曉星論科集 第2卷
HSJAS : 31~37 (1993)

지능형 교수 시스템(ITS)을 위한 전문가 모듈의 연구

김 정 두 · 류 경 현
(효성여자대학교 자연대학 전자계산학과)

A Study Expertise Module for Intelligent Tutoring System

Kim, Jung-Doo and Ryu, Kyeong-Hyun
(Department of Computer Science, Hyosung Women's University)

Abstract

This Paper proposes an expertise module as Intelligent Debugger which is a component of Intelligent Tutoring System. The module organized with the programming knowledge of arithmetic expression in the Pascal language.

The knowledge representation of the module uses both frame-based method and semantic network, and it is displayed through semantic analysis process. The module has the functions that discovers misconceptions and bugs of a novice student's programs, and provides them the solutions for their corrections.

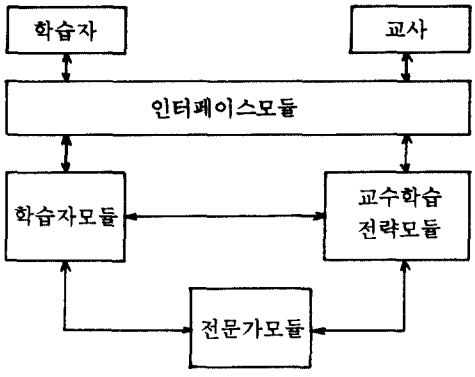
It is believed that repeated practice through ITS will help the novice students learn Pascal language more efficiently.

I. 서 론

60년대 Skinner의 행동 주의적 완전 학습 이론에 바탕을 두어 개발된 CAI(Computer Assisted Instruction)용 소프트 웨어(S/W)에서는 순차적/분기형(linear/branching) 반복 학습과 제한된 학습 반응에 따른 귀환(feedback)은 가능했으나 학생 개개인의 특성, 학생과 교사간의 대화 방법, 학습 곤란 요인의 진단등의 학습 시의 개인차를 고려한 개별 학습이 제공되지는 못하였다. 그래서 이러한 CAI의 한계성을 극복하기 위해 다양한 지식 표현(Knowledge Representation), 동적 학습자 모형의 구축, 추론(inference)에 의한 교수 방법 및 전략 수립, 자연 언어(Natural Language), 시각 정보 처리(Visual information processing)등의 인공 지능(Artificial Intelligence) 기법을 가미하여 개발된 것이 지능형 교수 시스템(Intelligent Tutoring System)이다.[1-10] 따라서 본 논문은 구조적 프로그래밍 언어인 Pascal에서 입력 변수(input variable)의 갯수에 따라 산술식(Arithmetic Expression)이 달라지는 예제 프로그램을 의미 분석(Semantic Analysis)하는 지능형 디버깅 시스템(Intelligent Debugging System)을 설계한다.

II. ITS의 개요

ITS는 학자의 견해에 따라 조금씩 다르기는 하나 〈그림 1〉과 같이 기본적으로 4가지 모듈로 구성된다. [11-15]



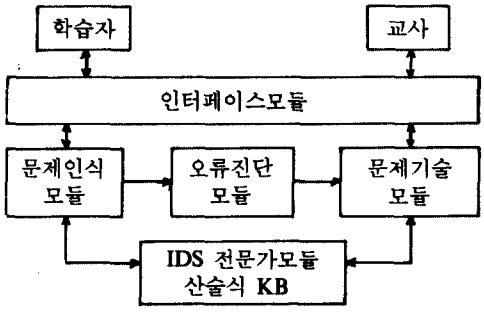
〈그림 1〉 일반적인 ITS의 구조

Ⅲ. IDS의 설계 및 구현

1. IDS의 특징

1) IDS의 구성 요소

IDS는 ITS의 기본 구성 요소를 바탕으로, 구조적 프로그래밍 언어인 Pascal의 산술식에서 사칙 연산에 대한 학습을 관리 운영하는 시스템을 말하며 ITS 중심의 문제 해결을 위한 IDS의 구조는 〈그림 2〉와 같다.



〈그림 2〉 IDS 시스템 구조

IDS의 특징은 전문가 모듈에 Pascal 언어의 산술식을 지식 베이스로 구축하는 것과 학습자의 잘못된 개념을 진단하는 오류 진단 모듈을 설계하는 것이다.

문제 인식 모듈은 입력 변수의 갯수에 따라 산술식이 달라지는 전문가 모듈에서 해결하고자 하는 문제의 한계를 특정 범위내로 한정시켜 문제 영역을 정의하는 곳이다.

