

인공지능 (Artificial Intelligence) 을 이용한  
고도의 구조해석 / 설계용 전문가  
시스템의 개발 (Ⅲ)

Development of Expert Systems for Structural Analysis  
and Design with Artificial Intelligence (Ⅲ)

연구기관  
한국과학기술원

과 학 기 술 처

# 제 출 문

## 과 학 기 술 처 장 관 귀 하

본 보고서를 " 인공지능(Artificial Intelligence)을 이용한 고도의 구조해석/설계용 전문가 시스템의 개발(III) " 에 관한 연구의 최종보고서로 제출합니다.

1990년 6월 25일

주관연구기관명 : 한국과학기술원

총괄연구책임자 : 최 창 근

연 구 원 : 김 이 두

연 구 원 : 최 인 혁

연 구 원 : 김 한 수

연 구 원 : 백 중 균

# 요 약 문

## 1. 제 목

" 인공지능(Artificial Intelligence)을 이용한  
고도의 구조해석/설계용 전문가 시스템의 개발(III) "

## 2. 연구배경

건설산업은 우리나라 전체 GNP의 10%이상되는 중요한 산업분야이며, 특히 해외건설은 한 때 우리나라 경제의 견인차역할을 수행하였다. 그러나 최근에는 해외건설이 외형이나 외화가득율에서 국제경쟁력을 상실해가고 있어 사회적문제가 되고 있는바, 이는 국제건설시장에서 수주여건의 악화에도 그 원인이 있겠으나 근본적으로는 전반적인 기술의 후진성이 가장 큰 원인으로 지적된다. 이와같은 문제의 해결을 위해서는 기술개발을 통한 건설산업의 고도산업화가 그 해결의 첩경이라 할 것인바, 그 중에서도 고도의 구조해석/설계기술은 가장 기본적이고 핵심이 되는 분야이다.

현재 우리나라에서 이 분야의 전반적인 기술수준은 선진국에 비해 극히 초보적인 수준에 벗어나지 못하고 있고, 또한 구조해석/설계를 위한 외국의 소프트웨어(SAP, STRUDL, NASTRAN 등)를 도입하여 사용하고 있으나, 설계경험의 축적에 의한 설계자동화시스템은 아직 요원한 단계이다. 외국의 경우에는 컴퓨터 산업등의 자국내의 관련기술축적을 바탕으로 대학 및 연구소와의 협동으로 고도의 구조해석 및 설계시스템을 구축하였으며, 인공지능기법을 이용한 전문가시스템(Expert System)으로 전환하기 위하여 연구노력을 기울이고 있다.

이와같은 외국의 구조해석/설계의 프로그램의 도입,사용은 고가의 구입비이외에도 이 분야의 해외의존도를 심화시키고 독자적인 기술개발에 어려움을 주게된다. 또한 설계내용에 있어서도 외국에서 개발된 시스템을 도입하여 사용하기에는 우리나라의 독자적인 설계관행에 따라야 하므로 어려움이 따른다. 이러한 어려움으로 인하여 요즘은 건설산업의 각 분야에서는 전문적인 연구소를 설립하여 독자적인 기술개발에 노력을 배가하고 있다.

본 연구는 이러한 노력의 하나로서 자체연구노력으로 보완된 전문가시스템 구축용 도구인 K-CLIPS를 이용하여, 건설산업의 대표적인 분야로서 예들들 수 있는 건축물의 설계와 교량의 설계에 대한 전문가시스템을 구축함으로써 독자적인 건설기술의 개발에 이바지하고자한다.

### 3. 연구내용

본 연구는 컴퓨터에 의한 구조해석 및 설계분야에서 독창적인 기술을 개발하기 위하여 우리나라 설계관행에 맞는 건물설계의 일관시스템(BUILDS)을 첨단과학의 한 분야인 인공지능의 기법과 연계함으로써 지능형 설계시스템을 구축하고자 한다.

이를 위하여 본 연구에서는 다음과 같이 세개의 부분으로 진행한다.

첫째, 전문가시스템의 개발용 도구인 CLIPS를 개선하여 독자적인 도구로서 K-CLIPS를 구축한다.

둘째, K-CLIPS를 사용하여 기존의 BUILDS 프로그램을 건축구조물의 설계진행에 따라 구조설계에서 필요로 하는 지식 즉, 건물의 구조시스템의 선정, 건물의 모델링, 입력데이터의 이해와 필요한 데이터의 처리, 해석 및 설계결과의 분석, 검토 그리고 자료의 개선등에 관한 경험적 지식을 이용하여 설계를 자동적으로 지원해 나가는 I-BUILDS를 완성하고자 한다.

