

CLIPS를 사용한 한글 전문가 시스템을 위한 사용자 인터페이스의 開發

Development of User-Interfaces for Expert System Using CLIPS

조성인*

김승찬*

S. I. Cho

S. C. Kim

Summary

In developing an Expert System(ES), there are two ways. One is to develop an ES using AI(artificial Intelligence) languages and another using ES-development tools. CLIPS is an ES-development tool and has a powerful inference engine in it. Using the tool like CLIPS, knowledge engineer can concentrate on constructing a knowledge base without wasting time in developing an inference engine.

However, CLIPS is lack of user-friendly interfaces for both knowledge enginners and users. Because CLIPS was developed in USA, it can not afford to use Korean language. Therefore, several user-friendly interfaces including *hmenu*, *htitle*, *hpcxdisplay* were developoed and added to CLIPS. CLIPS with the interfaces is called HCLIPS(Hangul CLIPS) in this paper. HCLIPS provides a new I/O device to be utilized for expert systems in Korean. HCLIPS can be efficiently used for developing expert systems in agriculture and consulting farmers interactively who are not familiar with computer programming and ES itself.

I. 緒論

생물체와 자연현상을 대상으로 하는 農業에 있어서 불확실하고 애매한 정보를 처리해야 하는 경우가 많은 데 이러한 문제점은 人工知能 기술을 이용함으로써 해결되어 지고 있다. 전문가 시스템은 人工知能의 한 분야로서 人工知能 이 실용화되는 과정에서 가장 각광받는 분야중의 하나이다. 전문가 시스템이란, 한 분야의 전문가들이 고도의 지식과 경험을 바탕으로 論理와 推論의 지적 활동을 통해 문제를 해결하는 과정을, 인간전문가를 대신하여 컴퓨터라는 도

구를 통해 실현하는 것이다.

전문가 시스템은 의학, 과학, 공학, 경영, 제조업 등 많은 분야에서 1980년대 초부터 실용적으로 응용되기 시작하면서 주목을 받아 왔다. 농업에 있어서 전문가 시스템의 적용 분야는 상당히 광범위하고 다양하지만 그 중에서 대표적인 예를 몇가지 들면 다음과 같다.

- 1) 농업기계의 고장진단 및 修理支援 시스템
- 2) 작물의 병충해 진단 및 방제 시스템
- 3) 農水産物의 재배양식, 생산, 품질 결정 지원 시스템
- 4) 농업기계 및 시설의 지능화

+ 본 연구는 1992년도 서울대학교 농업생명과학대학 발전기금의 지원으로 수행되었음.

* 서울대학교 농업생명과학대학 농공학과

5) 農畜林水産業 시설 및 유통체계 관리 시스템

일반적으로 전문가 시스템을 개발하는 방법은 전문가 시스템 개발 도구를 사용하는 방법과 직접 프로그래밍하는 방법이 있다. 이중 직접 프로그래밍하는 방법은 人工知能 언어인 LISP나 PROLOG 등을 이용한다. 이것은 전문가의 지식을 규칙화하는 지식베이스(Knowledge Base) 구축에 드는 시간과 노력보다 상대적으로 推論 기관 프로그램과 인터페이스의 개발 등에 더욱 많은 시간과 노력을 많이 소모하게 된다. 이러한 방법은 상당한 人工知能 프로그래밍 기술을 요구하게되고 추후에 개발된 시스템의 기능과 지식의 보완 및 유지에 있어 전체 프로그램의 틀 속에서 수정을 해야하는 등 어려움을 겪게 된다.

따라서 推論기관부와 지원 환경을 구비한 전문적인 개발 도구를 사용하는 것이 일반적이라고 할 수 있다. 현재 많이 쓰이고 있는 개발 도구로는 ART, KEE, KES, M1, OPS, XSYS 등이 있다. 이들은 프로그래밍의 고급기술이 없어도 전문가 시스템의 가장 중요한 부분인 지식베이스 구축이 용이하도록 되어 있다. 또한 지식베이스부는 별도의 파일로 관리되어, 개발된 전문가 시스템의 추후 검증과 성능 향상을 위한 지식베이스의 수정 및 보완이 용이하다. 또한, Shell Tool형태의 전문가 시스템 개발 도구를 사용하여 구축한 전문가 시스템은 다른 기종으로의 이식이 용이하다고 할 수 있다. 즉 사용자는 Shell Tool이 있는 시스템이면 어느 곳에서든지 전문가 시스템을 Loading하여 실행할 수 있게 된다.

