

논리적 사고력 함양을 위한 교육용 한글 프롤로그 언어의 사용자 인터페이스 설계에 관한 연구

송태옥[†] · 안성훈[†] · 김태영^{†*}

요약

컴퓨터 교육과정에 있어서 프로그래밍 교육의 필요성에 대해 많은 논란이 있지만, 프로그래밍 교육이 논리적 사고력 함양에 효과적이라는 것은 분명한 사실이다. 기존에 많이 이용되었던 BASIC 언어는 교육용 언어로서 부족한 점이 많다. 그러므로 프롤로그와 로고, GUI의 장점을 조화시켜 효과적인 프로그래밍 환경을 구축하여 논리적 사고력을 함양할 수 있는 교육용 한글 Prolog 언어의 사용자 인터페이스를 설계하고자 한다.

Design of an User Interface of Educational Hanguk Prolog Language(EHPL) for Improving the Ability of Problem Solving

Tae-ok Song[†] · Sung-hoon Ahn[†] · Tae-young Kim^{†*}

ABSTRACT

In computer education curriculum, there are many different opinions about the needs of programming education, but it is obvious that the effect of programming is positive to increase the ability of problem solving. Programming languages such as BASIC have several problems as an educational programming language. Therefore, in this research, we designed an user interface of Educational Hanguk Prolog Language (EHPL), which has an ability of problem solving by employing the advantages of LOGO, Prolog and graphic user interface(GUI). Thus, EHPL has more effective learning environment than the one without GUI.

1. 서론

현대 사회는 거의 모든 분야에서 컴퓨터가 중요한 역할을 담당하는 정보화사회로 접어들었다. 이러한 사회적 변화와 더불어 학교 교육에 대한 기대에도 많은 변화들이 있는데, 그 중에는 정보소양을 기르는 것 외에도 사고력이나 탐구력, 창의성과 문제해결력을 길러주는데 학교 교육이 제 몫을 해야 한다는 인식도 포함되어있다. 컴퓨터 프로그램은 이러한 기대에 부응할 수 있는 도구

로 알려져 있다[1].

컴퓨터 소양을 기르고 컴퓨터 활용 교육을 확산함으로써 정보화사회에 적절히 대응할 수 있는 자질을 기르기 위한 컴퓨터 교육의 일반적인 목표는 세 가지를 들 수 있는데[2], 그 중 논리적 사고력과 문제 해결력을 기르는 목표를 달성하는데 있어서 유용한 교육적 도구의 하나는 교육용 언어를 통한 프로그래밍이다.

프로그래밍은 프로그램을 설계하고 구현하는 일련의 과정이 Polya (1957)가 주장한 문제 해결 과정과 유사한 점이 많기 때문에 문제 해결의 한 중요한 유형으로 논의되어 왔다[3].

문제해결의 한 유형으로서, 프로그래밍의 중요성은 프로그래밍 언어 습득보다는 프로그래밍을

[†] 정 회 원: 한국교원대학교 컴퓨터교육과 박사과정
^{†*} 중 심 회 원: 한국교원대학교 컴퓨터교육과 조교수
논문접수: 1998년 11월 15일 심사완료: 1998년 12월 16일

통해서 학습자의 고등인지기술을 향상시킬 수 있다는 것에 있다. 즉, 코딩 과정에서는 논리적 사고력을, 오류 검증 및 수정작업에서는 반성적 사고 능력을 향상시킬 수 있다.

이러한 프로그래밍의 중요성에도 불구하고 일반 교육과정 속에 프로그래밍이 포함되어야 하는가에 대한 문제는 많은 논란의 대상이 되고 있다 [4][5][6]. 반대 의견 중에는 프로그래밍이란 직업적 기술이기 때문에 일반 교육과정에 포함되어서는 안 된다는 견해도 있다.

그러나 프로그래밍이 교육적으로 효과가 있다는 것은 분명한 사실이다[5][7][8].

또한 프로그래밍 교육은 학습자 개인의 측면에서 논리적 사고력이나 문제 해결 능력과 같은 고등인지기술을 향상시킬 수 있을 뿐만 아니라, 국가적인 측면에서 S/W 개발 선진국으로 도약할 수 있는 교육적 기회를 제공한다는 점에서 중요한 의의를 가지고 있다고 본다.

이러한 관점에서 볼 때, 우리 나라 프로그래밍 교육 환경은 적절하지 못하다. 그 이유는 학교 교육에서 다루어온 대표적 프로그래밍 언어인 BASIC 언어는 교육용 언어로서는 부적절한 몇 가지 문제점을 안고 있다. 즉, BASIC은 간결하지만 강력하지 못하고[7][9][10], 문법에 대한 기계적 암기로 인하여 학습자의 인지적 부담이 크며, 논리적 사고력 함양에 적절한 교육용 프로그래밍 언어가 아니라는 문제점을 안고 있다.

그러므로 프로그래밍 교육의 현실을 개선하여 고등 인지 기술의 함양이라는 프로그래밍 교육의 목표를 달성하기 위해서는 프로그래밍 환경의 개선이 필수적이다.

새로운 교육용 프로그래밍 언어는 간결하고 학습자와의 뛰어난 상호작용성을 지니고 있어야 할 뿐만 아니라 논리적 추론 기능과 강력한 GUI(Graphic User Interface) 환경을 갖추어야 한다.

