

전문가시스템도구인 K-CLIPS에서의 지식표현

Representation of Knowledge in K-CLIPS - Expert System Development Tool

최 창 근 * 오 백 중 균 **
Choi, Chang-Koon Paik, Jong-Gyun

ABSTRACT

Structural design, generally engineering design, is a complex process combining design knowledges and analysis techniques. While analysis techniques can be automated in an algorithmic fashion, relatively little work has been done in the area of the design automation. An effect approach method for the automation of the engineering design may be a hybrid system, in which design knowledges, specification requirements and interpretations are represented using an expert system methodology and numerically intensive operations of the design process are implemented using an algorithmic language such as FORTRAN.

The purpose of this paper is concentrated on the knowledge of K-CLIPS(KAIST-C Language Integrated Production System) used to design and implement this hybrid system. In K-CLIPS, some representation methods ; frame, production rule, fact and user defined function are used to construct the knowledge base. The hierarchical knowledges could be expressed more naturally with a little number of rules than other plain production systems.

1. 서론

전문가시스템은 전문가의 지식과 경험을 체계적으로 구성하여 주어진 문제를 해결하고자 하는 컴퓨터 시스템이다. 이런 새로운 분야의 연구로는 질병진단, 팜물탐사, 컴퓨터시스템디자인 등의 문제에서 성공적인 결과를 나타내고 있으며 구조공학분야에서도 활발히 연구되고 있다. 현재까지 개발된 구조공학 분야의 대표적인 전문가시스템은 Maher의 논문[ref.1]에 체계적으로 정리되어 있다.

구조 설계분야의 문제는 예비설계, 구조해석, 세부설계, 컴퓨터 그래픽 등의 복잡한 과정으로 이루어져 있으며 지금까지 개발된 시스템은 대부분이 이러한 기능의 일부분만을 수행하도록 구성되어 있다. 따라서 설계과정이 하나의 통합된 형태를 지닌다면 편리한데 이런 형태의 전문가시스템을 구축할 경우에는 먼저 FORTRAN과의 연계성을 고려해야 한다. 이를 위해서는 전문가시스템개발용 도구가 연계성을 가지고 있어야 한다. 그러나 기존의 대부분의 도구는 규칙에 의해 지식을 표현하므로 비결정적인 문제해결의 접근에 용이한 반면, 수치해석면에서는 효과적으로 대처하지 못하며, 더구나 기존의 FORTRAN, C등의 알고리즘 언어로 된 프로그램과의 복합적인 시스템을 구축하기가 어렵게 되어있다.

본 논문은 구조설계의 통합된 전문가시스템인 I-BUILDS [ref 2,3]의 도구인 K-CLIPS (KAIST-C Language Integrated Production System)의 지식표현 형식을 설명하고자 한다. K-CLIPS는 여러가지 형태의 지식을 규칙과 사실로 쉽게 표현할 수 있고 계층적인 지식을 표현하기에 알맞은 프레임형태의 지식표현 및 K-CLIPS내부에서 새로운 함수의 정의가 가능하다. 그리고 K-CLIPS는 LISP언어에 비해 보다 이식성(portability)이 좋고 상당한 양의 수치처리가 가능한 C언어로 만들어져 있기 때문에 이미 개발되어 있는 유한요소해석 프로그램과 컴퓨터그래픽 프로그램 등과 복합된 통합시스템을 구축하는 것이 용이하다.

2. 공학분야의 지식 표현

공학의 실무에서는 공학의 이론과 경험적 정보를 이용한다. 공학의 이론은 해당분야의 기본적인 법칙과 이론을 바탕으로 하여 현상에 대한 설명을 가능하게 하며, 경험적 정보는 이러한 이론적 지식을 적절히 적용, 해석하는 경영의 단계에 해당된다. Harmon과 King [ref.4]은 이론적 지식을 'deep knowledge'라 했으며 경험적인 지식을 'surface knowledge'라 하였다. K-CLIPS 개발은 이런 두가지의 지식을 모두 보다 편리하게 표현하고자 하는 것을 목적으로 한다.