그러나, 이러한 범용의 전문가 시스템 개발 도구들은 시스템 개발자를 위한 개발자용 환경(Development Environment)은 잘 具備되어 있지만 실제 전문가 시스템을 사용하게 될 일반 사용자를 위한 사용자 인터페이스(End-User Interface)부분은 충분하지가 않다. 또한, 이러한 개발 도구들은 가격이 상당히 고가이고 아울러 한글 입출력이 제공되지 않거나 편법을 이용하고 있어 국내에서 사용할 한글용 전문가 시스템 개발에 부적합하다.

따라서 위와 같은 문제점을 해결하지 않으면 전문가 시스템을 개발하더라도 농민과 같은 비 전문적인 지식을 가지는 사용자들에게 개발된 전문가 시스템의 보급이 어려워진다. 이에 본 연구는 미국 NASA에서 개발된 범용 전문가 시스템 개발 도구인 CLIPS에 맞는 한글 사용자 인터페이스 기능들을 개발하여 CLIPS에 추가함으로써 이러한 문제점들을 해결하여, 국내 농업 분야의 전문가 시스템을 용이하게 개발, 보급할 수 있는 농업용 전문가 시스템 개발 도구를 구축하는 데 그 목적이 있다.

II. 한글용 CLIPS(HCLIPS)의 開發

가. CLIPS의 소개

위에서 제기한 문제점 중 저렴한 가격으로 보급이 용이한 전문가 시스템 개발 도구에 적합한 것이 바로 공개된(public domain) 개발도구를 이용하는 것이다. 이에 부응하는 것이 바로 CLIPS이다. CLIPS는 C Language Integrated Production System의 약자로 미국 NASA, Johnson Space Center의 人工知能 연구팀이 개발한 것이다. CLIPS는 推論기관(Inference Engine), 사용자 인터페이스, 그리고 전문가 시스템 구축용 프로그래밍 언어를 갖춘 범용 전문가 시스템 개발 도구로서 기본적으로 LISP와 비슷한 명령 체계와 順方向 推論 방식(Forward chaining)을 위주로 하여, 逆方向 推論(Backward chaining)도 가능한 推論기관을 채택하고 있다. CLIPS의 지식베이스의 구성을 살펴보면 아래와 같다.

```
(defrule rule-name
  (CONDITIONS)
  =>
  (ACTIONS))
```

위의 규칙을 살펴보면, defrule의 다음의 rule-name은 규칙의 이름이고, IF (조건부), THEN

(행동부) 형식으로 되어있다. 조건부(CONDITIONS)가 사용자가 입력한 정보와 일치하면 행동부(ACTIONS)가 실행된다. 프로그램 구조면에서 보면 앞에서 언급한 바와 같이 LISP의 구조와 비슷함을 알 수 있다.

또한 CLIPS는 컴퓨터 기종에 관계없이(Platform Independent), PC, Sun, VAX, HP, Macintosh 및 Cray에서도 실행이 가능하다. CLIPS는 현재 미국 정부기관, 대학, 연구소, 일반 산업 및 경영 분야에서 다양한 전문가 시스템 구축에 쓰이고 있다.

CLIPS의 가장 커다란 특징은 공개된 프로그램으로 누구나 소유할 수 있다는 것이다. 그런데 CLIPS도 역시 개발처가 미국이어서 한글 입출력이 불가능하다. 더구나 CLIPS는 사용자 인터페이스가 빈약하여 비전문가가 사용하기에는 어렵다는 또다른 문제점을 발생시킨다. 그러나, CLIPS는 프로그램 소스(C 언어)가 공개되어 임의로 수정할 수 있고, 또한 I/O Reroute라는 방식을 통하여 입출력을 재설정할 수 있는 방법을 제공하므로, 이를 적절히 활용하면 위의 두가지 문제점을 해결할 수 있다. 이러한 이유로 CLIPS가 우리의 목적에 맞는 농업 전문가 시스템 개발 도구로서 적합하다고 할 수 있다.

