)
- Step 5~10 : skip
 - ①에서 ②로의 풀이 과정은 통분 과정이나 통분 오류로 판정됨.

□ Comparison 2 :

②와 ③ ($\frac{4}{5} + \frac{3}{5} = \frac{7}{5}$) Step=1

While (Step<=10)

{ IF Step=Error THEN Count++
ELSE Skip }

- Step 1~6 : skip
- Step 7 : 받아들림 오류 (Step 7의 계수기 1 증가)

• Step 8~10 : skip

- ②에서 ③으로의 풀이 과정은 진분수의 덧셈이나 정답 부분이 가분수로 남아있어 받아들림 오류로 판정됨.

□ Diagnosis Result

진단 결과는 Tracing Rule의 각 단계별 오류 계수기(Count)를 누적시킨 결과값이 되며, 오류 계수기의 값에 따라 다음과 같이 오류 유형에 따른 중요도(Important)를 결정한다.

While (Step<=10) {

IF Count=1 THEN Important = Low

IF Count=2 THEN Important = Mid

IF Count>=3 THEN Important = High }

이러한 분석 결과를 이용하여 학습전략 모듈은 진단규칙에 의해 오류 형태별 중요도에 따른 우선순위를 정하여 Feed Back 또는 오류 형태와 관련된 학습을 출력시킨다. 진단규칙에서 조건은 학습자의 지식 상태를 나타내는 사실로서 AND와 OR로 연결하고, 행동(action)은 학습자의 오류를 치료하거나 선수 학습 내용이 된다.

4. 구현 및 실험

본 연구에서 설계한 진단 모듈의 실효성을 입증하기 위해 분수 연산을 위한 프로토타입을 개발하였다. 분수 연산 ITS는 교육용 저작 도구인 PASS 2000과 CLIPS OCX를 이용하여 구현하였다. 인터페이스는 PASS2000을 웹에 플러그인하였으며 CLIPS는 추론 엔진으로 사용하였다.

CLIPS는 추론을 위한 추론 엔진과 RETE 알고리즘을 이용하여 패턴 매칭을 효율적으로 구현할 수 있는 인공지능 언어이다.

본 연구에서는 ITS 프로그램과 CAI 프로그램을 투입하여 풀이 과정 위주의 학습자 반응을 진단한 결과와 단답형의 학습자 반응을 진단한 결과를 비교함으로써 오류 진단의 효율성을 실험하였다.

실험을 위해 초등학교 48명을 대상으로 사전 검사를 실시하여 학습 수준이 낮은 아동들로 10명의 실험 집단을 구성하였다.

실험 집단에 ITS 프로그램과 CAI 프로그램을

투입하여 분수 덧셈 10문항을 풀도록 하였다. 사전 검사에 의한 학습자별 오류 결과는 <표 4>와 같으며 약분과 통분에서 가장 많은 학습 오류가 발견되었다.

<표 4> 사전 검사에 의한 학습자별 오류 결과

오류 학습자	덧셈	약분	통분	올림	오류개수
1	○	○	○	○	4
2		○	○		2
3		○		○	2
4	○		○	○	3
5		○	○	○	3
6	○	○	○	○	4
7		○	○		2
8	○	○	○	○	4
9	○	○	○	○	4
10		○	○	○	3
계	5	9	9	8	31

실험 집단을 대상으로 ITS 프로그램과 CAI 프로그램을 투입하여 학습자별 오류를 진단한 결과는 <표 5>와 같다.

<표 5> ITS와 CAI에 의한 진단 결과

오류 학습자	덧셈		약분		통분		올림		오류개수	
	ITS	CAI	ITS	CAI	ITS	CAI	ITS	CAI	ITS	CAI
1	○	○	○		○	○	○		4	2
2			○	○	○				2	2
3			○	○			○		2	1
4	○	○			○		○	○	3	2
5			○	○			○		2	1
6	○	○	○		○	○	○	○	4	3
7			○	○					2	1
8	○	○	○		○	○	○		4	2
9	○	○					○	○	2	2
10			○	○	○	○			3	2
계	5	5	8	5	7	5	8	3	28	18

실험 결과 사전 검사에서의 오류 결과와 비교하여 실험 집단에 대한 전체 오류 진단을 ITS가 90.3%, CAI가 58%로 나타나 CAI 프로그램보다 본 연구에서 제안한 진단 모듈을 내장한 ITS 프로그램이 더 정확하고 많은 오류를 진단할 수 있었다. 오류 유형별로 살펴보면 <표 6>과 같다.

ITS와 CAI의 진단율이 많은 차이를 보이는 이유는 CAI가 단답형 형태의 학습자 반응을 정답과 오답의 비교만으로 분석하였고, 프로그램의 구조상 여러 문제를 풀어야만 오류 유형을 분석할 수 있도록 설계되어 있기 때문인 것으로 판단된다.

<표 6> 오류 유형별 ITS와 CAI에 의한 진단 결과

오류유형 투입도구	덧셈	약분	통분	올림	오류개수
사전 검사 결과	5	9	9	8	31
ITS 오류개수	5	8	7	8	28
ITS 백분율	100%	88.8%	77.7%	100%	90.3%
CAI 오류개수	5	5	5	3	18
CAI 백분율	100%	62.5%	55.5%	37.5%	58%

이로써 본 연구에서 설계한 진단 모듈이 특정 문제에서 가능한 많은 풀이 과정을 필요로 하는 연산에서 더 효과적이고, 이러한 진단 기법이 여러 문제를 풀어야만 오류 유형 분석이 가능한 단점을 보완할 수 있는 것으로 나타났다.