셋째, 교량설계분야에서의 응용을 위하여 예비설계를 위한 기존의 지식을 조사하여, 교량의 설계단계에서 주요한 사항, 즉, 상부구조형식, 가설공법 그리고 기초형식의 선정을 지원해 주는 전문가시스템(BADES)을 개발한다.

### 4. 보고서의 구성

본 보고서는 내용상 크게 네부분으로 구성된다.

첫째, 전문가시스템용 도구의 개발에 관련한 부분이며,  
둘째, 건물설계용 전문가 시스템(I-BUILDS)의 구축에 관한 부분,  
셋째, 교량설계용 전문가시스템(BADES)의 구축부분, 그리고  
네째, 위의 연구와 관련한 연구자료를 부록으로서 포함하고 있다.

## SUMMARY

This study serves as a starting point for the development of expert systems in the area of structural design. The resulting System, I-BUILDS and BADES, functions as a designer's assistant in the structural design for buildings and bridge respectively.

This study is composed of three main parts, that is, the construction of K-CLIPS, I-BUILDS and BADES.

The programming environment called K-CLIPS is under development for computer implementation, in which a mixed language programming is utilized to couple two other languages and which is being upgraded to expand its capabilities. This includes the knowledge representation of O-A-V Triplet and user interface facilities which embed the computer graphics library and window/menu systems.

I-BUILDS performs the overall processes of building design according to the systematic design model. The user takes part in the design process by inputting a few primitive items. I-BUILDS represents the design information using Object-Attribute-Value triplet and heuristic rules. In I-BUILDS the human knowledge acquired from experiences of experts in structural engineering is incorporated into the conventional building design system, called BUILDS, to form a knowledge-based building design system.

BADES is divided into several subsystems. BADES-S, BADES-C and BADES-F are for preliminary bridge design. The tasks controlled by BADES-S are determine the span length, and to compare and select the appropriate superstructure types, ranging from small to middle spanned bridge. BADES-C chooses the construction methods of the selected superstructure types, while BADES-F chooses the foundation types.

# CONTENTS

## PART I : Development of Expert System Tool

1. Introduction to Expert System	
1.1 Introduction .....	19
1.2 The Components of KBES .....	21
1.3 Knowledge Representaion .....	26
1.4 Control Strategy .....	28
1.5 Applications .....	32
1.5.1 Application Domain .....	32
1.5.2 Examples of KBES in Structural Engineering .....	35
1.6 Evaluation of KBES .....	48
2. Tools	
2.1 Classes of Tools .....	52
2.2 Conditions for Engineering Application .....	62
3. Development of K-CLIPS	
3.1 Introduction .....	64
3.2 Knowledge Representation by K-CLIPS .....	67
3.2.1 Facts .....	68
3.2.2 Production Rules .....	68
3.2.3 Frames .....	69
3.2.4 User-Defined Functions .....	74
3.3 Graphical Interface .....	77
3.3.1 Video Mode Control .....	77
3.3.2 Window Functions .....	80
3.3.3 Menu Functions .....	82
3.3.4 Graphic Functions .....	83
4. References .....	93

## PART II : Development of I-BUILDS

1. Introduction	
1.1 Purpose .....	97
1.2 Scope of Study .....	99
2. Overview of I-BUILDS	
2.1 Introduction .....	101
2.2 BUILDS .....	103
2.3 Design Model of I-BUILDS .....	111
2.4 Components of I-BUILDS .....	118
2.5 Computer Implementation .....	119
3. Building of I-BUILDS	
3.1 Design Process and Control Strategy .....	121
3.2 Consultation Parts in Structural Design .....	124
3.3 Structuring Facts .....	125
3.3.1 Concept .....	125
3.3.2 Global Facts .....	126
3.3.3 Local Facts .....	127
3.3.4 Menu Structure .....	127
3.4 Structuring Rules .....	128
3.4.1 Concept .....	128
3.4.2 Global Rules .....	130
3.4.3 Local Rules .....	132
3.5 Member Design and Pattern Matching .....	132
3.5.1 Section Selection and Pattern Matching .....	132
3.5.2 Section Database .....	133
4. KBES for Preliminary Design	
4.1 Introduction .....	135
4.1.1 Purpose .....	135
4.1.2 Process of Preliminary Design .....	136