따라서 적절한 사용자 인터페이스 개발 도구만 갖춘다면 값이 비싸 보급에 문제가 있는 다른 전문가 시스템 개발 도구 및 LISP를 대신하여 CLIPS를 사용한 전문가 시스템의 보급이 유리한 입장에 있다고 할 수 있다.

나. HCLIPS의 開發

위에서 언급한 CLIPS의 두가지 문제점을 해결하기 위해서 한글 입출력은 CLIPS의 I/O Reroute 기능을 사용하였고 사용자와의 인터페이스(End-User Interface)를 구현하기 위해서는

CLIPS에 새로운 명령어(command)를 추가하여 해결하였다.

위에서도 언급하였지만 CLIPS는 C언어로 작성되었으며, 또한 프로그램 소스가 공개되어 있고 또한 새로운 명령어의 추가가 용이하도록 설계되어 있다. 본 연구에서는 CLIPS에 추가할 기능을 각각 module화 하여 프로그래밍하였다. 또한 CLIPS의 입출력 관계를 개선하기 위해 새로운 I/O device를 정의하여 추가하였다. 이렇게 새로 작성된 모듈(module)은 Borland C++ 3.0을 통해 컴파일(compile)되어 기존의 CLIPS 모듈과 링크(link)되었다. 작성된 프로그램의 전체는 그 양이 상당히 방대하여 일부만 부록에 수록하였다.

CLIPS의 모든 입출력은 logical device¹⁾를 통해 이루어지고 새로이 추가될 수도 있다. 이것이 I/O Reroute기능이다. 이 기능을 이용하여 CLIPS에 한글 입출력기능을 추가하였고, command line모드(CLIPS의 일반적인 사용자 인터페이스로 DOS의 그것과 동일하다)에서의 편집 기능도 확장하였다. 기존의 CLIPS에서는 입출력이 한 화면에 나오게 되어있는데, 기능이 확장된 CLIPS는 그림 1과 같이 입력창과 출력창이 구분되어 있고 상태줄(status line)이 표시되어 사용의 편리성을 증가시켰다. 확장된 기능으로는 20개 정도의 이전에 수행된 명령을 기억할 수 있는 history기능을 비롯하여, 사용자와의 인터페이스를 강화하기 위한 몇가지 명령어를 추가하였다. 추가된 명령어들은 hmenu, hcls, htitle, hpcxdisplay, hyesno, htexitdisplay 등이 있다. 그림 2는 이러한 사용자 인터페이스가 추가된 CLIPS를 이용한 한글 전문가 시스템의 구성도이다.

1) logical device : 사용자는 직접 하드웨어적인 장치를 통하지 않고 논리적인 장치를 통하여 입출력을 하고, 각 logical device에 해당하는 부분이 하드웨어적인 입출력을 담당하는 방식으로, 이는 사용자에게 장치에 관계없이 입출력할 수 있는 장점을 제공한다. 또한 새로운 장치가 추가되거나 입출력장치가 변경되더라도 사용자는 이에 신경 쓸 필요가 없고 상응하는 logical device를 설치하기만 하면 된다.