그러므로 본 연구에서는 이러한 요구 조건들을 충족시킬 수 있는 프로그래밍 교육 도구로서 교육용 한글 프롤로그 언어(EHPL; Educational Hanguk Prolog Language)를 구현하기 위하여, 이론적 고찰을 통하여 프로그래밍과 교육용 언어의 교육적 가치를 확인하고 EHPL에 적합한 GUI 방식의 사용자 인터페이스(EHPLUI; EHPL User

Interface)를 설계하고자 한다.

2. 이론적 배경

2.1 프로그래밍의 교육적 효과

교육적인 측면에서 프로그래밍의 의의는 두 가지로 요약할 수 있는데[11], 첫째, 프로그래밍 활동을 통해 학습자 스스로 사고력을 향상시킬 수 있다는 점과 둘째, 교과서의 문제들을 프로그래밍을 통해 해결할 수 있는 능력을 갖는 것은 정보화사회에 대비하는 지름길이라는 점이다. 여기서 사고력이란 합리적 사고력, 창의적 사고력, 확산적 사고 등을 의미하는 포괄적 용어이다.

한편, Opera(1984)는 프로그래밍의 교육적 효과를 선행연구 분석을 통하여 다음과 같이 제시하고 있다[1]. 프로그래밍 활동은 컴퓨터 소양(computer literacy), 문제해결(problem solving), 수학적 사고(mathematical thinking), 특수한 수학적 개념(special mathematics concepts), 논리적 추론(logical reasoning) 능력의 신장에 효과적인 활동이다.

2.2 이중 부호화 이론

시각 정보는 공간적으로 부호화 되며, 언어 정보는 계열적으로 부호화 된다. 즉, 시각적인 장면이 기억될 때에는 각 부분이 개별적이고 순차적으로 기억되는 것이 아니라, 현실에 존재하는 영상처럼 전체적인 흐름이 구성되면서 기억된다. 그러므로 어느 한 시점을 기억해내게 되면 앞뒤의 내용은 기억나게 된다. 반면에 언어 정보는 계열적으로 재생, 처리, 사용되므로 이미 기억된 순서에 새로운 언어를 삽입하거나 생략함으로써 재조직하여 사용할 수 있다[12].

이러한 사실에 기초하여 단어만으로 기억되거나 그림만으로 기억된 정보보다는 그림과 단어가 같이 결합된 정보가 훨씬 재생이 잘된다는 것을 알 수 있다. 그러므로 학습 장면에서 언어 정보와 시각 정보를 적절하게 구성하여 함께 제공하면 효과적인 학습이 가능하다.

교육용 언어의 필수 요건에 맞추어 LOGO와 BASIC의 특성을 나타내면 <표 1>와 같다[1].

2.3 교육용 프로그래밍 언어의 요건

교육용 프로그래밍 언어의 요건으로서 크게 다음의 세 가지를 들 수 있다[9][13].

1) 간결성(simplicity)

교육용 프로그래밍 언어는 구조적이며 체계적인 모양을 갖추어 학습자가 프로그래밍 언어를 학습하는 것이 쉬워야 한다.

이 요건은 다음의 강력성과 연관성이 많다.

2) 강력성(power)

교육용 프로그래밍 언어는 복잡한 문제들을 단순하게 표현하여 간단히 해결할 수 있는 프로그래밍 환경을 제공해야 한다.

강력한 프로그래밍 언어는 그래픽이나 오류수정, 순환 등의 기능을 지원해야 하는데 이러한 기능으로 인하여 학습자는 문제 해결력 증진의 기회를 더 얻을 수 있다[14].

3) 인지효율성(cognitive efficiency)

문제들에 대하여 사고하는 것을 쉽게 하는 프로그래밍 언어이어야 한다. 이런 언어는 문제 범위를 지배하는 규칙들에 대하여 가설을 시험할 수 있으며, 쉬운 표현 방법을 제공한다.

인지적 사고의 자연스러운 표현과 프로그래밍 언어에 의해 허용되는 표현간에는 차이가 있을 수 있는데, 이 차이는 문제해결에 있어서 학습자에게 큰 장애가 될 수 있다. 그러므로 이런 표현의 차이가 좁은 언어가 훌륭한 교육용 언어라고 할 수 있다.

이 요건은 간결성과 강력성의 두 가지 요건과 밀접한 상호 관련성이 있다.

이러한 세 가지 요건은 Prolog 언어에서 뚜렷이 나타나므로 교육용 프로그래밍 언어로서 Prolog의 교육적 가치를 재인식할 필요가 있다.

2.4 프로그래밍 언어

2.4.1 LOGO와 BASIC

<표 1> LOGO와 BASIC의 특성 비교

언어 요건	LOGO	BASIC
간결성	· 명령어를 입력하지마자 실행결과가 화면상에 즉시 나타난다.	· 작성한 즉시 실행 가능하지만, GOTO 명령의 사용으로 구조적 프로그래밍의 습관을 지닐 수 없다.
강력성	· 거북 그래픽이 교육적 특성을 극대화 한다. · 거북 그래픽은 상대 좌표를 사용하므로 초보자도 쉽게 그래픽을 사용할 수 있다. · 절차를 제공한다. · 오류 메시지는 알기 쉬워 오류 수정이 용이하다.	· 절대좌표를 사용하므로 초보자에게 어렵다. · 그래픽의 위치 표현이 연산능력을 요구하기 때문에 문자와 숫자처리를 배우고 나서야 그래픽 명령어를 학습할 수 있다. · 절차를 제공하지 않는다. · 오류 메시지는 오류수정에 별로 도움되지 않는다.
인지효율성	· LIST는 숫자, 문자 구분없이 어떠한 내용도 저장될 수 있다. · 절차적 언어이기 때문에 큰 프로그램으로 짜여지지 않고 작은 단위의 모듈로 나뉜다.	· 배열(ARRAY)은 크기가 고정되어 있으며, 숫자와 문자가 섞여 배열에 저장될 수 없다. · 프로그램 모듈화가 가능하지만, GOSUB 명령어와 행번호로 전체 프로그램에 들어있어 프로그램을 이해하기가 어렵다.