4.1.3 Scope of Study .....	139
4.2 Design Model .....	141
4.3 Knowledge of Preliminary Design .....	142
4.3.1 Synthesis .....	142
4.3.2 Analysis .....	151
4.3.3 Evaluation .....	181
4.3.4 Selection .....	185
4.4 Conclusion .....	185
5. Idealized Model	
5.1 Introduction .....	186
5.1.1 Purpose .....	186
5.1.2 Scope .....	188
5.2 Modelling Building Structure .....	188
5.3 Construction of Model .....	190
5.3.1 Coordinate System .....	190
5.3.2 Material Properties .....	191
5.3.3 Sectional Properties .....	192
5.3.4 Loadings .....	194
5.3.5 Building Geometry .....	195
5.3.6 Load-Case .....	198
5.4 Conclusions .....	199
6. Detailed Design	
6.1 Preparation of Input Data File .....	200
6.1.1 Needs .....	200
6.1.2 Concept of Study .....	201
6.1.3 Structures of Input Data .....	202
6.1.4 Translation of Data .....	205
6.1.5 Conclusion .....	206
6.2 Checking Input Data File .....	206
6.2.1 Concept .....	207

6.2.2 Data Type Checking .....	207
6.2.3 Compatibility of Data .....	207
6.2.4 Conclusion .....	216
6.3 Structural Analysis .....	216
6.4 Review Results .....	216
6.5 Refine Model .....	216
<b>7. Example Application</b>	
7.1 General Application .....	217
7.2 Preliminary Design .....	224
7.2.1 Example .....	224
7.2.2 Conclusion .....	233
7.3 Idealized Model .....	234
7.3.1 Example .....	234
7.3.2 Conclusion .....	250
7.4 Structural Analysis .....	251
7.4.1 Example .....	251
7.4.2 Conclusion .....	257
<b>8. Conclusions</b> .....	<b>258</b>
<b>9. References</b> .....	<b>259</b>

### PART III : Development of BADES

<b>1. Introduction</b>	
1.1 General .....	267
1.2 Purpose .....	267
1.3 Background .....	268
1.4 Scope of Study .....	269
<b>2. Overview of BADES</b>	
2.1 Research Review .....	270

2.2 Introduction of BADES .....	271
2.3 Design Model .....	272
2.4 Components of BADES .....	274
2.5 Application Range .....	275
<b>3. Bridge Analysis and Design Expert System</b>	
3.1 BADES-S .....	277
3.1.1 Types of Super-Structures .....	277
3.1.2 System Implementation .....	279
3.2 BADES-C .....	285
3.2.1 Introduction .....	285
3.2.2 System Implementation .....	285
3.3 BADES-F .....	291
3.3.1 Introduction .....	291
3.3.2 System Implementation .....	292
4. Conclusion .....	298
5. References .....	300

#### PART IV : Appendices

1. K-CLIPS User Guide .....	305
2. Papers .....	411

# 목 차

## PART I : 전문가시스템 구축용 도구의 개발

1. 전문가 시스템의 개요	
1.1 서론 .....	19
1.2 전문가시스템의 구성.....	21
1.3 지식의 표현방법.....	26
1.4 제어전략방법 .....	28
1.5 응용분야 .....	32
1.5.1 전문가시스템이 적합한 분야 .....	32
1.5.2 구조분야의 적용 예 .....	35
1.6 전문가시스템의 기대 .....	48
2. 전문가시스템 구축용 도구	
2.1 도구의 종류 .....	52
2.2 공학문제에 적합한 전문가시스템용 도구의 조건.....	62
3. K-CLIPS의 개발	
3.1 개 요 .....	64
3.2 K-CLIPS의 지식표현방법 .....	67
3.2.1 사실 .....	68
3.2.2 규칙 .....	68
3.2.3 프레임 .....	69
3.2.4 사용자정의함수 .....	74
3.3 확장된 인터페이스 함수 .....	77
3.3.1 비디오모드 .....	77
3.3.2 윈도우함수 .....	80
3.3.3 메뉴함수 .....	82
3.3.4 그래픽함수 .....	83
4. 참고문헌 .....	93