공학분야에서 나타나는 물리적 체계는 항상 많은 요소로 이루어진 복잡한 형상을 가진다. 이러한 구성 요소의 상호관련된 계층적인 구조는 명시적으로 표현하는 것이 타당하다. 물리적 체계에 관련한 지식은 크게 'static'과 'dynamic'한 지식으로 구분짓는데 [ref.5,6], 전자는 물리적 체계의 기하학적 형상, 재료의 성질, 하중상태, 조건, 가격등을 나타내는데 필요한 정보를 말하며, 이들은 주로 사실(facts)등으로 데이터베이스에 표현된다. 예를들면 구조 프레임은 보, 기둥과 이들의 위치, 단면성질, 연결상태, 부재력등의 사실적 지식으로 나타낸다. 후자는 물리적 체계와 관련한 특정한 문제를 어떻게 풀어 나갈 것인가에 대한 지식을 말하는데, 예를들면 부재를 설계한다던지 또는 구조해석을 수행한다던지 하는 일이다. 그러므로 문제가 확실히 이해가 가능하다면 dynamic한 지식은 이론을 근거로한 알고리즘에 주로 의존하게 된다. 그렇지만 문제의 이해가 불투명할 시에는 경험적 정보가 dynamic한 지식의 주종을 이루게 된다. 그러므로 dynamic한 지식은 문제의 유형에 따라 변하는데 반하여, static한 지식은 물체의 구성을 나타내므로 해당시스템에 관한 모든 문제에 대해서는 일정하게 된다.

2.1 Static Knowledge의 구성

가장 원초적인 형태의 static한 지식은 일련의 합당한 설명문으로 생각할 수 있다. 즉,

보 1은 기둥 1에 연결되어 있다.
기둥 1은 프레임 1의 부분이다.

* 정희원 한국과학기술원 토목공학과 교수
** 정희원 한국과학기술원 토목공학과 박사과정

이러한 지식을 표현하기 위하여 대부분의 규칙 기반 시스템에서는 다음에서와 같이 Object-Attribute-Value(O-A-V)형식의 개념을 이용하고 있다. 즉,

객체(Object)의 속성(Attribute)은 값(Value)이다.

여기서 객체, 속성, 그리고 값은 지식을 조절하기 위하여 사실을 다룰때 비교하기 위한 고유의 이름들이다. 이러한 표현기법은 모든 사실을 나타낼 수 있는 것은 아니지만 공학문제에서 나타나는 일반적인 관계 즉, '보(O)의 재료(A)는 콘크리트(V)이다.' 등은 쉽게 표현할 수 있는데, 각각의 구성에 대한 세부적인 설명은 다음과 같다.

객체 : 이는 물리적 실체 또는 개념등의 다른 형태의 정보를 나타내는데, 일반적인 사항이나 변수 또는 다른 객체를 가지는 특성이 있으면 어떠한 것이라도 객체가 될 수 있다. 구조공학에서는 '보', '프레임', '연결'등의 사항이외에도 '재료형태', '하중'등도 객체가 될 수 있다.

값 : 이는 객체를 묘사하는 수치나 특성을 나타내는데, 이러한 수치의 예로는 '콘크리트' 라는 객체를 묘사하는 사항으로서 탄성계수가 210kg/cm 이라는 사실을 나타낼 때는 '탄성계수'라는 사항을 속성으로 하여 '210'이라는 값을 가지게 할 수 있다. 특성의 예로는 객체간의 관계를 묘사하는 경우에 값으로서 다른 객체를 가지는 경우이다. 즉, '보'와 '콘크리트'가 각각의 객체인 경우에 '보'의 'made-of'라는 속성으로 '콘크리트'라는 값을 가지게 한다. 특별한 경우에는 값으로서, 일정한 일을 처리하게 하는 'demon'이라고 불리우는 함수를 가질 수도 있다.

속성 : 속성은 객체의 성격을 묘사하는 관계자로서 객체와 값, 객체와 객체간, 그리고 객체와 함수간의 관계를 묘사하는 역할을 한다.