문제 영역 : 입력 변수는 하나, 둘 또는 세 개 중 택일하고, 입력 변수가 9999를 만나면 끝나는 입력 처리를 한다. 또한 입력 변수의 부호가 음이면 다시 데이타를 읽어들이는 루틴을 작성한다. 그렇지 않으면, 읽어들인 변수값은 누적하고 카운터 변수를 사용하여 1씩 증가시켜 마지막에 평균을 구하고 그 값을 출력하시요.[10]

문제 기술 모듈은 교사가 제시한 문제에 대해 답안을 작성하는 것으로, 문제 기술(Problem Description)은 만족되어야만 하는 원리의 요구(Principal of Requirement)들로 표현되는데 이것은 프로그래밍 목표(Programming Goal)와 처리해야만 되는 데이타를 표현하는 오브젝트(Object)로 구성된다.

본 논문에서는 크게 세 가지로 구분하여 문제 기술을 하였는데 즉 입력 변수가 하나이고 연산자가 하나일 경우(ARITHA, ARITH1A, ARITH1AD, ARITH1S, ARITH1M, ARITH1D, ARITH1), 입력 변수가 두개이고 연산자가 하나일 경우(ARITH2A, ARITH23A, ARITH2S, ARITH2M, ARITH2MD, ARITH2D, ARITH2), 입력 변수가 세개이고 연산자가 두개일 경우(ARITH3A, ARITH3AD, ARITH3S, ARITH3M, ARITH3D, ARITH3)이다.

오류 진단 모듈은 학습자들이 작성한 예제 프로그램과 교사가 제시한 답안인 문제 기술을 의미 분석 과정을 통해 다양한 형태의 오류들을 발견하는 것으로, 이 모듈은 버그가 발생했을 때 단순히 목표와 프로그

램 소스 코드(Source Code)가 부합되지 않는다는 것을 설명하기 위해서 잘못된 코드 라인(Code Line)만을 제시해주는 것이 아니라 버그가 프로그램 어느 부분에서 발생했는가를 나타내 주고 버그를 어떻게 수정할 것인가를 진단해 준다.[25]

2) IDS의 기능

일반적으로 ITS는 모든 분야에 대해 적용되는 시스템이 아니라 주로 제한된 범위의 잘 구조화된 분야에 한정되어 개발되었는데 적용 영역별로 주제를 보면 다음과 같다.[1]

- 산술식에서 값의 계산
- 간단한 대수식의 해결
- 비결정적 문제 해결
- 디버깅 (전자 회로, 프로그램, 플랜)
- 의학 진단

다시 말하면, IDS의 문제 영역은 구조적 프로그래밍 언어인 Pascal에서 입력 변수의 갯수에 따라 산술식이 달라지는 예제 프로그램을 디버깅(Debugging)하는 문제다.

선택된 디버깅 적응 영역의 독특한 성질에 비추어 보아 강조되어야 할 주요 사항은 다음의 3가지로 요약될 수 있다.

1. 입력 변수의 갯수에 따라 산술식이 달라지는 지식 체계는 목표(Goal)와 계획(Plan)의 생성 및 실행으로서 서술된다. 결과적으로 전문가 모듈은 주로 목표와 계획의 추론에 있다.

2. 시스템은 디버깅 성격을 고려하여 교사가 제시한 문제에 대해 타당한 여러 가지 방법들을 인식해야 한다.

3. 학습자의 잘못된 개념을 진단해 주는 버그 데이타베이스(Database)를 작성해야 한다.

이러한 모든 사항들이 시스템에 설정되어야 한다.

2. 문제 해결을 위한 지식 획득

전문가로부터 어떤 방법으로 지식을 획득할 것인가 등을 결정한다. 전문가 모듈에서 지식 획득(Knowledge Acquisition)은 올바른 전문가 시스템을 구현하는데 필요한 가장 기초적인 사항이다. 일반적인 지식 획득 방법은 인터뷰(Interview), 프로토타입 분석(Prototype Analysis), 시뮬레이션 모형(Simulation Model), 문서, 영역 전문가(Domain Expert)와 연결된 소프트웨어(S/W) 지원을 통한 획득 방법이 있다.[19, 20]

본 논문에서는 이들 지식 획득 방법 중 문서에 의한 방법을 사용한다. 문서에 의한 방법은 문제 영역에 대한 문헌들을 통해 지식을 얻는 방법이다. 구조적 프로그래밍 언어인 Pascal에서 입력 변수의 갯수에 따라 산술식이 달라지는 예제 프로그램을 중심으로 전문가 모듈을 구축하기 위해 필요한 지식을 Pascal 교재[18], 자료 구조[21]에서 획득한다.