2.4.2 Prolog

Prolog는 Programming in Logic의 약자로서, 기호 처리를 쉽게 할 수 있으며 강력한 추론 기능을 가진 프로그래밍 언어이다. 또한 프롤로그는 그 자체로 문법을 표현하고 자체적인 프로그램 환경을 개발할 수 있음은 물론 여러 방면에서 그 유용성을 인정받고 있는 언어이다[15].

프롤로그의 특징은 다음과 같다[16].

- 1) 절차지향 언어와 달리 서술적 언어이다.
- 2) 프로그램의 대부분은 일련의 사실과 규칙의 형태로 기술된다.
- 3) 추론 기능을 이용하여 문제에 대한 모든 해를 구할 수 있다.
- 4) 패턴 매칭(pattern matching)시에 다른 패턴의 대응 여부를 확인하기 위해 설정된 가정들을 검토하는 백트래킹(backtracking)을 사용한다.
- 5) Pascal의 1/10 정도의 코딩만으로 문제해결이 가능하다. 이것은 간단한 문법, 효율적인 재귀적 구조(recursive structure)와 패턴 매칭 기능을 갖추고 있기 때문에 가능하다.
- 6) 대화식 프로그램의 개발이 가능하다.

2.5 HCI

HCI(Human Computer Interaction)란 인간과 컴퓨터의 조화를 모색하며, 사용하기 쉬운 인터페이스를 설계하는 학문이라는데 많은 사람들이 의견을 같이하고 있지만, 심리학, 생리학, 인간공학, 인지과학, 인공지능, 컴퓨터 과학 등의 학문과 연계되어 인간과 컴퓨터를 조화시킬 수 있는 방법을 모색하기 위한 모든 노력이 경주되는 학제적인(interdisciplinary) 학문이라고 할 수 있다 [17].

컴퓨터가 사회에 미치는 영향을 충분히 고려하지 않는다면 인류에게 미칠 수 있는 부정적인 결과를 막을 수 없다. 그러므로 사회가 받을 부정적인 영향을 사전에 예측하고 이를 최소화할 수 있는 방법에 대한 연구가 필요하다. 이런 역할을 하는 학문 중의 하나가 HCI이다.

결국 HCI는 인간이 컴퓨터 활용의 주체로서 바로 설 수 있도록 인간을 보조하는 아주 중요한 역할을 하는 학문이 아닐 수 없다.

HCI에서 사용되는 개념 중의 하나인 메타포(metaphor)는 프로그래밍을 쉽고 간편하게 할 수 있으며, 인간 중심의 인터페이스를 구현하는데 필수적인 요소이다. 메타포는 사용자 인터페이스 내의 일상적 기능과 유사한 대상이나 작동 방식에 생활 용품이나 행동 양식을 추상화하거나 간략화하여 적용한 것을 말한다. 예를 들어 윈도우에서 프린트 작업을 할 때 프린트 모양의 아이콘을 누르는 경우, 프린터 모양의 아이콘을 메타포라고 볼 수 있다.

3. 설계

3.1 EHPL

3.1.1 정의

EHPL은 프롤로그 언어의 뛰어난 논리적 추론 기능과 로고 언어의 장점 그리고 강력한 GUI 환경이 적절히 조화를 이루는 한편, 논리적 사고력

과 같은 고등 인지 기술 함양을 위한 인지적 요구 사항들을 수용하고, 최대의 교육적 효과를 달성할 수 있는 교육용 한글 프롤로그 인터프리터 언어를 말한다.

3.1.2 특징

EHPL은 Masterson(1984)이 주장한 교육용 프로그래밍 언어의 세 가지 요건을 충족시키면서 아래와 같은 강력한 학습 환경을 갖추고 있다.

1) 메타포를 이용한 쉬운 프로그래밍

체계적으로 분류되고 그림으로 표현된 메타포를 선택하는 것으로 프로그래밍이 가능하므로 프로그래밍이 재미있고 쉽다.

2) 인지 부담의 감소

객체의 형(type)을 선언할 필요가 없으므로 쉬운 프로그래밍이 가능하다. 사실 객체의 형을 학습자들이 이해하고 사용한다는 것은 어려운 일이다. 그러므로 EHPL은 입력된 값의 특성을 분석하여 자동적으로 타당한 형으로의 내부적 변환을 처리한다.

형을 선언하지 않음으로 인해서 학습자의 입력 작업을 대폭으로 줄일 수 있으며 학습자가 이해해야 할 기본적인 문법의 양도 줄어들게 된다.

또한 메타포를 적극 활용하여 문법을 기계적으로 암기할 필요가 없으므로 학습자의 인지적 부담을 줄일 수 있다.