5. 결론

분수 연산에 있어 대부분의 교육용 소프트웨어가 학습자의 오류를 진단하는데 한계를 가질 수밖에 없었던 것은 학습자의 응답을 정답과 오답만으로 분리함으로써 학습자의 지식 상태를 효과적으로 분석할 수 없었기 때문이다.

따라서 ITS의 개발 성패는 학습자의 반응을 효과적으로 분석할 수 있는 지능형 시스템의 개발에 달려 있다고 볼 수 있다.

본 연구에서는 분수학습을 통하여 학습자 중심의 인터페이스를 고려한 풀이 과정 위주의 지능형 교수시스템을 설계하였다. 학습자의 개인차에 따른 반응을 다양하게 입력받을 수 있게 하고, 학습자 응답으로부터 연산 과정의 단계별 오류를 추출 분석하여 학습자의 지식 결함과 관련된 함축된 오류를 효과적으로 진단할 수 있도록 하였다. 연구의 타당성을 실험하기 위해 시스템을 분수 연산 프로그램에 내장시켜 실험한 결과 연산 오류 유형을 정확하고 신속히 진단할 수 있었다.

또한 기존의 CAI 프로그램과 비교하여 학습자의 오류를 쉽게 판단하여 학습의 성취도를 향상시킬 수 있었다.

향후 연구 방향은 분수 연산에 있어서 분수의 유형을 더욱 세분화하여 지식을 추가하고 아동 개인에게서 나타나는 여러 변인들을 학습자 모델로

구축하여 분수 연산에 있어서의 오류 원인을 더 자세히 추론하고 그에 알맞은 학습을 할 수 있도록 하여 완전 개별화 학습을 할 수 있는 프로그램의 개발 및 연구가 기대된다.

참고문헌

[1] Alessi, S.M., & Trollip, S.R., Computer Based instruction : methods and development. Englewood Cliffs, NJ, 1985.

[2] Javaid, K., The Development of CAI : An Expert System In Education. East Carolina Univ, 1987.

[3] Kearsley, G.P., Computer Based training : A guide to selection and implementation. Reading, MA : Addison-Wesley, 1983

[4] Martha C. Plson, J. Jeffrey Richaedon & Elliot Soloway, Foundations of Intelligent Tutoring System, Lawrence Erlbaum Associates, inc. Publishers, 1988.

[5] Park, O., Perez, R.S., & Seidel, R.J., Intelligent CAI : Old wine in new bottles or a new vantage? In G.P. Kerasley (Ed.), Artificial Intelligence : Applications and Methodology, Reading, MA : Addison Wesley, 1987.

[6] Visetti, Y.M., Dague, P., "Plan inference and student modelling in ICAI," AAI, pp 77-81, 1987.

[7] Wenger, E., Artificial intelligence and tutoring system: Computational approaches to the communication of knowledge. Los Altos, CA: Morgan Kaufmann, 1987.

[8] 김주영, "학습자 모델과 그 진단 방법의 분석," 한국정보 과학회 추계 전산 교육 학술 발표회 논문집, pp. 54-63, 1990

[9] 김진석, "국민학교 아동의 분수 계산에서 오류 유형 분석," 한국교원대학교 석사학위논문, 1995.

[10] 백영태, "지능적 교육 시스템을 위한 학습자 모듈 설계에 관한 연구," 인하대학교 석사학위 논문, 1993.

[11] 심임섭, "지능형 교육시스템," 정보과학회지 제 12권 제 9호, 1994.

[12] 임성택, "ITS를 위한 가감 연산 오류 유형 분석 기법," 컴퓨터교육연구 제 2권 1호, 1995.

서 병 태

1991년 인천교육대학교 졸업
 1999년 인하대학교 전자계산교육과 졸업(석사)
 1997년~현재 인천 과학교육 연구원 연구사
 연구 분야: 지능형 교수 시스템, 웹기반 교육, 저작 도구 및 멀티미디어 교육, 컴퓨터 교육
 e-mail JIMAROMA@hitel.net

한 선 관

1991년 인천교육대학교 졸업
 1995년 인하대학교 전자계산교육과 졸업(석사)
 1998-1999년 인하대학교 전자계산공학과(박사과정)
 1999년~현재 인천교육대학교 컴퓨터교육과 강사
 연구 분야: 에이전트 시스템, 전문가 시스템, 지능형 교육 시스템, 인공 지능, 원격 협력 학습
 e-mail fish@eslab.inha.ac.kr

조 근 석

1982년 인하대학교 전자계산학과
 1985년 Queens Colleg/CUNY M.A., Computer Science
 1991년 City University of New York, Ph.D., Computer Science
 1988 - 1991 Logic Based Systems Lab. 연구원
 1991~현재 인하대학교 전자계산공학과 부교수
 연구 분야: 인공지능, CSP, 스케줄링 시스템, 에이전트 시스템, 전문가 시스템, 지능형 교육 시스템, 지능형 전자 상거래 시스템
 e-mail gsjo@dragon.inha.ac.kr