## PART II : 건물설계 전문가 시스템

1. 서론	
1.1 연구의 목적 .....	97
1.2 연구의 범위 .....	99
2. I-BUILDS의 개요	
2.1 서론 .....	101
2.2 건물설계의 일관시스템(BUILDS) .....	103
2.3 건물의 설계모델 .....	111
2.4 I-BUILDS의 구성 .....	118
2.5 Computer Implementation .....	119
3. I-BUILDS의 구축	
3.1 건물의 설계과정과 탐색 .....	121
3.2 전문가시스템의 지원영역 .....	124
3.3 사실의 구성 .....	125
3.3.1 사실의 구성개념 .....	125
3.3.2 Global 사실 .....	126
3.3.3 Local 사실 .....	127
3.3.4 메뉴 트리 구조 .....	127
3.4 규칙의 구성 .....	128
3.4.1 규칙의 구성개념 .....	128
3.4.2 Global 규칙 .....	130
3.4.3 Local 규칙 .....	132
3.5 부재의 설계와 패턴매칭 .....	132
3.5.1 부재단면의 선택과 패턴매칭 .....	132
3.5.2 부재의 데이터베이스 .....	133
4. 예비설계용 전문가시스템	
4.1 기초사항 .....	135
4.1.1 예비설계의 목적 .....	135
4.1.2 예비설계의 진행 .....	136

4.1.3 연구의 범위 .....	139
4.2 예비설계의 모델 .....	141
4.3 예비설계의 지식 .....	142
4.3.1 종합단계 .....	142
4.3.2 분석단계 .....	151
4.3.3 평가단계 .....	181
4.3.4 선택단계 .....	185
4.4 소결론 .....	185
5. 구조물의 이상화	
5.1 서론 .....	186
5.1.1 연구의 필요성 .....	186
5.1.2 연구의 범위 .....	188
5.2 건물의 모델링 개념 .....	188
5.3 모델의 구축 .....	190
5.3.1 건물좌표계 데이터베이스 .....	190
5.3.2 재료성질 데이터베이스 .....	191
5.3.3 단면성질 데이터베이스 .....	192
5.3.4 건물 하중 데이터베이스 .....	194
5.3.5 건물형상 데이터베이스 .....	195
5.3.6 하중의 경우 .....	198
5.4 소결론 .....	199
6. 세부설계단계	
6.1 입력파일의 구성 .....	200
6.1.1 연구의 필요성 .....	200
6.1.2 입력파일의 작성개념 .....	201
6.1.3 입력파일의 구조 .....	202
6.1.4 데이터의 전달 .....	205
6.1.5 소결론 .....	206
6.2 입력파일의 검토 .....	206
6.2.1 개념 .....	207

6.2.2 데이터 타입의 검토 .....	207
6.2.3 데이터의 타당성 .....	207
6.2.4 소결론 .....	216
6.3 구조해석 .....	216
6.4 해석결과의 검토 .....	216
6.5 부재의 개선 .....	216
<b>7. 예제의 실행</b>	
7.1 일반적인 수행예 .....	217
7.2 예비설계의 예 .....	224
7.2.1 실행예 .....	224
7.2.2 결과토론 .....	233
7.3 구조물의 이상화 .....	234
7.3.1 실행예 .....	234
7.3.2 결과결론 .....	250
7.4 구조해석 .....	251
7.4.1 실행예 .....	251
7.4.2 결과결론 .....	257
<b>8. 결 론</b> .....	258
<b>9. 참고문헌</b> .....	259

### PART III : 교량설계 전문가 시스템

<b>1. 서 론</b>	
1.1 개 요 .....	267
1.2 연구목적 .....	267
1.3 연구배경 .....	268
1.4 연구범위 .....	269
<b>2. BADES의 배경</b>	
2.1 예비설계의 조사사항 .....	270

2.2	BADES의 개요	271
2.3	BADES의 설계모델	272
2.4	BADES의 구성	274
2.5	BADES의 범위	275
3.	BADES의 구축	
3.1	상부구조형식의 선정시스템(BADES-S)	277
3.1.1	상부구조형식의 선정기준	277
3.1.2	선정시스템의 구축	279
3.2	가설공법의 선정시스템(BADES-C)	285
3.2.1	개요	285
3.2.2	선정시스템의 구축	285
3.3	기초형식의 선정시스템(BADES-F)	291
3.3.1	개요	291
3.3.2	선정시스템의 구축	292
4.	결론	298
5.	참고문헌	300