3) 개념 학습의 기회 제공

메타포를 찾으며 사물의 분류와 개념에 대하여 이해할 수 있는 기회를 제공한다.

4) 강력한 논리적 기능

Prolog의 뛰어난 논리적 요소를 계승하여 단일화(unification), 백트래킹(backtracking), 추론 기능을 모두 지원한다.

5) 한글 프로그래밍 환경

객체나 술어 등의 프로그램 구성 요소에 한글 사용이 가능하다.

6) 멀티미디어 기능 강화

로고의 터틀 그래픽(Turtle Graphic)을 지원하므로 편리하고 쉽게 그래픽을 사용할 수 있다.

또한 단순히 컴퓨터의 내장 스피커를 이용하여 비프음만 지원하던 것을 미디(mid)나 웨이브

(wav), 동영상(avi) 파일을 재생할 수 있는 멀티 미디어 기능이 강화되었다.

7) 사용자 인터페이스 강화

비주얼 프로그래밍의 장점인 편리한 그래픽 사용자 인터페이스의 개념을 도입하여 프로그래밍이 쉽고 빠르다.

8) 쉬운 주석(comment) 처리

입력한 명령문을 임의로 주석 처리할 수 있으므로 명령문을 삭제하거나 설명문으로 변화시키는 종래의 불편함을 해소하였다. 이러한 주석 처리는 실시간 결과 출력에도 반영된다.

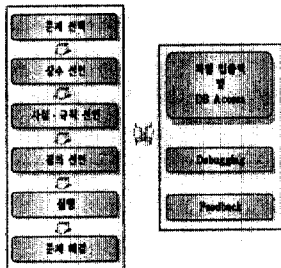
9) 실행 결과의 실시간 출력

프로그램을 실행시키지 않아도 실행 결과가 항상 출력되므로 실행 결과 확인이 용이하다.

10) 빠른 피드백(feedback)

프로그래밍 중에 발생한 오류에 대해서는 즉각적인 피드백을 지원하므로 오류의 원인 파악이 빨라 디버깅이 용이하다.

3.1.3 프로그래밍 과정



(그림 1) EHPL을 이용한 프로그래밍 과정

(그림 1)은 학습자가 EHPL을 이용하여 문제를 해결하기 위한 프로그래밍의 일반적인 과정을 나타낸 그림인데, 이러한 과정이 반드시 순차적으로 일어나야 함을 의미하는 것은 아니다.

즉, 프로그래밍 과정은 학습자의 지적·정의적 특성이나 해결해야 할 문제의 특성에 따라 문제 해결의 순서와 시간의 차이를 보일 수 있으며, 상수 선언은 선택적으로 활용할 수 있다.

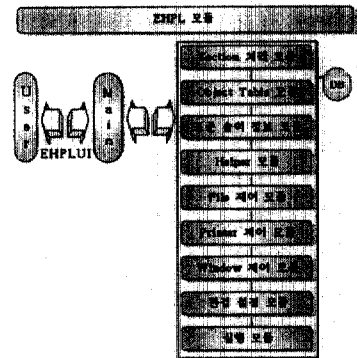
EHPL은 기본적으로 실행 결과를 결과출력창으로 출력하므로 학습자는 자동 실행 옵션을 해제하여 별도의 실행 과정을 선택할 수 있다.

3.1.4 모듈 구조

(그림 2)는 EHPL의 모듈 전체의 구조를 나타낸 것이다.

학습자는 EHPLUI를 통하여 Main 모듈과 상호작용을 한다. Main 모듈에서는 발생한 이벤트의 성격에 따라 적절한 하위 모듈을 호출하여 그 이벤트를 처리한다.

객체 정보 테이블(Object Table) 모듈에서는 학습자가 선언한 술어나 객체에 대한 정보뿐만 아니라 질의를 통하여 새로 구축한 객체 정보를 효율적으로 처리한다. EHPL의 핵심적인 모듈이라고 볼 수 있다. 객체 정보 테이블은 패턴 매칭이 가능하도록 하며, 내부적으로는 EHPL에 의해 관리된다. 객체 정보를 동적으로 관리하므로 객체 정보의 처리는 인자의 개수나 형(type)에 영향을 받지 않는다.



(그림 2) EHPL의 모듈 구조도

Section 처리 모듈은 선언된 객체 정보를 처리하는 모듈로서 상수와 사실, 규칙, 질의 선언을 처리하는 모듈이다. 학습자가 입력한 요소에 대하여 분석한 후 그 결과를 저장함으로써 객체 정보 테이블에 있는 기존의 객체 정보를 갱신되거나 새로운 정보가 추가된다.

Helper 모듈은 EHPL에 대한 도움말이나 문제 해결에 필요한 학습 자료를 제공하는 모듈이다.

Printer 제어 모듈은 프로그램의 소스나 질의에 대한 결과와 같은 자료를 프린터로 출력하는 모듈이며, File 제어 모듈은 생성한 정보나 프로그램 소스를 저장하거나 읽는 등의 파일 처리 기능

을 갖추고 있다.

Window 제어 모듈은 MDI 창들을 화면에 배치하거나 여닫는 관리기능을, 실행 모듈은 학습자의 질의를 실제로 실행하는 기능을 가진 모듈이며, 환경 설정 모듈에서는 파일 입출력을 위한 경로나 프린터의 상태, 작업 화면에 대한 설정을 변경할 수 있다.

