

지능형 튜터링 시스템 실행에 관한 연구

(A Study of on the Implementation of an Intelligent Tutoring System)

林 岐 榮*, 金 炯 來**

(Gi Young Lim and Hyung Rae Kim)

要 約

지능형 tutoring system(ICAI)을 intelligent interface, acquisition module, tutoring controller, tutorial knowledge의 4부분으로 구성하여 prolog을 저자 언어로 사용하여 ICAI을 VAX-11/750 ultrix os아래 실행시켰다. 또한 목표지향형 tutoring system에 사용할 알고리즘을 제안하였다.

Abstract

We constructed an intelligent tutoring system that compose four devices i.e. intelligent interface, acquisition module, tutoring controller and tutorial knowledge.

We implemented the intelligent tutoring system using prolog as an authoring language in the ultrix os under VAX-11/750, and we propose goal oriented tutoring system algorithm.

I. 서 론

1960년에 IBM에서 이진법을 가르치기 위한 CAI 프로그램을 개발해 컴퓨터가 교육용으로 이용될 수 있다는 사실이 입증된 이후로 교육에의 컴퓨터적용은 계속 연구되어 왔다.

CAI¹⁾ (computer assisted instruction 혹은 CBL : computer based learning이라고도 함)이란, 학습자와 컴퓨터의 상호작용으로 컴퓨터에 의해 학습내용이 체계적으로 진행되는 것을 뜻한다. CAI에 의한 학습은 학습자 개인의 능력과 특성에 맞게 개별 학습을 제공해 준다. 개별학습이란, 학습자가 학습목표에 도달하기 위해 학습자의 개인차를 고려해서 자신의 학습속도로 학습을 진행할 수 있도록 도와주는

것을 뜻한다. 그러므로 CAI는 개별학습을 통해 학습시간을 단축시켜 주고, 학습자에게 보다더 적극적으로 학습에 참여하도록 유도함으로써, 어떤 학습방법보다 가장 효율적인 학습을 제공해 줄 수 있다. 1980년대에 접어들면서 활발하게 연구되어온 인공지능(A.I)이 CAI(컴퓨터 보조학습)에 응용되기 시작하면서 CAI 연구개발은 새로운 전기를 맞게 되었다.

따라서 AI를 이용하여 학습과 교수과정에 응용하려는 접근으로부터 ICAI(intelligent computer assisted instruction : 지능형 컴퓨터 보조학습)가 세계적 관심분야로 등장하게 되었다.

이러한 ICAI의 성능을 수행할 수 있도록 AI 언어인 prolog를 저자언어로 사용하여, 지능형(intelligent) tutoring system을 실행시켜 보고자 한다.

II. 지능형 튜터링 시스템의 개요

ICAI의 접근은 네개의 모듈로서 구성된다. 즉, 전문가모듈, 학습교사모듈, 학생모듈, 그리고 인터페

*正會員, 暎園專門大學 電子計算科

(Dept. of Computer Eng., Kyeongwon College)

**正會員, 建國大學校 電子工學科

(Dept. of Elec. Eng., Kunkuk Univ.)

接受日字: 1990年 2月 19日

이스모듈로 구성된다. 전문가 모듈은 배워야 될 모든 학습내용과 학생의 학업성취 정도를 평가하기 위한 학습내용, 전문가의 지식 및 문제해결 절차가 이 모듈속에 담겨져 있다. 학생모듈에서는 학생의 학업성취정도 및 그 심취과정에서 나타낸 학습행위, 학습과정에서 학생이 범할 수 있는 모든 가능한 오류들, 그리고 학생의 학습상 문제점과 필요를 진단하고 처방하는데 고려해야 되는 개인차 변인들을 이 모듈속에 포함시킨다. 학습교사 모듈은 다양한 종류의 학습전략내용, 학생의 학습상 문제점과 필요를 적절한 학습내용을 처방하는데 필요한 규칙들이 이 속에 담겨져 있다. 인터페이스모듈은 학생과 시스템간의 인터페이스에 필요한 학생의 정보입력을 이해하고, 학생에게 학습내용을 전달시켜 줄 수 있는 기술적 성능을 찾고 있다.