이러한 O-A-V양식에 의한 static한 지식을 표현한 예를 그림 1 에서 보여준다.

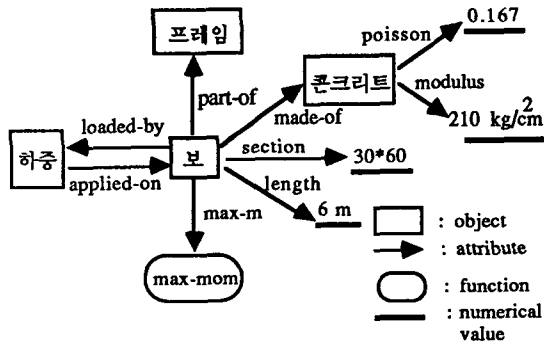


그림 1 O-A-V 양식에 의한 보 지식의 표현의 예

2.2 Dynamic Knowledge의 구성

Dynamic한 지식은 과학적이고 이론적인 법칙으로 된 고도의 지식으로부터, 오랫동안의 경험에서 우러나

온 체계적으로 구조화하기 힘든 지식에 이르기까지 넓은 영역으로 퍼져있다. 이러한 모든 범주의 지식을 표현하기에 적합한 특정한 표현기법은 아직까지는 발견하기가 어렵고, 실무에서 대두되는 여러가지 과학적 지식은 여러가지 형태로 표현할 수 있는데, 예를들면 해석문제에 관하여 지금까지 많이 이용된 알고리즘에 의한 절차식 표현이 해당된다. 이 예로 힘에 관한 보의 간단한 해석적 해답을 구하는 것으로부터 유한요소법, 경계요소법등의 복잡한 수치적인 코드에 이르기까지 다양하다.

그러나 경험에 의하여 생겨난 경험적 지식은 알고리즘식 지식의 표현으로는 곤란하므로, 그 해결책으로 가장 많이 이용되는 형태가 규칙(rules)을 사용한다. 이러한 규칙은 크게 두 개의 부분으로 구성되어 있는데 조건부와 행위부라 이른다. 조건부는 static한 지식베이스에 있는 사실에 대한 비교문을 담고 있는데, 이러한 사실이 참이면 행위부가 실행되는데, 이 행위부에 의하여 사실이 조정되고, 새로운 사실이 static한 지식베이스에 추가된다.

Static한 지식베이스와 규칙을 검토하고 적용시키는 일련의 작동행위를 하는 부분을 '추론기관'이라고 부른다. 추론기관의 작동원리에는 static한 지식베이스와 dynamic한 지식으로부터 현재의 사실에 비추어 적용가능한 규칙을 차례로 검토하여, 조건부가 만족되는 규칙들을 선정한 후, 이 중에서 일정한 규정에 의하여 적용시킬 규칙을 선택하고(이를 conflict resolution이라 한다.), 이를 적용시켜 static한 지식베이스의 사실을 변화시켜 나가는 과정을 사용자가 제시한 목표를 만족할 때까지 반복하는 전진추론방식과, 사용자가 원하는 목표를 만족시키는 행위부를 갖는 규칙을 선정하여 그 규칙의 조건부를 검토해 보고 요구되는 사실이 static한 지식베이스에 존재하면 목표를 만족시키는 것으로 보고, 없으면 이러한 조건부의 지식을 새로운 임시목표로 설정하여 위의 과정을 반복하는 후진추론방식이 있다. 여기서 주의할 사항은 규칙의 구성은 추론기관과는 별로 관계가 없다는 것이며 이 두가지의 방법은 일련의 초기조건에 대하여 동일한 결과를 주게되지만 실행시간에는 차이가 있다.