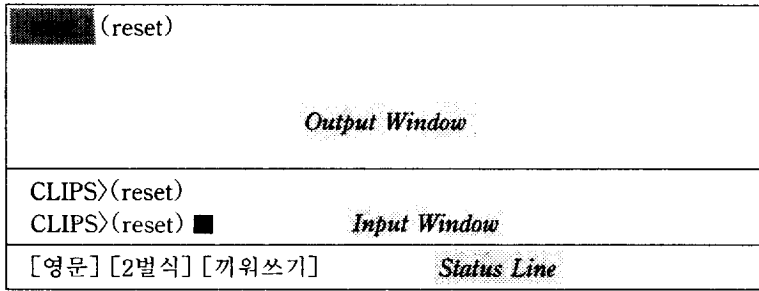


Fig. 1 Divided I/O windows of HCLIPS

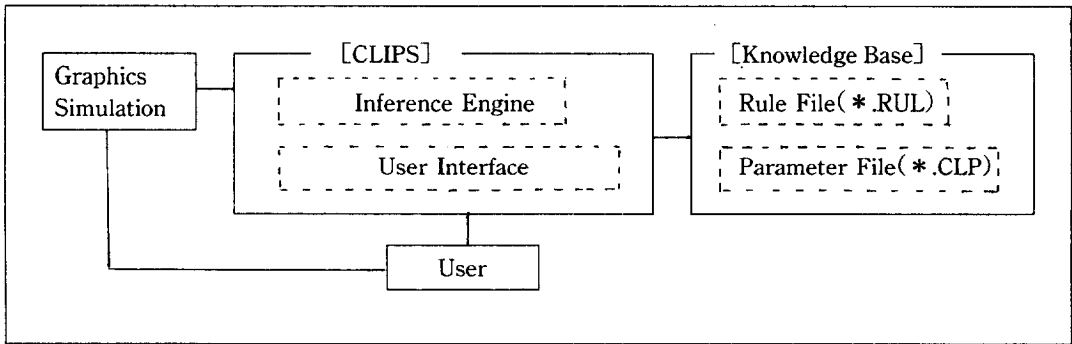


Fig. 2 Block diagram of an Expert System using HCLIPS

다. HCLIPS의 기능

① hmenu

HCLIPS에서 가장 중심이 되는 명령어로 사용자에게 다양한 형태의 질문을 할 수 있도록 한다. hmenu는 하나의 multifield를 인자로 갖는 명령어으로써, 주어진 인자에 따라서 4가지 기능 중의 한가지를 수행하는 명령어이다.

이러한 hmenu의 기능을 사용하는 방법은 기본적으로 CLIPS의 deffacts구문을 써서 multifield(LISP의 list와 비슷한 것) 변수를 선언하고, 이 변수를 hmenu에게 전달함으로써 실행될 수 있다. 이 경우 multifield에 속한 인자들에 따라 hmenu는 각각의 기능을 수행한다. hmenu가 사용하는 인자들의 의미는 표 1에 표시하였다.

이들 인자를 받아서 hmenu는 미리 정해진 형태로 화면을 구성하게 된다. hmenu가 구성하

는 화면의 한 예를 보면 그림 3과 같다. 화면에서 보면 영역 A는 출력창으로 CLIPS의 모든 기본 출력은 이곳으로 나타나게 된다. 다음 영역 B는 hmenu의 질문이 출력되는 곳이고, 영역 C는 사용자의 입력창으로 메뉴형식의 질문일 경우 메뉴가 보이고, 사용자 직접 입력형식일 경우 사용자의 입력을 받는 곳으로 사용하게 된다. 다음 영역 D는 사용자에게 지시사항을 전달하는 곳으로 질문의 형태에 따라 지시사항이 변하므로, 이곳의 지시에 따라 사용자는 질문에 응답하게 된다. 영역 E는 도움말 및 그림을 보여주는 곳으로 사용자가 F1, F2, F3의 키를 각각 누를 때, 이에 상응하는 적당한 화면을 보여준다. 영역 E의 초기화면은 사용자에게 주어진 질문에 도움이 되는 그림을 보여준다.

질문이름	질문의 이름에 해당하는 것으로 사용자가 임의로 정한다.	
질문내용	이 예약어 다음에 나오는 내용이 질문이 된다. 항상 이중 따옴표로 묶어 주는 것이 좋다.	
예상되는답	“질문내용”과 “답의형태”사이에는 기대값이 온다. 즉, 이곳에 어떤 내용이 있으면, 이들을 바탕으로 Menu를 구성하여 사용자에게 보여주고 선택하게 한다.	
답의 형태	단답	사용자는 한가지 답만 할 수 있다. 만약 Menu가 구성될 수 있는 상황이면, Menu를 보여주고 그 중의 한가지만 선택하게 하고, 그렇지 않으면 사용자가 직접 입력하게 된다.
	복답	이는 Menu가 구성된 경우에만 의미가 있는 것으로 사용자가 동시에 여러가지를 선택할 수 있게 한다.
	수치	사용자가 수치 자료만 입력할 수 있게 한다. Menu가 구성되면 이는 무시된다.
예상되는 수치범위	사용자가 수치입력을 할 경우에 수치의 범위를 제한한 필요가 있다면 이곳에 값을 설정하면 된다. 최소값과 최대값이 설정이 된다. 만약 값의 설정이 없다면 제한이 없어지게 된다.	
도움말	이 뒤에 도움말이 온다.	
도움그림	사용자에게 질문할 때 도움이 되는 그림을 보여주기 위한 그림 화일의 이름을 적어준다. 이때 그림의 저장 방식은 16색/PCX형태이다.	
질문이유	이 뒤에 질문을 하는 이유를 쓰면 된다.	