이러한 ICAI를 행하는 시스템을 지능형 tutoring system³⁾이라 하고 (그림 1)과 같이 구성된다.

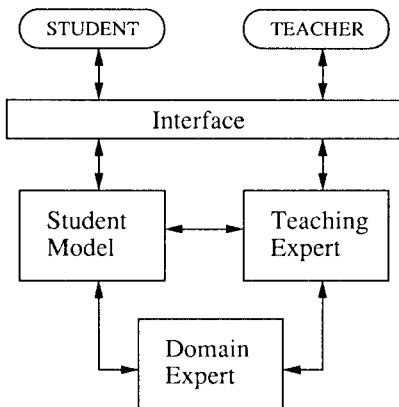


그림 1. 일반적 지능형 튜터링 시스템 구성
Fig. 1. General intelligent tutoring system.

1. Prolog을 사용한 지능형 튜터링 시스템

Prolog는 자연언어의 이해, relation data base 사용과 수학적인 논리를 취급하는 등 인공지능 분야에 서 광범위하게 사용되고 있는 언어로써 1970년초 marseille-aix 대학의 정리 증평기로서 자연어 처리 시스템을 구현하는데 사용하였다.

Programationet logique의 앞사를 따서 prolog라 불리지었다. prolog의 구성은 사물(object)이거나 그들간의 객관적인 관계를 선언하는 사실(fact) 사물과 그들간의 수학적인 관계를 정의하는 규칙사물과 그들간의 관계를 묻는 질문등으로 구성된다.

사실이나 규칙을 prolog에게 마련하였다면 prolog에서 질문을 할 수 있다. 질문의 양식은 사실의 양식과 같으며 다만 사실앞에 “?”의 부호를 부치고 있다.

일반적으로 prolog는 질문을 받으면 전문내용을 D.B(data base)에서 찾게된다. 질문의 사실과 부합되면 “Yes”라고 반응하고 그와 같은 사실이 D.B(data base)안에 있지 않으면 “no”라고 대답한다.

Prolog에서는 단어대신 행(clause)을 사용하는 형태로서 각행은 horn clause라고 불리우는 형을 하고 있다. horn은 보통 statement로 나타내기도 한다.

따라서 다음과 같은 prolog horn clause 인

Sort 1([X|Y], X, Z) :- ! sort(X, Z)가 있다면 다음과 같은 안내를 생성시킬 수 있는 튜터링 시스템을 구현하고자 한다.

Guide :

Sort 1 a list with head X and tail Y

2. 지능형 튜터링시스템의 구성

본 연구에서는 prolog를 사용한 지능형 tutoring system⁴⁾을 다음 (그림 2)과 같이 4 부분으로 구성 하였다.

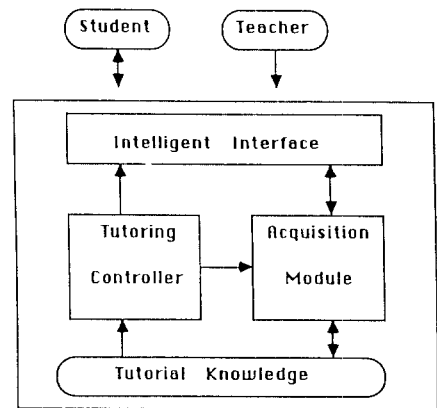


그림 2. Prolog를 사용한 지능형 튜터링 시스템
Fig. 2. Intelligent tutoring system using prolog as an authoring language.

1) 지능형(Intelligent) interface

이 부분은 사용자의 인터페이스역할을 한다. 이 부분을 잘 사용함으로써 사용자는 ICAI system과 친숙하여지므로 지능형 인터페이스라 하였다. 요즘 많이 사용되는 window system과 VIPS(visual prog-

ramming synthesizer)⁵⁾을 사용함으로써 학습효과를 높인다.

Window system은 컴퓨터에서 실행하는 각종 상이한 작업들을 동시에 작업할 수 있다. 몇개의 키 조작이나 mouse(마우스)를 이용하여 응용 프로그램에서 다른 응용 프로그램으로 옮기는데 소요되는 시간을 절약할 수 있다.