3. 문제 해결 전략

일반적으로 문제 해결 전략에는 전진 추론(Forward Chaining) 방식과 후진 추론(Backward Chaining) 방식이 있다. 전진 추론은 사실(Fact)로 부터 시작하여 목표(Goal)에 이르도록 하는 방법으로 Data-Driven, Bottom-Up 방식이라고도 하며, 주로 도달 가능한 목표의 수가 아주 많은 경우에 사용한다.

또한 후진 추론 방식은 Goal-Driven, Top-Down 방식이라고도 하는데 주로 목표의 수가 적고 진단이나 조정에 사용되는 제어 전략(Control Strategy)으로서, 특정한 목표의 참(True)과 거짓(False)을 검사할 수 있는 장점이 있다.[22, 23]

그러나 이들은 논리적으로 동일한 의미를 지니고 있다. 단지 이들이 언제, 어떻게 사용되느냐에 따라서 구분된다.[2]

4. 지식 베이스 구축

문서에 의해 획득한 지식을 Frame-Based 방식과 Semantic Network[22, 23]을 사용하여 학습자들이 학습해야 할 내용 즉 입력 변수의 갯수에 따른 입력 처리(INPUT PROCESSING), 입력 타당성(INPUT VALIDATION), 사칙 연산자에 의한 산술식, 누적(SUM), 평균(AVERAGE), 카운터(COUNT), 출력(OUTPUT), 버그 진단(GUARD-EXCEPTION)등을 지식 베이스로 구축한다.

먼저 목표 지식 베이스의 형식은 〈그림 3〉에 제시되어있다.

〈그림 3〉에서 DPS는 프레임 구조를 정의한 것이고, INSTANCES Slot의 속성은 목표 GOAL-NAME의 속성 리스트(Attribute List)에 추가되어 리스트들의 속성을 나타낸다는 의미다.

다음 예는 입력 변수가 두 개 일 때 목표 입력 처리2의 프레임 구조를 제시해 놓았다.

〈그림 4〉의 목표 SENTINEL-CONTROLLED-LOOP-TWO는 1개의 INSTANCES 슬롯과 4개의 필러로 구성되는데 이들 각 필러는 또한 계획의 이름(PLAN-NAME)이 되기도 하는 계층적 구조를 가지고 있다.

다음 계획은 목표를 추론하기 위해 만들어진 데이타 세트(Data Set)로 판에 박힌 문구를 기리킨다. 이것도 각각에 대해 슬롯과 필러로 정의된다.

계획 지식 베이스의 형식을 살펴보면 다음 〈그림 5〉와 같다.

다음 예는 입력 변수가 두 개 일 때 목표 입력 처리2의 4가지 필러중의 하나인 SENTINEL-READ-PROCESS-WHILE2 을 TEMPLATE 슬롯과 내용 기술로 정의해 놓았다.

```
(DPS GOAL-NAME
     INSTANCES (PLAN-NAMES)
     NAME-PHRASE "GOAL이 하는 일이 무엇인가를 기술")
```

〈그림 3〉 목표 정의

```
(DPS SENTINEL-CONTROLLED-LOOP-TWO
     INSTANCES (SENTINEL-READ-PROCESS-WHILE2
                SENTINEL-PROCESS-READ-WHILE2
                SENTINEL-READ-PROCESS-REPEAT2
                SENTINEL-PROCESS-READ-REPEAT2)
     NAME-PHRASE "INPUT PROCESSING-TWO")
```

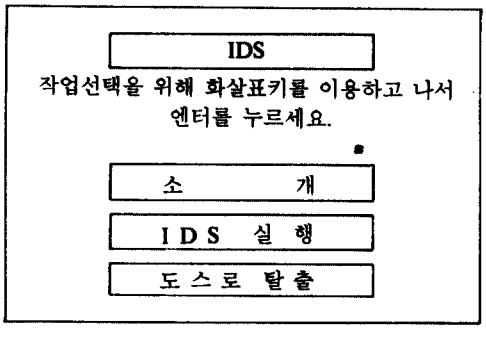
〈그림 4〉 입력 변수가 두 개일 경우의 입력 처리 목표 정의

```
(DPS PLAN-NAME
     TEMPLATE (CONTEXT DESCRIPTION))
```