3. K-CLIPS의 구조

K-CLIPS는 NASA/Johnson space center의 Artificial intelligence section 에서 개발한 CLIPS [ref.7]를 바탕으로 하여 그 기능을 확장한 전진추론(forward chaining)방식의 전문가시스템 개발용 도구이다. K-CLIPS의 기본 구조는 주된 지식베이스와 사용자로 하여금 지식베이스를 조절하도록 해주는 일련의 보조모듈로 그림 2 와 같이 되어있다. 이 그림에서 볼 수 있듯이 지식베이스와 추론기관은 분리되어 있다. 이렇게 함으로써 단지 지식베이스만 교환하면 여러 분야에 적용이 가능하게 되어 있다. 지식베이스의 관리가 매우 용이하다.

다음은 K-CLIPS의 구조를 간단히 설명하고자 한다.

지식베이스에 있는 지식은 4개의 부류로 구분할 수 있는데 이는 다음과 같다.

- o Facts
- o Rules
- o Frames
- o User-defined-functions

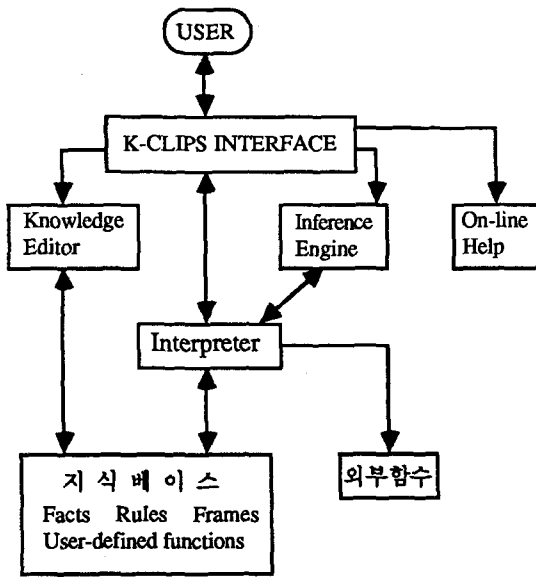


그림 2 K-CLIPS의 구조

따라서 지식기반 전문가 시스템을 개발하는 과정은 획득된 지식을 지식베이스 내의 4가지 형태로 표현하는 것이라 할 수 있다. 사용자는 사실(facts) 및 프레임으로 static한 지식을 표현하고 이를 조절하기 위한 규칙 및 함수로 이루어진 dynamic한 지식을 설정함으로써 특정분야의 문제해결을 지적으로 해결할 수 있는 프로그램을 작성할 수 있다.

지능형 에디터(knowledge editor)는 사용자로 하여금 지식베이스에 있는 지식을 불러내어 수정, 추가, 삭제 및 재배열 등을 가능하게 해준다. 현재 K-CLIPS에서 사용할 수 있는 에디터로는 MicroEmacs [ref.3]가 있다.

추론기관(inference engine)은 지식베이스에 내장된 지식을 처리하는 모듈로서 match-select-execute의 사이클로 구성된다. K-CLIPS의 추론기관은 Rete Algorithm [ref.8]을 바탕으로 구성되어 있어 보다 효율적으로 추론할 수 있다.

온 라인 help 기능은 시스템을 구축할 때 K-CLIPS의 정의된 함수들에 대해 보다 상세한 설명을 요구할 경우에 사용된다.

인터프리터(interpreter)는 지식을 단계적으로 테스트하면서 시스템을 구축할 수 있게 하는 모듈로서 K-CLIPS의 모든 모듈은 이것위에 존재한다.

외부함수는 필요시 FORTRAN, C, Ada 등의 언어로 된 프로그램으로 구성하여 전문가시스템 개발에 사용될 수 있다.

4. K-CLIPS의 지식표현

1) 사실(Facts)

K-CLIPS 시스템의 데이터 표현의 기본형태이며 추론과정에서 static한 지식의 저장은 working memory에 보관하게 되는데, 이는 K-CLIPS에 사용되는 사실들의 집합으로 이루어지며 이러한 사실에 대하여 규칙에 적용된다. 하나의 사실을 추가하는 assert, str_assert에 이루어지고 제거하는 것은 retract에 의해 이루어진