2) Acquisition modular⁶⁾

이 시스템은 기존의 지식에 새로운 정보를 첨가하는 부분으로서, 이 모듈을 사용함으로써 교사가 가르칠 코스웨어를 내장시킨다. 내장된 코스웨어란 학습을 실험할 수 있도록 직접학습에 관련된 소프트웨어이다. 이것은 저장된 영역의 코스웨어의 내용이 중복되거나 일치하지 않으면 체크하는 것을 가능하게 해야 한다. 또, 프로그램 작성능력을 배양하는 학생행동을 실현시키기 위해 사용된다. 학생이 코스웨어를 실행시 틀린 부분이 발생하면 tutorial knowledge로 부터 이를 수정하기 위해 이 모델을 이용한다. 이 모델을 잘 사용함으로써 tutoring system을 효율이 좋은 지능형으로 만들 수 있다.

3) Tutoring controller

Tutor의 중요한 부분으로 tutorial knowledge로 부터 정보를 학생과 시스템 사이에서 상호 조정하는데 사용한다. 학생에게 필요한 코스웨어를 나타내고 요구할 수 있다.

- 항상 학생의 시도를 구성하고 나타낸다.
- Tutorial knowledge로부터 적당한 힌트와 예를 전송한다.
- 학생에게 goal-oriented feed back을 제공해 준다.
- 학생이 tutor의 반응을 잘 모를때 help에 대해 설명을 요청할 수 있다. tutoring controller는 코스웨어의 흐름을 제어한다.
- Intelligent interface와 acquisition module을 통하여 정의된 코스웨어를 조정한다.
- Intelligent interface를 통하여 주어진 topic을 시도하고 학생의 행동과 나타남을 번역하기 위해 tutorial knowledge를 요구한다.

4) Tutorial knowledge

필요한 지식을 저장할 tutor의 내부의 K. B(knowledge base)이다. 가르칠 내용에 대한 정보와 어떻게 가르칠 것이다 등을 포함하고 있다. 이것은 힌트, 예, 그리고 코스웨어를 포함한다.

Tutoring knowledge는 해를 구하는(solution)모듈과 수정하는(debugging)모듈 2개의 부분으로 나타낼 수 있다.

(1) Solution 모듈

이 모듈은 문제해결을 위한 tutor의 해결 방법이다. Solution 모듈은 각각의 문제해결을 위한 solution을 포함하는데, tutor는 어떻게 문제를 해결할 것인가를 학생에게 안내하는 계층적 구조 형태(hierarchical frame)에 따라 계획을 생성한다.

Tutor는 학생이 도움을 요청할 때 내장된 정답인 해와 현재의 해를 비교하는 과정을 반복하여 문제를 해결하여 간다.

계층적 구조형태에서 G 프레임은 문제(problem)이고 I 프레임은 해(solution)을 나타낸다. 그러므로 G 프레임은 I 프레임들(그림3)과 같이 생성시킨다.

(2) Debugging 모듈

이 모듈은 shapiro의 "algorithmic program debugging"⁷⁾의 형태를 도입한 것으로 debugging 모듈은 우리가 필요로 하는 정교함을 만들기 위해 수행능력을 개선하기 위한 것이다. 학생이 쉽게 프로그램을 배울 수 있도록 tutor가 error를 진단하고 정답을 쉽게 할 수 있도록 프로그램이 내장되어 있어야 한다.

- 현재목표에 error가 나오면 tutor는 틀린 프로그램 추적의 설명과 함께, 왜 틀렸는가를 학생에게 알려는 진단 메시지를 내보내야 한다. 학생이 도움을 요청하면 tutor는 그를 도와주어 error가 나오지 않게 안내한다.

* Tutoring 알고리즘은 다음과 같다.