〈그림 5〉 계획 정의

```
(DPS SENTINEL-READ-PROCESS-WHILE2
 TEMPLATE
 (array ((MATCH (WHILE (⟨⟩ (*VAR* NEW) (*VAR* STOP)) (*VAR* ?)) ())
 (LABEL MAINLOOP: ((AT 1)))
 (MATCH (BEGIN (*VAR* *)) ((AT 1)))
 (MATCH (READ (*VAR* NEW) (*VAR* NEW1)) ((AT 3) (TOP)))
 (LABEL NEXT: ((AT 4)))
 (MATCH (IF (⟨⟩ (*VAR* NEW) (*VAR* STOP)) (*VAR* ?)) ((AT 3) (BELOW 4)))
 (LABEL PROCESS: ((AT 6)))
 (LABEL INTERNAL-GUARD: ((AT 6)))
 (MATCH (:= (*VAR* NEW) (*VAR* SEEDVAL)) ((ABOVE 1)))
 (LABEL INIT: ((AT 9)))) (10)))
```

〈그림 6〉 목표 입력 처리2의 필러 4가지 중 하나의 계획 정의



〈그림 7〉 IDS의 초기 화면

〈그림 6〉의 TEMPLATE 슬롯은 계획에 따라 구현해야 할 Pascal 코드의 형태를 기술해 놓았는데 이것은 파스칼 문장(Pascal Statement), 하위 목표(Subgoal), 레이블(Label)로 구성된다.

파스칼 문장은 원래의 파스칼 문법으로 표현하기 보다는 리스트(List)로 작성되어있다. 예를 들면 WHILE NEW 〈 〉 STOP DO 를 리스트 표현으로 작성하면 다음과 같다.

(WHILE (⟨⟩ (*VAR* NEW) (*VAR* STOP)) (*VAR*?))

여기서 *VAR* NEW 와 *VAR* STOP 은 전역 변수(Global Variable)로써 전자는 입력 데이타를 포함하는 파스칼 변수로 바뀌지고 후자는 상수로 바뀌진다. 그리고 *VAR* ? 와 *VAR* * 은 와일드 카드(Wild-Card) 패턴으로 임의의 연속된 파스칼 문장으로 바뀌진다. 하위 목표는 TEMPLATE 슬롯에서 사용되는것으로 내용에 따라 번갈아 다른 목표들을 사용하여 수행한다. 예를 들면 다음과 같다.

(READ (*VAR* NEW) (*VAR* NEW1))

레이블은 그 스탭을 포함하는 루틴이나 서브루틴에서 엔트리의 참조점으로 사용되는 이름이다. 문제 영역에 사용되어진 IDS 전문가 모듈의 지식 체계는 목표 프레임 25개와 계획 프레임 65개로 구성되어 있다.

5. IDS의 구현

IDS는 의미 분석 과정을 통해 전달되는 버그 메세지를 옴니 CGA 한글 카드를 이용하여 처리하였으며 다음 〈그림 7〉은 IDS을 운영하였을 때의 초기 화면을 제시한다.

결 론

본 논문은 입력 변수의 갯수에 따라 산술식이 달라지는 Pascal 언어의 예제 프로그

램을 디버깅하는 전문가 모듈인 지능형 디버깅 시스템(IDS)을 설계하였다.

본 시스템은, 지식 획득을 Pascal과 자료구조 교재를 통해서 획득하였고, 학습 내용을 Frame-Based 방식과 Semantic Network를 사용하여 지식 베이스로 구축했다. 또한 이것은 IBM PC/AT MULISP 한글 환경하에서 구현되어졌다. IDS의 기능은 학습자의 프로그램과 교사의 답안인 문제 기술을 의미 분석 과정을 통해 학습자의 잘못된 개념과 버그를 발견해 줄 뿐 아니라, 그것에 대한 해결 방법도 제시해 준다.

따라서 초보 학습자들이 IDS를 통해 반복 연습함으로써 Pascal 언어를 보다 효율적으로 학습할 수 있으리라 생각된다.

앞으로의 연구 과제는 IDS에서 수행되는 전문가 모듈의 범위가 한정되므로 이를 Pascal 언어의 제어문(Control Statement), 반복문(Iteration Statement), 조건문(Condition Statement) 등의 분야까지 확장하여 활용될 수 있게 하는 것 뿐만 아니라 학습자들의 잘못된 개념을 조언해주는 버그 데이타베이스를 학습자가 의도하는 지식과 코드로 잘 부합시켜 설계하는 것이다.