다. 그리고 여러개의 사실을 한번에 추가하는 일은 deffacts로 한다. 여기서 정의되는 사실은 리스트(list) 형식으로 표현된다. 그림 3은 이런 형식의 지식을 표현한다.

```

(deffacts Menu-Database
 (lateral moment-resisting R.C. 1 30 commercial )
 (lateral moment-resisting Steel 1 45 commercial )
 (lateral frame-shearwall R.C. 20 40 commercial )
 (lateral shearwall R.C 20 35 any ))

(assert (lateral frame-shearwall R.C. 25 70 residential ))

(str_assert ("vertical rcslab any any any any any "))
  
```

그림 3 사실(fact)에 의한 지식표현

2) 프레임 (Frames)

프레임 구조는 계층적인 지식을 표현하기에 매우 용이하다. 가장 상부레벨의 노드는 일반적인 개념을 표현하고 하부레벨로 내려갈수록 보다 세부적인 경우를 표현한다. 그리고 상부레벨의 노드가 갖는 속성은 하부레벨의 노드에 상속될 수 있다.[ref.9] K-CLIPS에서는 이 노드에 해당하는 역할을 하는 것을 클래스라고 부른다. 그림 4와 같은 계층적인 지식은 그림 5와 같이 표현된다.

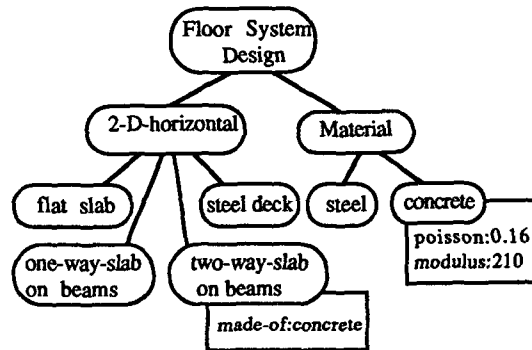


그림 4 계층적인 지식구조

```

(define_class Floor-System-Design (2-D-horizontal Material))

(define_class Material (steel concrete))
(create_slot concrete ^poisson 0.16 ^modulus 210)

(define_class 2-D-horizontal (flat-slab steel-deck one-way-slab two-way-slab))
(create_slot two-way-slab ^made-of concrete)
  
```

그림 5 프레임에 의한 지식표현

```

(defrule ana-CONS-con1 "constraints checks"
  (declare (salience 5000))
  (design-phase Generate-and-Select u)
  (sub-phase ?dir)
  ?rem <- (CANDIDATE ?ID STATUS ?status & ~ eliminated NAME ?dir $?rest)
  (or (CANDIDATE ?ID CONS UPLIFT ?ok1 &: (numberp ?ok1) &: (< ?ok1 0)
      (CANDIDATE ?ID CONS COMPO $? DB ?ok2 &: (numberp ?ok2) &: (< ?ok2 0) $?)
      (CANDIDATE ?ID CONS EVALU ?ok3 &: (numberp ?ok3) &: (< ?ok3 0) $?))
  =>
  (retract ?rem)
  (assert (CANDIDATE ?ID CONS STATUS eliminated NAME ?dir $?rest)))

```

그림 6 규칙에 의한 지식표현

3) 규칙 (Rules)

Dynamic한 지식표현의 기본방법으로 규칙은 조건 부분과 조건이 만족될 경우에 취해질 행위 부분으로 구성되어 있다. 전문가시스템을 개발하려는 사람은 관심분야의 문제를 풀기위한 방법으로 규칙을 정의해야 한다. 이런 까닭에 규칙기반 시스템이라고 불린다. 이런 규칙은 추론엔진에 의해 현재 적용가능한 규칙이 선정되고 또한 만족된 규칙의 행위부분을 행하게 된다. 규칙은 defrule에 의해 정의된다. 그림 6 은 건물의 제한조건을 체크하는 규칙을 나타내고 있다.