#### PART IV : 부 록

1.	K-CLIPS의 사용법	305
2.	발표논문	411

## PART I : 전문가 시스템의 개요

## 1. 전문가 시스템

### 1.1 서론

인공지능은 사람이 할 수 있는 일을 컴퓨터로 할 수 있도록 하는 방법에 대한 연구를 하는 학문이라 할 수 있다. 이러한 인공지능의 연구분야로서는 시각 및 음성에 대한 인식, 자연언어의 이해, 그리고 특정한 영역의 문제풀이를 다루는 전문가 시스템등이 포함된다. '인공지능'에서 '인공(Artificial)'이란 말은 '자연(Natural)' 지능, 즉 사람이 생각하는 것에 대한 비교단어로서 기계에서 지능을 구현한다는 의미이며, '지능(Intelligence)'이란 말은 필요한 지식을 가지고 있으면서 요구에 따라 그 지식으로부터 추론과정을 통하여 어떠한 결론에 도달하는 처리를 한다는 것을 의미한다.

전문가시스템은 컴퓨터 프로그램을 통하여 지능적으로 문제를 해결하려는 오랫동안의 노력으로 그 성과가 실용화된 인공지능의 한 분야이다. 인공지능에 관한 연구의 초기에는 연구의 관심이 일반문제 풀이기 (General Problem Solver)에 집중되었다. 그러나 이러한 방향의 문제풀이는 많은 양의 지식을 다루게 되고, 상식이란 지식의 정의가 불명확한 이유로 특정한 분야의 지식을 다루는 방향으로 전환하게 되었는데 이것이 오늘날 우리가 전문가 시스템이라 하는 시초가 되었다. 지금까지의 이러한 인공지능의 기법을 적용한 연구의 분야는 대부분이 게임이나 정리의 증명과 같은 문제에 국한되어 왔는데, 이러한 초보적 단계에서 실생활에 근접하는 시도로서 전문가시스템이 주목을 받게 되었다.

전문가 시스템은 문제를 푸는데 있어서는 일반적인 문제풀이전략을 이용하지만 전문가 시스템이 문제해결을 지능적으로 하기 위해서는 해당분야에 대한 전문적인 지식이 요구된다. 이러한 전문가 시스템은 현장에서 사람들이 문제를 해결하는 방식과 똑

같은 방식으로 문제를 해결하려는 것이므로 이미 특정업무에 정통한 전문인의 지식과 기술을 구조화하여 보존할 수 있으며, 또한 이를 널리 배포하여 사용할 수 있기 때문에 그 실효성이 높게 기대된다.

전문가 시스템은 초기의 몇 개의 비교적 성공적인 전문가시스템이 알려짐으로서 주목을 받게 되었고, 이를 바탕으로 더욱 성공적인 시스템의 개발이 가능하리라는 확신으로 상당한 관심이 주목되었다. 이러한 초기의 시스템으로는 감염질병 진단시스템인 MYCIN, 지질학적 정보를 분석하는 PROSPECTOR, VAX 컴퓨터를 설계하는 R1 (XCON) [이상 참고문헌1] 등이 있다. 이러한 시스템의 성공은 그 전문분야의 경험자들이 푸는 전문성의 수준으로 문제를 해결하는 능력과 전문가 시스템이 초보자와 쉽게 대화가 가능하다는데 기인한다.

초기의 이러한 전문가 시스템은 프로그램 문장이 순차적으로 행하여지는 기존의 프로그래밍 기법을 이용하였는데, 이러한 경험을 바탕으로 전문가 시스템이라 불리우는 기법이 나타나게 되었다. 이 기법의 특징은 기존의 프로그램의 순차적인 성질을 탈피하여 문제풀이의 전략과 해당 문제에 대한 지식을 분리할 수 있다는 것이다. 기존의 프로그램의 특징과 새로운 전문가 시스템의 기법의 특성을 비교하면 표 1.1과 같다. 표 1.1을 바탕으로 전문가시스템의 특징을 요약하면 다음과 같다.

1. 전문지식과 그 지식의 제어가 분리되어 컴퓨터에 내장된다. 즉 문제를 해결하기 위하여 무엇에 관한 지식이 사용되었는가와 특정한 문제를 해결하기 위하여 그 지식이 어떻게 적용되었는가에 관한 두 개의 부분으로 시스템이 구분되어 있다.
2. 문제를 풀기 위하여 사용된 지식이 기본적으로 심볼 혹은 문자로 표현되어있고 수치적인 형태가 아니다.