ALGORITHM:TUTORING

```

Repeat
  IF student can't continue to program he requests for help
  THEN
    IF student's behavior is anticipated
    THEN
      tutor provides guidance:
      GUIDING(student solution, ideal solution):
    IF student's behavior is unanticipated
    THEN
      tutor helps student to synthesize program:
      Debugging Module(student solution, query):
    ELSE
      CHECK BEHAVIOR(ideal solution, current diagnosing goal):
      IF behavior is anticipated
      THEN
        ideal solution ← TRANS(ideal solution)
        DELETE current ideal goal from ideal solution
        ADD current diagnosing goal to ideal solution
        Debugging Module(ideal solution, tutor):
      IF behavior is unanticipated
      THEN
        student may ask tutor for diagnosis:
        Debugging Module(student solution, query):
    UNTIL student solution is proved to be correct:
    Solution Module(← knowledge Acquisition(student solution):
  
```

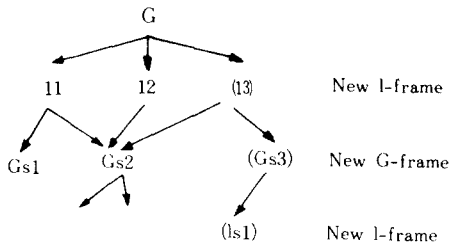


그림 3. 새로운 G-frame, I-frame 생성
Fig. 3. New G-frame, I-frame.

3. 목표지향형 튜터링 구성 (goal-oriented tutoring algorithm)

Tutorial knowledge에서 수정모듈(debugging module) tutoring 알고리즘을 먼저 「Top goal」지정하는 목표지향형 (goal-oriented) tutoring 알고리즘으로 바꾸어 보 기위해 다음과 같은 알고리즘을 제안한다.

ALGORITHM:TUTORING

```
Select top goal as the current goal;
REPEAT
  IF student can't continue to program, he requests for help
  {
    tutor provides declarative guidance to show goal planning
    and clause guidance;
  }
  substitute all the unsolved subgoals by ideal solution;
  invoke debugging module to help the student debug errors;
IF the current goal contains unanticipated goals
{
  tutor helps student to synthesize program by interactive query
  the necessary question;
  student may ask tutor for diagnosis;
  IF the current goal is proved to be correct
  {
    select next unsolved goal as the current goal;
  }
}
UNTIL no goal left to be solved
```

Shapiro의 튜터링은 학생이 도움을 요청하면 안내를 제공하여 준다. 만일 학생이 문제를 풀 수 없으면 문제를 분석하여 학생을 도와준다. 그리고 학생이 문제를 해결했으면 해결된 문제를 점검해 나간다. 학생의 해와 내장되어 있는 이상적인 해가 일치하는가를 확인하고 일치하지 않으면 수정 모듈에 의해 문제를 해결해 나간다.

학생의 문제해결이 근접할 때까지 이러한 과정을

계속 반복해 나간다.

이에 비해 목표지향형 튜터링 방식은 교사에 의해 먼저 목표를 설정해서 단계적으로 해결하는 방법으로 shapiro의 튜터링은 학생 스스로가 문제해결을 해 나가도록 하는데 비하여 목표지향형 튜터링은 교사에 의해 목표를 설정하여 학생의 문제해결을 해 나가는 방법이다. 따라서 현재 학생의 해 (solation)과 이상적인 해 (ideal solution)와 동일한가에 따라 학생을 평가하여 이상적인 해를 만든다.

이에 대한 흐름도는 다음과 같다.

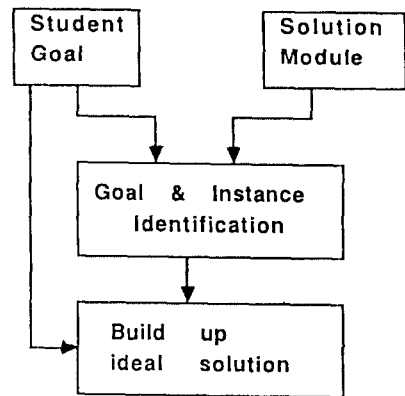


그림 4. 이상적 해의 구성 흐름도
Fig. 4. Build up ideal solution.

Ⅲ. 튜터링 시스템의 설계

튜터링 시스템 K. B (knowledge base)를 이용하여 내장된 교사의 course와 학생의 course가 일치되도록 지시하여 주는 역할을 할 수 있도록 다음과 같은 부분으로 구성되어 있어야 한다.