4) User Defined Functions

K-CLIPS에서는 기본적으로 제공하는 함수(예; assert, retract ...)를 사용하여 만든 새로운 함수를 사용자에게 제공하며 이러한 함수들은 defun에 의해 정의된다. 한번 정의된 함수는 K-CLIPS의 기본 함수와 동일하게 작업이 수행되어 진다. 이와같이 사용자가 정의한 함수를 사용하여 여러가지 기능을 수행할 수 있다. 첫째로 복잡한 수치적 계산을 간편하게 만들 수 있다. 둘째로 연속적인 명령을 하나의 함수에 정의함으로써 사용자와의 인터페이스(interface)를 편리하게 한다. 셋째로는 클래스 간의 복잡한 기능을 간편하게 한다. 이것의 사용 예를 그림 7 에서 간단히 나타냈다.

```

(defun my-function (arg1 arg2)
  (sqrt (+ arg1 arg2)))

(defrule test1 "
  (data ?a ?b)
  =>
  (bind ?length (my-function ?a ?b)
  (fprintout t " Square-root of " ?a " + " ?b " is " ?length
  crlf))

```

그림 7 User-defined-function의 사용 예

5. 예비설계의 예

여기서는 K-CLIPS의 지식표현방법에 의해 건물에 관한 예비설계를 수행한 예를 보인다. K-CLIPS의 사용은 대부분의 기종 (IBM 호환기종, VAX, UNIX, Macintosh, 기타 일반기종)에서 실행이 가능한 대, 본 예제의 실행은 SUN 3/50을 이용하였다. 예제의

수행은 철골구조물에 대해서 설명되어 있다.

예제의 대상 건물의 데이터 입력은 다음과 같이 이루어 진다.

- [Ques.1] Building ID.? : STEEL BUILDING
- [Ques.2] What's your name? : KAIST CIVIL DEPT.
- [Ques.3] Choose the occupancy type of the building among following types. [If necessary, type 'help'.]
 - * commercial * residential * industrial
 - * institutional * public * special
 Do you want commercial or residential? : commercial
- [Ques.4] Items of material are shown below.
 - * R.C. * Steel * Composite
 Do you want R.C. or Steel? : Steel
- [Ques.5] Which design method do you select?
 - * WSD(Working Stress Design)
 - * USD(Ultimate Strength Design)
 Do you want WSD or USD? : WSD
- [Ques.6] Enter information of height.
 - (a) Total number of floor levels? : 30
 - (b) Building height, first floor level to top of roof beam(cm)? : 9000
- [Ques.7] Enter data in the narrow-frame direction.
 - (a) Number of bays? : 6
 - (b) Overall dimensions(cm)? : 4320
- [Ques.8] Enter data in the wide-frame direction.
 - (a) Number of bays? : 5
 - (b) Overall dimensions(cm)? : 3000
- [Ques.9] Is there CORE?(Y/N) : N

앞에서 제시된 데이터로 선택가능한 구조 시스템으로는 모멘트 저항골조와 프레임 전단트러스가 선택되었다. 이러한 수평력 저항 시스템은 사용자가 입력한 그리드(7*6)상에 배치되고 건물형상에 대한 후보가 제시된다. 이들 후보는 분석-부재설계-평가의 단계로 각각 검토되고 선택단계에서 모두 제시된다[그림 8 참조]. 사용자는 이 단계에서 원하는 후보를 선택할 수 있다.

```

SUB-PHASE synthesis OF DIRECTION narrow IS STARTED !!!
-----
**>> Lateral system "frame-sheartruss" is selected.
**>> Lateral system "moment-resisting-frame" is selected.

[Ques.10] Enter loads applied on building :
o Dead Load(DL)(kg,m) : 300
o Live Load(LL)(kg,m) : 600
o Wind Load(WL)(kg,m) : 170

[Ques.11] Enter strength of steel [Fy>=2400](kg,cm)? : 4000
-----
SUB-PHASE selection OF DIRECTION narrow IS STARTED !!!
-----

[ SUM ] DRIFT SPEED FRAME BIAxIAL DEPTH CORE BLOCK COL COST UPLIFT FREQ.
**>> CANDIDATE gen1 : str. system : moment-resisting-frame
-2.273 -0.763 0.000 -1.000 .000 -0.56 0.00 0.000 -0.979 0.000 0.000 1.000
**>> CANDIDATE gen26 : str. system : frame-sheartruss
-2.255 -1.472 0.000 0.000 .000 -0.56 0.00 0.000 -1.222 0.000 0.000 1.000
**>> CANDIDATE gen27 : str. system : frame-sheartruss
-0.960 -0.767 0.000 0.000 .000 -0.56 0.00 0.000 -0.634 0.000 0.000 1.000

[Ques.12] How do you select what you want, by yourself or computer?
Type user or auto : auto
**>> CANDIDATE ID. : gen27 is selected!!!