표 1.1 기존의 프로그램과 전문가 시스템의 특성

기존의 프로그램	전문가 시스템
데이터의 표현과 사용	지식의 표현과 처리
지식과 제어의 통합	지식과 제어의 분리
알고리즘 과정	추론과정
데이터베이스의 처리	지식베이스의 처리
답의 유일성과 완전함	적절한 해결책
실행도중의 설명이 불가	실행과정 설명이 가능
수치처리 위주	심볼처리 위주

3. 전문가 시스템을 구축할 경우에는 지식과 지식을 처리하는 과정이 외형상으로 확실히 표현된다.

4. 전문가 시스템은 사람의 경험을 통하여 얻게되는 전문성과 판단력을 포함한다. 이러한 경험은 아주 귀중한 것이지만 대부분이 이론적으로 증명할 수가 없는 것들이다. 경험은 원래 기본적인 원리로부터 생겨난 것이지만 대부분은 직감에 의한 지식이다.

### 1.2 전문가 시스템의 구성

생성시스템(Production System) 이라고도 불리우는 전문가 시스템은 사람의 사고과정을 모방하여 다음과 같이 크게 세 부분으로 구성된다.

o 규칙 : 경험에 의하여 생겨난 상황의 처리방법을 나타내는 사항으로서, 조건부(LHS)는 규칙의 적용여부를 판단하기 위한 조건을 나열하며 실행부(RHS)는 규칙을

적용할 때 수행되어질 사항을 표시한다.

○ 데이터 베이스 : 데이터 베이스는 특정한 작업에 필요한 데이터를 가지고 있는 부분인데, 일부는 영구적으로 변하지 않는 지식을 담고 있고 (이를 지식베이스라고 함), 나머지는 현재 문제풀이의 상황에 관한 기록을 하는 부분으로 일의 진행에 따라서 변하는데 이를 Working Memory(Context) 라고 한다.

○ 제어 방법 : 이것은 규칙이 지식베이스와 비교되는 순서를 제어하고 일치되는 규칙이 여러개 있을 경우에는 어떠한 것을 우선적으로 적용할 것인지에 관한 문제를 해결하는 방법등에 관한 것이다.

이러한 요소들로 구성되는 전문가 시스템은 사람이 사고하는 과정을 유사하게 모방할 수 있으므로 전문영역에 속하는 실제적인 생활에 응용이 가능하다. 이러한 전문가 시스템의 구성요소들의 세부적인 역할과 그 관계를 나타내면 그림 1.1과 같이 표현된다.