- a. Knowledge accommodation (domain expert)
- b. Build-course for teachers (teaching expert)
- c. User's-exercise for students (student model)

1. Knowledge accommodation module은 학생이 배워야 할 예를 K. B안에 공급하여 학습내용과 문제해결절차와 기준이 중점이 되는 전문가 모듈 (domain expert)와 같은 역할을 한다.

이 모듈은 (KBN, IC, concept) ? 의 형식을 취하여 새로운 개념을 K. B에 저장하고 검토한다.

여기서 KBN은 K. B name이고 IC의 구성은 다음과 같다.

(IC) ::= inconstraint ((Actions), (concept)) :
 - (condition)
 (concept) ::= (Goal)
 (condition) ::= (Goals)
 (Goals) ::= (Goal), (Goals) | (Goal)
 (action) ::= ...
 (그러나 tutor IC는 항상 []의 set이다)
 예) accommodate K.B.[]. sort(—, —)?
 K.B는 다음과 같은 형으로 표현된다.
 Knowledge-Base-name (rule)
 :
 Knowledge-Base name (rule)

2. Build-Course for teachers

교사는 사용될 예제 program을 K.B안에 내장된 course를 탐색하여 course module을 만들거나 text-edit를 통하여 보충, 삭제하거나 다시 만들어 사용할 수도 있다.

이 모듈의 형식은

:build-course(course-name, KBN) ?

교사는 내장된 K.B에서 KBN(knowledge base name)에 의해 training course와 course name을 사용한다. 만약 K.B안에 build-course가 없다면 accommodation module을 이용하여 사용될 course를 load 시킨다. 이 때 학습의 목적을 나타낼 top goal를 교사는 사용할 수 있다.

3) User's exercise for students

학생은 교사에 대해 세워진 course로 부터 배운다.

Exercise(course-name, student program)

학생의 program은 student program에 저장된다. 만약 학생의 행동의 예상되면 tutor module은 학생의 현재 진단 goal을 찾아낸다. 그렇지 않으면 accommodation module을 사용한다. 교사에 의해 정의된 top goal을 정의된 find 명령에 의해 찾아본다. 따라서 교사가 요구하고자 하는 사항을 미리 살펴볼 수 있다. find에 의해 나타난 사항은 끝나기를 원하면 OK를 key in 하고 아니면 help를 key in 한다.

a. Define top-goal: ?

find를 이용해 fact를 다음과 같은 form으로 응답한다.

Predicate-name(—, ..., —)

b. Input the clause for predicate name(—, ..., —)

더이상 입력시킬 절이 없을때 ok를 입력한다.
 어떤 도움을 받기를 원할때 help의 h를 입력한다.

c. Argument type of predicate-name(...)?

다음과 같은 형식으로 응답한다.

ai, ..., an.에서 n은 인수의 수이다.

ai = X는 atom을 나타낸다.

d. add(Y)a clause, (h)elp?

다할 절(clause)이 있으면 “Y”그렇지 않으면 “r”를 key in 한다.

e. Retract(Y) or (r)eplace a clause, (h)elp?

retract을 원하면 “Y”를 key in 한다.

replace을 원하면 “r”를 key in 한다.

도움을 원하면 “h”를 key in 한다.

f. Which?

d와 e에서 절의어에서 “Y”나 “r”가 나오면 해당하는 절을 입력한다.

g. Try again, or (h)elp

계속할 것인가, 도움이 필요한가의 필요한 항목 선택한다.

IV. 튜터링 시스템의 실행

System에 입력후 usr/users/ai/shapiro/의 current directory를/cd/shapiro/acgui dir 바꾼다.

왜냐하면 저장되어 있는 course ware인 tutor를 다음과 같은 형식으로 불러 들인다.

prolog-S100000 load, tutor

그 다음에 실행 시킨다는 명령이 나오면 accommodation module에 의해 저장된 course를 불러준다.

다음 그림5는 prolog을 가르킬 tutoring system 실행결과의 예로서 학생과 교사의 역할을 할 tutor 그리고 top goal로서 fact와 clauses은 multi-window system을 써서 나타낸 것이다.