Table 1 Parameters used for 'hmenu'

영역 A			
F1-도움말		F2-도움그림	F3-질문이유
질문	과수업이외에 농업,임업,축산업 등을 겸업하는 비율은 얼마나 됩니까? 영역 B	도움말	시설원에는 70% 이상으로 본다. 영역 E
사용자의 대답	70% 이상 절반 정도 30% 이하 영역 C	그림	
	한가지만 고르세요 영역 D		

Fig. 3 Displayed windows using 'hmenu'

다음 예들은 각각 인자들을 바꾸어 다른 형태의 hmenu의 기능을 사용하는 것을 보여주고 있다.

그림 4를 보면, 사용자에게 “질문내용” 다음의 질문을 하고, “도움그림” 다음의 그림인 “GPPIC001.PCX”를 보여주면서 메뉴 형식의 질문을 한

다. “답의형태”가 “단답”이므로 사용자는 한가지의 답만 고를 수 있게 된다. 사용자가 F1을 누르면 도움말을 출력하고, F2를 누르면 “도움그림”의 그림화일인 “GPPIC001.PCX”를 영역 E에 보여주고, F3을 누르면 “질문이유”에는 인자가 없으므로 아무런 응답이 없게 된다.

(“검업”
 “질문내용” “과수업이외에 농업,임업,축산업 등을 검업하는 비율은 얼마나 됩니까?”
 “예상되는답” *70%이상 “절반정도” *30%이하
 “답의형태” *단답
 “예상되는수치범위”
 “도움말” *시설원예는 70% 이상으로본다
 “도움그림” *GPPIC001.PCX
 “질문이유”)

Fig. 4 Question type 1 : User should choose one answer.

다음 그림 5를 보면, “예상되는 답” 다음에 인자를 추가하지 않아서 가장 간단한 형태로 hmenu를 사용하는 형태이다. 즉 사용자에게 질

문을 하고, 직접 답을 요구하는 경우이다. 이때 “답의 형태”의 인자는 “단답”을 추가하거나 생략하면 된다.

(“재배작물”
 “질문내용” *대상 지역의 처리할 주된 청과물은 무엇입니까?
 “예상되는답”
 “답의형태” *단답
 “예상되는수치범위”
 “도움말” *도움그림 “질문이유”)

Fig. 5 Question type 2 : User should put a string required.

셋째로 그림 6은 역시 “예상되는 답” 다음에 인자를 추가하지 않은 것으로 사용자가 숫자만 입력할 수 있는 형태로 “예상되는 수치범위”에 값이 [-1000, 1000]을 할당하였으므로 이 범위에서 수치데이터를 입력할 수 있다. 만약 “예상되는 수치범위”에 값이 할당이 되지 않으면 범위제한은 실행환경(PC, W/S)에만 제한을 받는다. 만약 사용자가 입력한 내용을 적절한 수치 데이터로 변환할 수 없을 경우에는 경고음을 내고 다시 입력을 하게 한다.

넷째로 그림 7은 사용자에게 메뉴형식의 질문을 하고, 사용자는 그 중의 2가지 이상을 선택할 수 있는 형태이다. 이 경우 선택은 스페이스(긴 글쇠)로 하고, 선택된 것은 색깔이 변하고 앞에 ‘*’가 표시되어 구분할 수 있도록 했다.

위와 같은 방법으로 각각의 질문형태에 맞는 인자들을 설정한 후에, 이들 multifield들을 hmenu의 인자로 넘겨주면 hmenu의 기능을 사용할 수 있다. 이때 이를 직접 사용하는 방법도

가능하지만 그림 8과 같은 rule을 만들어 놓고 사용하는 것이 편리하다. 아래의 rule은 이미 질문한 내용은 검사를 하여 더 이상 질문을 하지 않도록 하고 있다. 즉 모든 질문을 한번씩만 자동적으로 수행하게 하는 것이다.