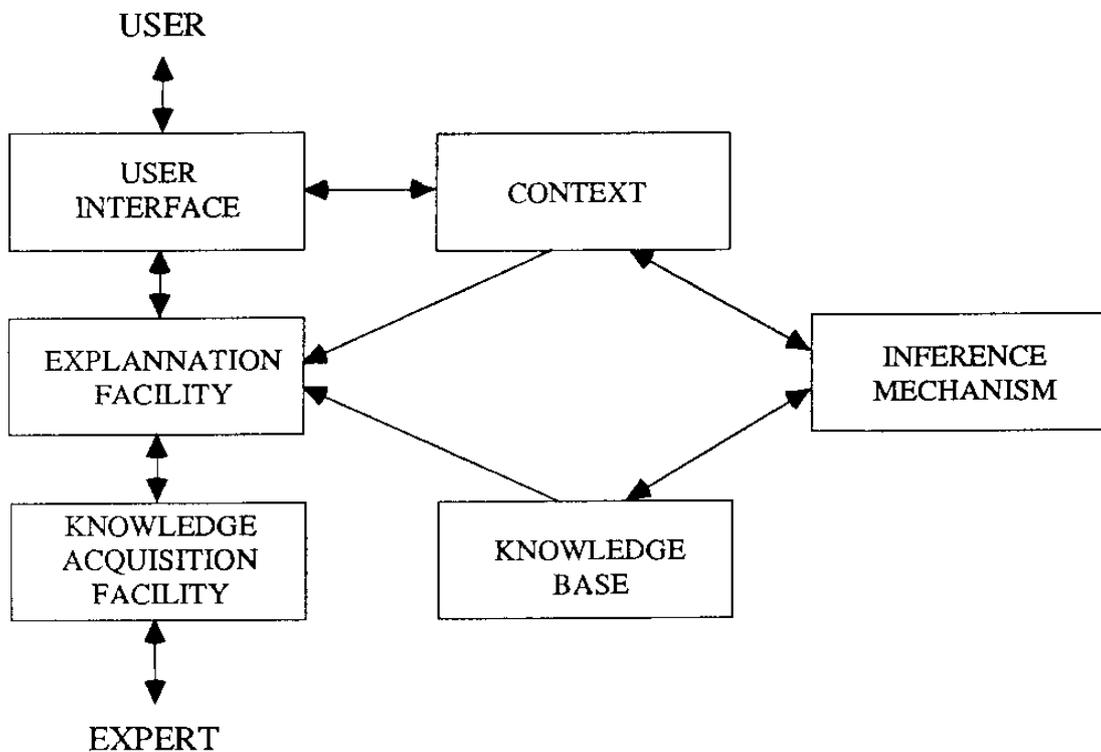


그림 1.1 전문가 시스템의 구성

그림 1.1을 보면 전문가 시스템의 기본구조는 전문분야의 지식을 담고 있는 지식베이스, 현재의 문제풀이상황을 기록하는 Context, 전문분야의 지식을 다루는 제어방법을 나타내는 추론기관으로 구성되고, 추가되는 기능으로는 전문가 시스템의 확장을 쉽게 하기 위한 지식획득시설, 사용자에게 편의를 제공하기위한 사용자 인터페이스 및 설명 기능등이 있다.

\* 지식베이스(Knowledge Base)

지식베이스에는 해결하려는 분야에 관한 지식들을 포함하게 되는데, 이를 production memory, long term memory, rule memory (rulebase) 라고 한다. 이러한 지식들은 그 성격상 전문적인 지식(Deep Knowledge)과 경험적인 지식(Surface Knowledge)으로 크게 분류가 된다. 전자는 정적평형조건등과 같이 기본적인 원리와 같은 부류의 지식이며, 후자는 전자와 같은 기본적인 원리로부터 경험에 의존하여 생겨난 것이다. 구조해석 및 부재의 설계는 Deep Knowledge에 가까우며 여러가지 구조물의 형상에서 후보를 선정하는 문제들은 Surface Knowledge 로 구분될 수 있다. 보통 Deep Knowledge는 기존의 알고리즘식 프로그램 언어인 FORTRAN, BASIC, C, PASCAL 등을 이용하여 함수(Function) 및 프로그램으로 표현이 되며, Surface Knowledge는 생성시스템(Production System)에서는 사실(Facts)과 규칙(Rules)으로 나타내어진다. 사실은 선언적인 지식으로 나타나는데 이는 물체, 개념등과 그들의 관계를 표현하는데 이용된다. 규칙은 조건부분(Condition 또는 Premise Part)과 실행부분(Action 또는 Consequence Part)으로 나누어지는데 실행부분이 작용하기 위해서는 조건 부분이 참(True) 값이 되어야 한다.

\* Context

Context 는 문제해결의 현재상태를 반영하는 사실(Facts) 들로 구성되어 있으며, 이를 working memory, short term memory, data memory (database)라 한다. Context 의 조직은 문제영역의 성질에 따라 달라지게 되며, 이들은 또한 해당문제의 해결과정 중에 수시로 변하면서 사실들이 추가 및 삭제가 된다. 이 내용들은 추론기관 (Inference Mechanism) 에 의하여 문제해결의 다음 단계를 결정하는데 영향을 주게된다.

\* 추론기관(Inference Mechanism)

추론기관 (inference engine)은 지식을 처리하는 사항으로, 이는 production memory 에 내장된 규칙을 이용하여 working memory에 내장된 사실을 조절한다. 전형적인 지식 베이스 전문가 시스템용 도구에서는 그림 1.2와 같은 과정을 반복하게 설계 구현되었다.