데이터 베이스는 객체 정보 DB와 에러 메시지 DB, 메타포 DB로 구성되어 있다.

3.1.4 구성 요소

원칙적으로 프로그를 모델로 삼았기 때문에 EHPL의 어휘적 요소와 프로그램 요소는 비슷하지만, 여기에는 커다란 차이점이 두 가지 있다.

첫째는 EHPL은 기존의 프로그보다 입력에 있어서 제한사항이 줄어들었다는 것이다.

예를 들어 변수명은 반드시 대문자로 시작하여야 한다는 것과 같은 문법은 더 이상 의미가 없다. 왜냐하면 EHPL에서 학습자가 변수를 사용할 때 메타포를 이용하여 변수를 선언하게 되므로 어떠한 문자가 입력되더라도 변수로 받아들이기 때문이다.

둘째로 기존의 프로그에서는 요소들의 입력 작업이 복잡하고 지루하지만, EHPL에서는 쉽고 재미있다는 것이다. 그 이유는 EHPL이 GUI의 편리한 프로그래밍 환경 아래에서 메타포를 이용할 수 있으므로 입력 작업이 다양하게 이루어질 수 있다는 이유 때문이다.

3.2 EHPLUI 설계

3.2.1 정의

EHPLUI는 EHPL과 학습자의 원활한 상호작용을 위한 그래픽 기반의 사용자 인터페이스이다.

3.2.2 특징

EHPLUI의 특징은 다음과 같다.

1) 기본적으로 윈도우의 표준 컨트롤을 이용하지만, 학습자의 흥미 유발을 위해서 그래픽 컨트롤을 제작하여 적극 이용한다.

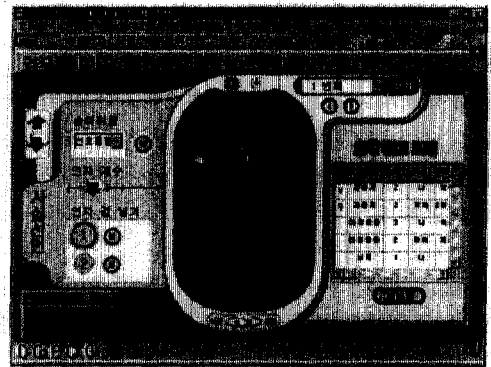
2) 메타포를 이용하여 학습자와 원활한 상호작용이 가능하다.

3) 학습자가 한번 입력한 정보는 재사용이 가능하므로 입력 작업이 더욱 쉽도록 도와준다. 즉, 동일한 정보나 비슷한 정보를 입력해야 할 경우, 입력을 위한 컨트롤에 자동적으로 관련 있는 정보가 표시되므로 선택만 하면 된다.

4) 인터페이스 재구성 능력이 있기 때문에 상황에 따라 입력하고자 하는 요소의 개수와 형태가 달라져도 유연한 입력이 가능하다.

5) 여러 요소들을 입력해야 할 때, 현재 작업창을 종료하고 다른 메뉴를 선택해야 하는 것과 같은 이중 작업이 필요 없고 하나의 창에서 모든 입력을 마칠 수 있다.

3.2.3 IDE 설계



(그림 3) IDE 화면

(그림 3)은 EHPL의 통합 개발 환경(Integrated Development Environment; IDE)을 나타낸 그림이다. IDE는 MDI(Multiple Document Interface) 방식으로 하나의 응용 프로그램 안에서 여러 개의 윈도를 표현할 수 있도록 하였고, Menu, Toolbar, MDI windows, Status information bar의 네 부분으로 구성되어 있다.

1) Menu

(그림 3)의 '가' 부분은 메뉴로서, 화일, 편집, 실행, 표준술어, 환경설정, 윈도, 도움말 메뉴로 나눌 수 있으며, 툴바에서 아이콘으로도 지원된다.

표준술어 메뉴의 하위메뉴는 표준 술어의 범주를 논리적으로 나누어, 술어를 이해하기 쉽고 사

용하기 쉽도록 구성되어 있다.

<표 2>에서 나타낸 바와 같이 주메뉴는 7개의 하위 메뉴로 구성되어 있다.

<표 2> 메뉴 구조

메뉴	하위메뉴
파일	새로만들기, 열기, 저장, 다른이름으로 저장, 페이지설정 인쇄
편집	실행취소, 잘라내기, 복사, 붙여넣기, 전체선택
실행	실행
내장술어	인, 출력, 그래픽, 연산, 데이터 베이스, 제어용 술어 입력, 사용자 술어 찾기
윈도	윈도 정렬
환경설정	화면설정, 프린터설정, 경로설정, 실행 옵션
도움말	내용, 찾아보기, HTML 소개

2) Toolbar

(그림 3)의 '나'부분은 툴바를 나타내며, 상단은 프로그래밍의 진행 정도를 추정하는 막대와 입력 작업을 빠르게 하기 위한 버튼으로 구성된다.

그리고 하단은 메뉴에서 지원하는 명령들을 간편하게 선택할 수 있도록 아이콘으로 구성되어 있다. 그를 지어진 각 영역은 상위 메뉴에 대응하며, 각각의 버튼들은 하위 메뉴에 대응한다.