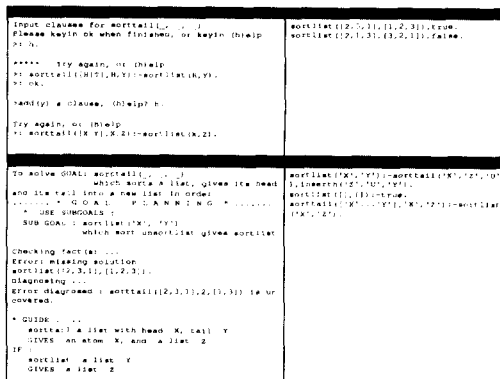


그림 5. Tutoring system 실행 예

Fig. 5. Example of implementation of an intelligent tutoring system.

V. 결 론

ICAI(intelligent computer assisted instruction) 의 접근은 전문가 모듈, 학습교사 모듈(teaching expert) 학생모델(student model) 그리고 인터페이스 모듈로 구성된다.

본 논문에서 prolog을 가르칠 지능형 tutoring system을

- 1) Intelligent Interface
- 2) Acquisition module
- 3) Tutoring controller
- 4) Tutorial knowledge의 4부분으로 구성하여 VAX-11/750 ultrix os 아래 UNSW prolog^h로 실행시켰다.

또한 Tutorial knowledge에서 목표 지향형(goal-oriented) tutoring system을 사용하기 위한 알고리즘을 제안하였다.

參 考 文 獻

[1] Atkinson. R.C and Wilson H.A 1968. "Computer assisted instruction," Science 162. 73

[2] Rich "Artificial intelligence," 1980.

[3] Yazdani, M: "Intelligent tutoring systems:

An overview," Expert System, July 1986. vol. 3, no. 3. pp. 154-162.

[4] Hsu, C.C., Lin, J.S. and Liu, D.R.: "An Intelligent Tutor for PROLOG Programming," Proceedings of ICS-86, Tainan, Taiwan, R.O.C. 1986, pp. 444-451.

[5] Cheng, K.Y., Hsu, C.C., Lin, I.O., Lu, M.C. and Hwu, M.S.: "Vips *A Visual Programming Synthesizer," The 2nd IEEE Computer society workshop on Visual Language, Dallas, Texas, June, pp. 92-98, 1986.

[6] Kitakami, H., Kunifuji, S., Miyachi, T. and Furukawa, K.: "A Methodology for Implementation of a Knowledge Acquisition System," International Symposium on Logic Programming, Atlantic City, New Jersey, pp. 131-142, 1984.

[7] Shapiro, E.Y.: "Algorithmic Program Debugging," PhD thesis. ACM Distinguished Dissertation, 1982.

[8] Sammut, C.: UNSW PROLOG USER MANUAL, University of New South Walse, October, 1980.

[9] 임기영, 김형래 "지능형 튜터링 시스템 실행에 관한 연구" 1989년도 대한전자공학회 추계종합 학술대회 논문집 p. 596-598. 11. 18, 1989.

著 者 紹 介



林 岐 榮 (正會員)

1953年 7月 12日生. 1977年 2月 건국대학교 전자공학과 졸업 (공학사). 1980年 2月 건국대학원 전자공학과 졸업 (공학석사). 1989年 8月 건국대학원 전자공학과 박사과정수로. 1981年 12月~1982年 8月 일본 在賀大學 객원연구원. 1984年 5月~1984年 8月 미국 Geogia Tech. 컴퓨터 연수. 1987年 9月~1988年 8月 국립 대만대학교 전자계산소 연구원. 1980年 3月~1990年 현재 경원전문대학 전자계산과 교수. 주관심분야는 Expert system, neural network 등임.



金 炯 來 (正會員)

1947年 8月 27日生. 1966年 3月~1970年 2月 연세대학교 이공대학 전기공학과. 1970年 3月~1972年 2月 연세대학교 대학원 전기공학과(공학석사). 1972年 3月~1980年 2月 연세대학교 대학원 전기공학과(공학박사). 1975年 5月~현재 건국대학교 공과대학 전자공학과 교수. 1982年 7月~1984年 2月 미국 Northwestern대학교 교환교수. 1988年 3月~현재 건국대학교 공과대학 학장보검 교학과장