② htitle

HCLIPS는 사용자가 전문가 시스템 구성시 쉽게 Title화면을 구성할 수 있는 기능을 제공한다. 이 기능을 수행하는 명령어가 htitle로써, 이의 사용법은 그림 9와 같다.

```

("재배면적"
 "질문내용"      "대상 지역의 재배 면적은 얼마입니까?"
 "예상되는답"
 "답의형태"      "수치"
 "예상되는수치범위"  -1000  1000
 "도움말"        "단위는 (정보)입니다. 1정보=3000평"
 "도움그림"      "질문이유")
    
```

Fig. 6 Question type 3 : User should put a number.

```

("재배작물"
 "질문내용"      "대상 지역의 처리할 주된 청과물은 무엇입니까?"
 "예상되는답"    "사과" "배" "감" "복숭아"
 "답의형태"      "복답"
 "예상되는수치범위"
 "도움말"        "처리할 청과물을 모두 고르시오."
 "도움그림"      "질문이유")
    
```

Fig. 7 Question type 4 : User can choose multiple answers.

```

;;; hmenu를 사용하기 위한 Rule의 한 예
(defrule interface-rule
  ?f <- (?x "질문내용" $?rest) ; hmenu를 사용하는 facts를 찾아냅니다.(MATCHING)
  ->
  (assert (asked ?x))           ; 한번 질문한 내용은 기억하여
                                ; 다시 질문하지 않도록 한다.
  (bind ?ans (hmenu $?rest))   ; hmenu를 구동하여, 그 결과를 돌려받는다.
                                ; 결과는 항상 multifield이다.
  (assert (?x ?ans)))          ; 사용자가 대답한 결과를 fact-list에 추가한다.
    
```

Fig. 8 Programming a rule to use 'hmenu'

```

;;; Example of a title-screen fact.
(deffacts title
  (fruit-title "제목" "청과물 종합처리시설 선정 시스템"
  "지역특성에 맞는 청과물 종합처리시설 선정을 위한 전문가 시스템"
  "서울대학교 농업생명과학대학"
  "농공학과 농업기계전공"
  "1992년 9월 7일"
  전화 : (0331) 290-2380 "그림" 0 0 "EGASPL.PCX")
  )
    
```

Fig. 9 An exemplary fact to use 'htitle'

여기서 fruit-title은 title의 이름이고, title 다음에는 큰 제목과 작은 제목의 두개의 인자가 오고, 다음에는 세개의 인자가 사용자가 임의의 내용을 담을 수 있는 공간을 위해 남겨져 있다. 보통 이곳에 시스템 구성자의 이름, 소속, 제작 날짜 등을 써 넣을 수 있고, 한 인자의 내용은 한 줄을 넘어도 수행되도록 되어 있다. 이들이 실제로 화면에 표시되는 곳은 화면의 오른쪽 가운데로 정해져 있다. 다음에 htitle은 화면 왼쪽에 그림을 표시할 수 있는 기능도 제공하는데, 이 기능을 사용하기 위해서는 "그림"이라는 예약어를 쓰고 그 이후에 그림이 표시될 상대적 위치, 표시될 그림 화일의 이름을 적어주면 된다. 이때 상대위치의 기준은 화면의 왼쪽 상단 1/3쯤의 위치이다. 이 htitle 기능 역시 시스템의 항상 先頭에서 실행되는 화면이라면 그림 10과 같은

rule을 구성하여 사용한다.

③ hpcxdisplay

hpcxdisplay은 사용자에게 그림을 보여줄 수 있는 기능을 제공하기 위한 것으로 그 사용법은 2가지가 있는데, 그림 11을 보면 첫번째는 현재의 커서위치에 그림을 보여주는 것이고, 두번째는 임의의 위치에 그림을 보여는 것이다. 이때 만약 그림이 화면의 하단에 위치하여 일부만 표시되면 자동적으로 화면이 위로 이동하여 전체를 볼 수 있도록 한다.