3) MDI windows

(그림 3)의 '다'부분은 MDI 윈도를 나타내며, MDI 윈도는 소스 윈도, 메시지 윈도, 질의 윈도, 결과출력 윈도로 구성되어 있다.

소스 윈도는 작성한 프로그램의 소스를 보여주며, 소스를 수정할 수 있는 창이다. 소스를 수정할 때에는 수정하고자 하는 부분을 마우스로 클릭하면, 해당되는 선언 상자가 열리게 되므로 쉽게 수정할 수 있다.

그리고 소스 윈도는 메타포로 정보를 표시하는 메타포 모드와 문자로써 텍스트 모드를 표시하는 두가지 모드를 이용하여 소스 보기 및 수정이 가능하다. (그림 4)는 메타포 모드에서 입력 정보를 출력한 것을 나타낸 그림이다.



(그림 4) 메타포 찾기 인터페이스

메시지 윈도는 학습자의 입력 오류나 실행상의 오류, 기타 상황의 변화를 알려주는 역할을 하는 창이며, 질의 윈도는 문제 해결을 위한 질의를 보여주는 창이다.

결과출력 윈도는 질의한 결과를 보여주는 창인데, 이 창에서는 텍스트뿐만 아니라 정지 영상, 동영상 및 사운드의 출력도 가능하다.

또한 실행 옵션에서 실시간으로 질의 수행결과를 결과출력 윈도로 출력할 것인지 아니면 차후에 일괄적으로 수행할 것인지를 지정할 수 있다.

4) Status Information bar

(그림 3)의 '라'부분은 상태 정보 막대를 나타낸다. 상태 정보 막대(Status information bar)는 프로그래밍 활동을 돕기 위하여 현재 IDE의 상태를 나타낸다.

아이콘의 순서대로, 각 Section에 선언된 객체의 수에 관한 정보, 프로그램의 저장 상태 정보, 파일 정보, 입력 대기 상태, 시스템 시간 정보, 사용하고 있는 데이터베이스 파일명의 정보를 표시하는데, 해당 icon에 마우스를 위치시키면 풍선 도움말이 나타나 적절한 정보를 표시한다.

3.2.4 입력 인터페이스 설계

입력 인터페이스는 메타포 지정 인터페이스, 상수 선언을 포함한 사실과 규칙 선언 인터페이스, 질의 선언과 그래픽 선언 인터페이스로 나뉘어진다.

1) 메타포 검색 인터페이스

메타포를 검색하는 방법은 두 가지가 있다.

문자 검색 상자에 찾을 메타포의 이름을 입력한 후 직접 검색하는 방법과 분류 상자를 이용하여 상위 개념의 사물을 찾은 후 하위 분류 상자에서 상위 개념에 소속된 하위 개념의 사물을 찾는 단계적인 검색방법이 있다.

(그림 5)는 메타포 찾기 인터페이스를 나타낸 그림이다. 자연이라는 상위 개념에 식물과 동물, 우주라는 하위 개념이 포함되어 있다는 것을 나타낸다. 이와 같이 분류는 상위 개념과 하위 개념으로 계층적인 구성을 이루고 있다.

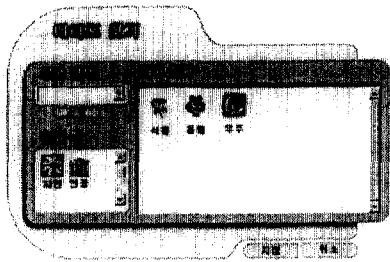
만약 학습자가 동물을 선택하면, 동물의 하위 개념들이 하위분류에 메타포로서 나타난다.

2) 사실 선언 인터페이스

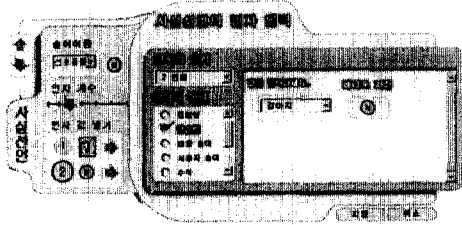
단순 객체(simple object), 복합 객체(compound object), 다중 객체(multiple object), 순환 객체(recursive object) 등의 선언이 가능하다.

(그림 6)은 단순 객체의 입력 인터페이스를 나

타낸 그림이다. 이 상태를 CUI 방식으로 표현하면 '선호동물(나, 강아지)'와 같다.



(그림 5) 메타포 찾기 인터페이스



(그림 6) 단순 객체의 입력 인터페이스

복합 객체는 술어 내의 인수가 다른 술어로 정의된 객체이며, 프로그래밍을 간단하게 하는 중요한 요소이다. 다중 객체는 인수 안의 술어 제한 술어의 개수가 2개 이상인 객체를 말한다. 순환 객체는 인수로 정의된 객체의 수가 미리 알려져 있지 않은 경우에 인수로써 자신의 술어를 정의하는 객체이다.

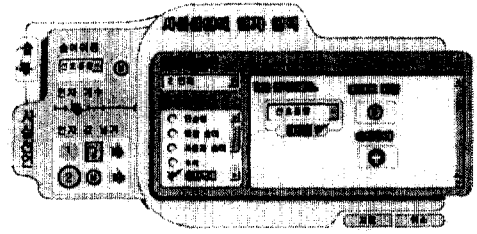
'집합형'은 리스트(list)형을 의미하며, 단순형은 술어가 없는 객체를, 내장 술어는 EHPL에서 지원하는 내장 술어를, 사용자정의는 학습자가 정의한 술어를 의미한다. 술어 이름과 인자의 개수를 지정하면, 지정한 개수에 따라 인자를 입력할 수 있도록 입력 인터페이스가 재구성되어 나타난다. 'M'이라는 버튼을 누름으로써 메타포 검색을 통한 메타포 삽입이 가능하다.