참 고 문 헌

1. D. Sleeman, J. S. Brown, Intelligent Tutoring System, Academic Press, Inc(1982).
2. Greg Kearsley, Artificial Intelligence & Instruction, Addison-wesley Pub(1987).
3. 溝口文雄, 多段階 教授 最適化, 情報 處理, Vol. 18, No. 11, pp. 1130－1137(1977).
4. 豊田順一, 中村析一, 知的 CAI 知識 表現 教授法, 情報 處理, Vol. 29, No. 11, pp. 1266－1273(1988).
5. Jeff W. Rickel, Intelligent Computer-Assisted Instruction : A Survey Organized Aroung System Component, IEEE Transactions on systems, Man and Cybernatice, Vol. 19, No. 1, pp. 40－57(1989).
6. Franklin C. Roberts and Ok-Chon Park, Intelligent Computer-Assisted Instruction : An Explanation and Overview, Educational Technology, pp. 7－12(1983).
7. 山本秀樹, 甲裴 鄕子, 大里 莫理子, 椎野努, 語學 訓練用 知的 CAI 學習者 意圖 把握 會話制御方式, 情報處理學會論文集, Vol. 31, No. 6(1990).
8. 西田 富士夫, 高松忍, 谷忠明, 日下浩次, 知的 CAI 用 說明 理解 情報抽出 ,情報處理學會論文誌,Vol. 31,No. 3,pp. 481-489(1990).
9. 中山和彦, CAI를 위한 준비, 科學教育, 通卷 314虎, pp. 32－54(1990).
10. 溝口文雄, 佐伯畔, CAI 教授 論理 學習者 意志決定 機構, 情報 處理, Vol. 15, N0. 2, pp. 101－109(1974).
11. 김 세엽, 정 용득, 기하 증명 교육을 위한 지적 CAI의 적용에 관한 연구, 정보 과학회 추계 전산교육 논문집, pp. 42－52(1990).
12. 김 형수, 노 환주, 양 해술, 기하 논증 학습을 위한 ICAI 시스템, 정보 과학회, Vol. 17, N0. 2, pp. 613－616(1990).
13. 오 선영, 곽 덕훈, 백 두권, 황 종선, 학생 모델 구축을 위한 ITS의 설계 에 관한 연구, 정보 과학회, Vol. 16, No. 2, pp. 525－528(1989).
14. 임 기영, 김 형래, 지능형 튜터링 시스템 실행에 관한 연구, 전자공학회지 논문, 제 27권 제 9호, pp. 1372－1377(1990).
15. 최 영미, ITS의 수식 생성 최적화 기법 ,정보 과학회,Vol. 17,No. 2, pp. 593－595(1990).
16. 이 옥화, 교육용 소프트웨어(CAI)의 개발과 인공지능의 역할, 춘계 전산 교육 심포지움 발표 논문집(1989).
17. 이 수현, 상호 협조적인 과제들을 위한 ICAI 기본구조의 확장, 추계 전 산교육 학술 발표회 발표 논문집(1989).
18. Elliot D. Koffman, Problem Solving and Structured Programming in Pascal, second edition, 연합 출판(1985).
19. Riichiro Mizoguchi, Osamu Kakusho, Knowledge Acqusition Systems, 大阪大學産業科學研究所, Vol. 3, No. 6(1988).
20. John K. Debenham, Knowledge Systems Design, Prentice Hall Advances in computer science series(1989).

21. 金 慶泰, 閔 勇植 共著, 資料 構造, 裕豊出版社 (1987).
22. Edward R . Dougherty, Charles R. Glardina, Mathematical methods for Artificial Intelligence and Autonomous systems, prentice－Houl international, Inc(1988).
23. Eugene Charniak, Drew Mcdermott, Introduc tion to Artificial Intelligence, Addison wesley (1987).
24. Harald Wertz, Streotyped Programming Debugging : an aids for novice programmers, Int. J. Man-Machine Studies, 16,pp. 379－392 (1982).
25. Lloyd D. Fosdick, Leon J. Osterwell, Data Flow Analysis in Software Reliability, ACM, Vol. 8, No. 3, pp. 305－330(1976).
26. 최 인현, 최 인열 공저, mulisp 언어입문, 대림 (1990).
27. Microsoft, Mulisp reference memual, soft warehouse(1983).
28. Winston, Horn, LISP 3rd edition, Addisonwesley Pub(1989).
29. 한 애경, 저작 시스템 IIAS의 연구 분석과 코스의 구현, 고려대학교 교육 대학원(1990).
30. 홍 성옥, 곽 덕훈, 백 두권, 황 종선, 학습자 반응 시간을 고려한 ICAI 시스템(T-BUGGY)의 설계, 정보 과학히 Vol 18, No. 1, pp. 81－84 (1991).