```

그림 8 건물형상의 평가

6. 결론

본 논문에서는 K-CLIPS의 지식표현방식에 대해 논하였다. K-CLIPS의 지식표현방법을 정리하면 다음과 같다.

- 1) 프레임에 의해 계층적인 지식을 표현하기가 용이하고
- 2) 상부 클래스의 속성과 속성값을 하부 클래스에 상속할 수 있다.
- 3) 리스트 형태로 구성되는 사실에 의해 여러 가지의 데이터를 구성할 수 있다.
- 3) 규칙은 조건부분에서 프레임과 사실들로 구성된 데이터베이스와 패턴매치를 용이하게 하므로 문제해결이 쉽다.
- 4) 새로운 함수를 선언할 수 있는 기능을 가지므로 사용자가 이용하기 편리하다.

앞으로 K-CLIPS에 추가될 사항을 정리하면 다음과 같다.

- 1) 구축될 시스템이 커질 경우, 작업을 각 단계별로 분리하여 문제를 해결하는 분할 방식이 필요하다. 현재의 K-CLIPS 기능으로는 단지 문제의 순서에 따라 지식베이스의 출입순서를 조절할 수 밖에 없기 때문에 각 문제에 관련된 규칙만을 분리함으로써 추론시간을 단축시키는 기능을 추가하고
- 2) 사용자의 인터페이스를 효율적으로 하기 위하여

window/menu 방식을 제공하고 graphic library를 구축하여 기본 함수(primitives)를 제공함으로써 언제든지 요구될 경우에 사용할 수 있게 그 기능을 확장시킨다

- 3) 지금까지의 모든 지식은 정확하다는 가정하에 표현을 하였다. 그러나 지식은 항상 불확실성을 가지고 있으므로 이런 불확실성을 처리할 수 있는 기능이 추가 될 것이다.

참고 문헌

1. M.L. Maher, Expert System for Civil Engineering : Technology and Application, ASCE.
2. 한국과학기술원, 인공지능을 이용한 고도의 구조해석/설계용 전문가시스템의 개발(I), 과학기술처, 1988.
3. 한국과학기술원, 인공지능을 이용한 고도의 구조해석/설계용 전문가시스템의 개발(II), 과학기술처, 1989.
4. P.Harmon and D. King, Artificial Intelligence in Business : Expert System, John Wiley and Sons, New York, 1985.

5. F.Chehayeb, A Framework for Engineering Knowledge Representation and Problem Solving, Thesis of Ph.D Degree, Dept. of Civil Engineering, MIT, U.S.A., 1987.
6. Y.J. Paek and H. Adeli, Representation of Structural Design Knowledge in a Symbolic Language, Journal of Computing in Civil Engineering, Vol.2, No.4, October, 1988.
7. Joseph C. Giarrantano, CLIPS User's Guide, 1987.
8. Forgy,C, RETE : A fast algorithm for the many pattern/many object pattern match problems, Artificial Intelligence, Vol.19, 1982.
9. D.A.Waterman, A Guide to Expert System, Addison-Wesley, US, 1986.
10. Phillip J.L., William J.G., and Del D., Expert System Tools for Civil Engineering Applications, Expert Systems in Civil Engineering, 1986.
11. Sobelman and Krekelberg, Advanced C : Techniques and Applications, QUE, 1985.
12. D.Sriram, DENSTINY : A Model for Integrated Structural Design, Artificial Intelligence, Vol.1, No.2, pp.109-116, 1986.