(그림 7)은 순환 객체의 되부름을 선언하는 인터페이스를 나타낸 그림이다.

이것을 CUI 방식으로 표현하면 '선호동물(나, 선호동물, empty)'와 같다.

이 객체 선언이 다른 객체의 선언과 다른 점은 순환의 끝을 표시하는 '되부름' 지정 부분이 있다는 것이다. 이 '되부름' 선언은 버튼을 누름

으로써 사용이 가능하다.



(그림 7) 순환 객체의 입력 인터페이스

만약 호출된 술어가 empty인 경우는 호출이 중단되고 백트래킹(backtracking)이 진행된다. 이와 같은 방식을 되부름(recursion)이라고 한다.

'추가' 버튼의 기능은 하나의 술어를 추가할 수 있도록 하는 기능을 한다.

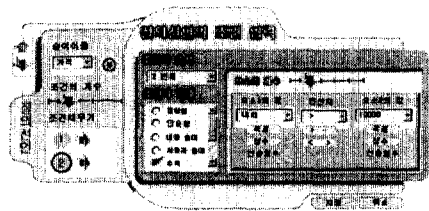
이 외에도 선택적으로 이용할 수 있는 상수 선언 인터페이스가 있다. 이 인터페이스는 단순 객체의 입력 인터페이스와 거의 흡사하며 반드시 이용해야 할 필요는 없다. 그러나 이 인터페이스를 EHPL이 지원하는 이유는 프로그램 수정을 용이하도록 하기 위해서이다.

2) 규칙 선언 인터페이스

규칙을 선언하는 규칙(rules) 선언 인터페이스는 질의 선언 인터페이스와 흡사하다.

3) 질의 입력 인터페이스

질의에는 조건이 없는 단순 질의와 조건이나 수식이 있는 복합 질의가 있다. 만약 '가격'이라는 술어를 사용하여 질의하고 싶은 경우에 먼저 '가격'을 술어로 지정하면 자동적으로 2개의 인자를 입력할 수 있도록 인터페이스가 재구성된다. 이것이 가능한 이유는 EHPL이 객체 정보 테이블에 '가격' 술어에 대한 정보가 있기 때문이다.



(그림 8) 복합 질의 선언 인터페이스

(그림 8)은 복합 질의 선언의 인터페이스이고,

'수식' 버튼을 눌렀을 때 나타나는 인터페이스를 나타낸 그림인데, 수식을 지정하면 수식에 포함되는 요소의 개수를 지정한 후 각 요소의 값을 요소의 형태에 따라 지정할 수 있다.

수식도 역시 하나의 조건으로 간주하여 조건의 개수에 따라 입력 인터페이스를 재구성하여 학습자에게 보여준다. '산술 함수' 버튼을 누르면, EHPL에 내장된 산술용 내장 함수들이 나타나므로 원하는 산술함수를 선택하면 된다. '+' '-' 버튼은 산술연산자를, '<' '>' 버튼은 관계연산자 중의 하나를 선택할 수 있는 기능을 가지고 있다.

위에서 설계한 사용자 인터페이스 환경에서 입력한 모든 정보는 객체 저장 테이블에 저장되어 검색과 연산에 이용된다.

3.1.7 교육적 효과

EHPL을 활용함으로써 기대되는 교육적 효과는 다음과 같다.

1) 고등 인지 기술을 개발

프롤로그 프로그래밍의 특성상 문제 해결의 과정에서 학습자의 논리적 사고력과 반성적 사고력, 문제해결력과 같은 고등 인지 기술을 개발할 수 있다.

2) 흥미 유발 및 학습태도 개선

로그 언어 활용의 장점을 계승하고, 메타포를 이용하므로 이해하기 쉽고, 흥미유발이나 학습에 대한 태도를 개선시킬 수 있다.

3) 인지적 부담의 감소

객체의 특성에 따라 사용자 인터페이스가 자동적으로 재구성되므로 입력 작업의 양이 줄며, 문법을 기계적으로 암기할 필요가 없다.

4) 문제해결 및 프로그래밍에 대한 동기유발

사용하기 편리한 인터페이스와 교육용 언어의 요건을 갖춘 EHPL은 학습자들에게 프로그래밍에 대한 자신감을 심어주어 프로그래밍 활동을 지속적으로 할 수 있도록 내재적 동기를 유발시킬 수 있는 아주 중요한 요소이다.

5) 다른 언어에 대한 관심의 증대

EHPL 프로그래밍에 대한 자신감은 다른 언어로 관심의 범위를 확대시켜준다.

6) 정보과학 영재 교육에의 활용

Passow는 영재교육의 목표를 달성하기 위해서

는 교수 학습의 사태 속에서 강조되어야 할 학습 경험들이 있는데, 그 경험 속에는 문제 해결 과정, 문제해결에 필요한 정보와 자료, 연구 방법론과 수단 등에 대한 이해와 문제해결에 필요한 정보와 자료들을 효과적으로 이용할 수 있는 능력을 기르도록 안내하는 활동도 포함되어 있다[18].