④ hyesno

hyesno는 사용자에게 [예, 아니오]의 응답이 기대될 때 사용하는 명령이다. 사용법은 hpcxdisplay와 같이 2가지가 있는데, 그림 12에 그 사용예가 있다. 예제 1에서와 같이 표시될 좌표가

```

::: This rule displays a title-screen.
(defrule display-title
  (declare (salience 1000))
  (? "제목" $?x)
  ->
  (htitle $?x))
    
```

Fig. 10 Programming a rule to use 'htitle'

```

[Example 1]
(hpcxdisplay "FRUIT.PCX")

[Example 2]
(hpcxdisplay 100 100 "FRUIT.PCX")
    
```

Fig. 11 Examples of 'hpcxdisplay' syntax

```

[Example 1]
(hyesno "지역특성에 맞는 청과물 종합처리시설 선정을 위한 전문가 시스템.
준비가 되겠습니까?")

[Example 2]
(hyesno 5 "지역특성에 맞는 청과물 종합처리시설 선정을 위한 전문가 시스템.
준비가 되겠습니까?")
    
```

Fig. 12 Examples of 'hyesno' syntax

지정되지 않으면 현재 줄의 가운데에 문답상자가 출력되면서 사용자에게 대답을 요구하게 된다. 예제 2에서는 임의의 위치에 문답상자가 출력하고 있다. 이때도 역시 화면의 조절이 필요하면 적절히 대응한다. 사용자의 응답은 각각 TRUE 또는 FALSE로 변환되어 시스템으로 넘겨진다.

htextdisplay는 사용자에게 긴 문장 또는 화일을 적절한 크기로 정리하여 보여주는 기능이다. 사용자에게 긴 문장을 보여주는 것과 화일을 보여주는 두 기능을 사용하는 법은 그림 13과 같다.

이 기능을 잘 활용할 수 있는 곳은 推論 후에 결과를 사용자에게 출력할 때이다. 이를 적절히 사용하여 사용자에게 결과를 출력하는 rule의 예는 그림 14에 있다.

⑤ htextdisplay

```
[Example 1]
(htextdisplay "...LONG text string...")

[Example 2]
(htextdisplay "화일" "textfilename")
```

Fig. 13 Examples of 'htextdisplay' syntax

```
(defrule print-out
  (declare (salience -5))
  (recom ?x cf ?y& : (> ?y 80) $?a)
->
  (open "results" results "a")
  (assert (open-results yes))
  (printout results crlf crlf "지금까지의 입력하신 정보에의하여 판단하면,")
  (printout results crlf "대상지역의 적정모델은"
    ?x "으로 하는것이 좋다고 생각합니다." crlf)
  (format results "판단의 신뢰도는 %5.1f 정도입니다." ?y)
  (if (> ?y 90)
    then
      (printout results crlf "다시 말하면, 모형" ?x "가 아주 적당하다고 생각합니다."
        crlf crlf))
  (if (> ?y 80)
    then
      (if (< ?y 90)
        then
          (printout results crlf "다시 말하면, 모형" ?x "가 적당하다고 생각합니다." crlf)))
  (close results))

(defrule show-results
  (declare (salience -6))
  ?open-results <- (open-results yes)
->
  (retract ?open-results)
  (htextdisplay file "results")
  (system "del results > NUL"))
```

Fig. 14 Program to display results using 'htextdisplay'

⑥ hcls

그림 1에서와 같이 HCLIPS가 시작하면서 나타나는 기본적인 출력창을 지우기 위한 명령어이다. 사용법은 일반 CLIPS 명령어와 같이 "(hcls)"로 사용한다.

III. 結 論

농업생산에 있어서의 전통적인 생산력 증가의 한 방법은 강한 힘을 지닌 농기계와 농약의 사용에 의존해왔다고 할 수 있다. 이러한 방법은 이미 그 한계를 드러내고 있다. 미래의 효율적 농업 생산 방법은 농업과 관련된 기존의 또는 새로운 지식의 효율적인 이용에 있다고 할 수 있다. 전문가 시스템은 판단과 의사결정에 필수적인 서로 다른 형태의 또는 애매한 지식들을 통합할 수 있기 때문에 이러한 일들을 가능하게 한다. 농업 전문가 시스템 개발을 비롯한 人工知能의 농업응용연구는 한국농업의 미래를 위해 필수적인 연구 분야 중 하나가 되어야 할 것이다.