EHPL은 다른 언어에 비해 논리적인 요소를 많이 가지고 있으므로 이러한 활동에 적절한 교육용 언어이다.

4. 결론

프로그래밍의 초기 단계에 있어서, 문법에 대한 기계적 암기나 사용하기 불편한 CUI와 같은 프로그래밍 환경은 학습자의 인지적 부담을 가중시켜, 결국은 프로그래밍에 대한 흥미와 관심을 잃게 하여 프로그래밍 교육의 목표 달성과는 요원해진다. 그러므로 학습자를 위한 사용하기 편리하고 쉬운 사용자 인터페이스는 프로그래밍 언어의 필수 기능에 못지 않게 중요한 요소이다.

본 연구에서는 체계적으로 분류된 메타포를 이용함으로써 프로그래밍에 대한 학습자의 흥미와 관심을 증대시킬 뿐만 아니라 문제해결력이나 논리적 사고력과 같은 고등 정신 기술의 함양에까지 영향을 미칠 수 있는 사용하기 쉽고 편리한 사용자 인터페이스를 설계하였다.

OOP(Object Oriented Programming)를 지원하며, DCOM(Distributed Component Object Model)이나 CORBA(Common Object Request Broker Architecture)를 지원하는 프로그래밍 언어를 이용하여 EHPL을 구현하면, EHPL을 최소한의 노력으로 웹 기반의 학습 환경으로 쉽게 이식시킬 수 있다.

참고문헌

- [1] 이유순(1995). 논리적 사고력 및 문제해결력 신장을 위한 컴퓨터 프로그래밍 교육. 이화여자대학교 교육대학원 석사학위논문.
- [2] 이태욱(1996). 컴퓨터교육과 관련된 한미교육과정 비교 분석에 관한 연구. 정보과학회지, 14(12).

[3] 우연상(1993). logo 프로그래밍의 발견식 수업과 지시적 수업이 문제 해결 능력에 미치는 효과. 한국교원대 대학원 석사학위논문.

[4] Soloway(1986). Learning to program = Learning to Construct Mechanisms and Explanations. Communications of the ACM, 29(9), pp.850-858.

[5] 전성운, 백영균(1992). 교육과 컴퓨터. 양서원.

[6] 김수한(1991). 논리적 사고력 신장을 위한 LOGO 프로그래밍 활동의 효과분석. 한국교원대 대학원 석사학위논문.

[7] Papert(1980). Children, Computers and Powerful Ideas. Basic Books.

[8] Mitterer(1986). LOGO and the Transfer of Problem Solving : An Empirical Test. *The Alberta Journal of Educational Research*, 32(3), pp.176-194.

[9] Masterson(1984). Languages for Students. Byte.

[10] Tim, John(1983). Learning and Teaching with Computers: Artificial Intelligence in Education. The Harvest Press Limited.

[11] 백영균, 류희찬(1992). 아동과 컴퓨터. 양서원.

[12] 김희수(1995). 멀티미디어 설계와 개발. 교육과학사.

[13] 김명렬(1989). 컴퓨터 교육용 프로그래밍 언어 CELL의 설계에 관한 연구. 홍익대학교 대학원 박사학위 논문.

[14] 나현미(1991). 컴퓨터문맹 탈피를 위한 교육용 프로그래밍 언어의 비교연구. 동국대학교 대학원 석사학위논문.

[15] 박성배(1998). 인공지능 프로그래밍을 위한 Turbo Prolog 이론과 실제. 웅보출판사.

[16] 조동섭(1988). 인공지능언어 Turbo Prolog 입문. 영진출판사.

[17] 윤철호(1996). 인간컴퓨터인터페이스. 대영사.

[18] 송용대(1998). 영재교육(http://www.kerinet.re.kr/consult_1.html 경기도 과학 교육원).

[19] Dalbey, Linn(1985). The Demands and Requirements of Computer Programming. *Journal of Educational Computing Research*, 1(3), pp.253-274.

[20] Shneiderman(1998). Designing the User Interface. Addison Wesley.

[21] Preece(1994). Human Computer Interaction. Addison Wesley.

송 태 옥

1991 부산교육대학 초등교육과
교육학 학사
1998 한국교원대학교 컴퓨터
교육과 교육학석사

1998~현재 한국교원대학교 컴퓨터교육과
박사과정

관심분야: 컴퓨터 교육, 인성·창의성교육, HCI,
원격교육, 네트워크, 3D 그래픽 등

E-Mail: kinggem@comedu.knue.ac.kr

안 성 훈

1990 청주교육대학 과학교육과
교육학 학사
1997 한국교원대학교 컴퓨터
교육과 교육학석사

1997~현재 한국교원대학교 컴퓨터교육과
박사과정

관심분야: 컴퓨터 교육, 원격교육, 네트워크 등

E-Mail: ninano@comedu.knue.ac.kr

김 태 영

1985 한양대학교 산업공학과
공학 학사

1990 Texas A&M University
컴퓨터과학과 공학석사

1994 Texas A&M University 컴퓨터과학과
공학박사

1994 삼성데이타시스템즈(주) 정보기술연구소
선임연구원

1994~현재 한국교원대학교 컴퓨터교육과 조교수
관심분야: 데이터베이스, 네트워크, 원격교육

E-Mail: tykim@comedu.knue.ac.kr