따라서, hmenu를 비롯한 사용자 인터페이스 기능이 확장된 HCLIPS와 같은 전문가 시스템 개발도구를 사용하면, 농업 전문가 시스템 개발자는 사용자 인터페이스나 推論기관의 설계 등 부차적인 문제는 신경쓰지 않고 지식의 가공, 즉 지식베이스의 작성 또는 데이터베이스의 構築 등 좀 더 실질적인 시스템 개발에 精進할 수 있을 것이다. 이는 농업 전문가 시스템의 개발 기간의 단축과 개발 비용의 절감에 많은 기여를 할 수 있다고 생각한다. 또한 HCLIPS로 개발된 전문가 시스템은 컴퓨터 및 전문가 시스템의 사용에 익숙하지 않은 사용자도 손쉽게 사용할 수 있다.

앞으로 WorkStation상에서의 사용자 인터페이스도 개발할 예정인데, 그렇게 되면, HCLIPS를 사용하여 개발된 농업 전문가 시스템은 PC나 WorkStation, 어디에서나 아무 수정없이 HCLIPS내에서 바로 실행될 수 있게 된다. 이것은 전문가 시스템 개발 전용 도구를 이용하는 또 하나의 장점이 된다고 할 수 있다.

참 고 문 헌

1. Giarratano, Joseph C. and Gary Riley. 1989. Expert Systems. PWS-KENT Publishing Company, Boston, USA.
2. Huggins, Larry F., John R. Barrett and Don D. Jones. 1986. Expert Systems : Concepts and Opportunities. Agricultural Engineering, vol. 67(1) : 21-23.
3. Harrison, Patrick R. 1990. COMMON LISP AND ARTIFICIAL INTELLIGENCE. Prentice - Hall International Editions
4. Rewerts, Chris C., Bernie A. Engel, Joseph B. Rogers, and Don D. Jones. 1990. An End-User Interface for Agricultural Software. AI Applications, vol. 4(2) : 57-65.
5. Software Technology Branch, Lyndon B. Johnson Space Center(NASA). 1991. CLIPS Reference Manual Vol. I , II. COSMIC , Athens, GA, USA.
6. Software Technology Branch, Lyndon B. Johnson Space Center(NASA). 1991. CLIPS User's Guide. COSMIC , Athens, GA, USA.
7. 노상하, 이승구, 조성인, 이종환, 김만수. 1992. 청과물 종합유통처리시설의 현대화를 위한 기술 개발 및 보급 방안. 농림수산부.

부 록

```

/* HCLIPS.c 의 일부, 새로운 명령어를 추가하는 부분 */
VOID UserFunctions(void)
{
    DefineFunction("hpcxdisplay", 'v', PTIF hpcxdisplay, "hpcxdisplay");
    DefineFunction("htitle", 'm', PTIF htitle, "htitle");
    DefineFunction("hmenu", 'm', PTIF hmenu, "hmenu");
    DefineFunction("hcls", 'v', PTIF hcls, "hcls");
    DefineFunction("htextdisplay", 'v', PTIF htextdisplay, "hpcxdisplay");
    DefineFunction("hyesno", 'b', PTIF hyesno, "hyesno");
}

/* HMENU.c의 일부 */
/* 각각의 기능에 따라 적절한 함수를 호출하는 부분 */
void runhmenu(struct hmenu_t *hmenu_p, DATA_OBJECT_PTR returnvalue_p)
{
    hdrawhmenulayout(hmenu_p);
    switch( hmenu_p->VType)
    {
        case MULTIVALUED :
        case SINGLEVALUED : itemChoice(hmenu_p,returnvalue_p);
                            break;
        case NUMERIC      : getNumeric(returnvalue_p);
                            break;
        case MANUALED     : getManual(returnvalue_p);
                            break;
        default           : SetMultifieldErrorValue(returnvalue_p);
    }
    hantextemulation( );
}

/* hmenu 명령어와 Module 내부를 이어주는 부분 */
void hmenu(DATA_OBJECT_PTR returnvalue_p)
{
    DATA_OBJECT multifield;
    if(ArgCountCheck("hmenu", EXACTLY, 1) == -1) return;
    if(ArgTypeCheck("hmenu", 1, MULTIFIELD, &multifield) == 0) return;
    if(multifield2hmenufactor(&multifield, &hmenu_struct) == -1) return;
    runhmenu(&hmenu_struct, returnvalue_p);
}

```