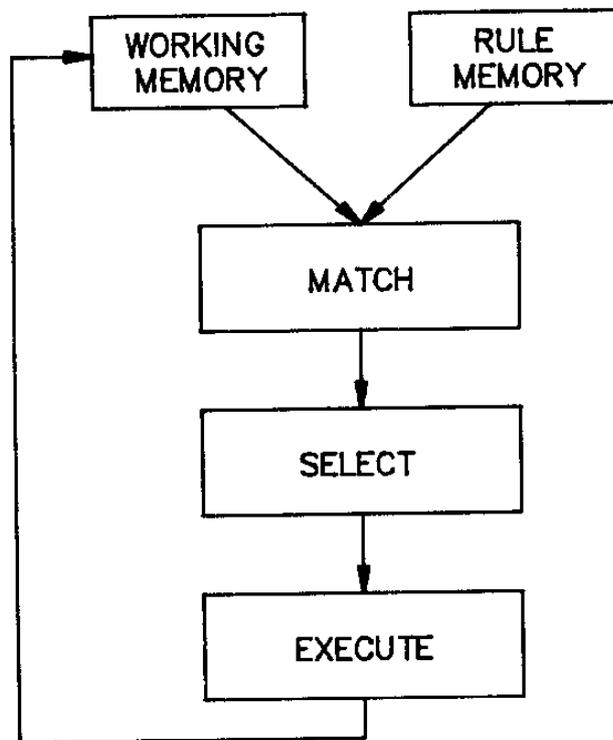


그림 1.2 추론 사이클

(1) working memory의 사실과 production memory의 규칙의 조건부분(left-hand-side)의 패턴을 비교하여 적용가능한 모든 규칙을 찾는다.

(2) 적용 가능한 규칙중 적절한 하나의 규칙을 선택한다.

(3) 그 규칙의 실행부분(Right-Hand-Side)을 적용하여 working memory 의 사실(facts)들을 변경시킨다.

\* 지식획득시스템(knowledge acquisition subsystem)

지식획득시스템은 전문가와 전문가 시스템용 도구와의 인터페이스 (interface)로서의 역할을 한다. 이는 전문가의 지식을 지식베이스에 입력하거나 필요한 경우에 내장된 지식들을 변경하는 작업을 가능하게 한다. 이러한 기능은 보통 사용이 간편한 에디터에 의해 제공되고 있다. 이를 사용하여 파일에 저장된 규칙을 수정하거나 추가할 수 있고 추론 과정에서 지식 베이스를 조절할 수 있다. 또한 전문가에 의해 지식의 획득이 어려울 경우 자체적으로 지식을 획득할 수 있는 기능(self-training facility)을 가질 수 있다.

\* 설명시스템(explanation subsystem)

설명시스템은 전문가 시스템의 추론 과정을 사용자에게 제시해 주는 기능으로 이는 전문가시스템의 실행과정에서 현재의 상태까지 이르게 된 경위 또는 현재의 질문이 필요한 이유에 대한 설명을 해주며 가장 보편적인 설명형식은 결론이 나오기까지의 유추과정을 거슬러 올라가는 형식을 취하는데 이는 사용자로부터의 질문에 대한 추론 결과를 어떻게 혹은 왜 선정하였는지에 대해 설명하고 추론에 관계된 일련의 규칙들을 제시해 준다. 이러한 설명시스템은 추론된 결과에 대한 의구심을 해소시킬 수 있고 기존의 프로그램의 양식보다 능동적인 자세를 취해 전문가 시스템이 전문가의 모습으

로 역할을 할 수 있게 되는 것이다.

\* 사용자 인터페이스(user interface)

사용자 인터페이스는 사용자에게 전문가 시스템용 도구사용을 위한 편의를 제공하는 것으로 전문가 시스템의 구현을 위한 기본적인 명령문을 제공하고, 사용자가 그 문체에 맞게 설계된 문장(syntax)을 쉽게 선택할 수 있게하며, 또는 컴퓨터 그래픽을 이용하여 사용자와의 대화를 증진시킨다.

### 1.3 지식의 표현방법

지식베이스에 내장될 지식의 표현 방법은 지금까지 개발된 KBES에서 사용되어왔던 여러가지 모델에서 그 형태를 찾아볼 수 있는데, 현재까지 실용화되어 많이 이용하는 지식표현의 방법들을 요약하면 다음과 같다.

- (1) Production Rule
- (2) Semantic Network
- (3) Frame 또는 Schema Representation
- (4) Logic Based Representation

\* Production Rule

하나의 Production Rule은 LHS(Left Hand Side) 와 RHS(Right Hand Side)의 쌍(Pair)으로 이루어진다. LHS 상의 심볼의 열거 (List)는 가정 또는 조건을 나타내는데 비해 RHS 상의 심볼의 열거는 행위를 나타낸다. 이러한 지식표현기법의 장점은 지식을 독립적인 규칙으로 나타내는 것이 가능하므로 각 모듈에 따라 패키지화할 수

있으며, 일정한 패턴하에 따라 주어진 상황에 대한 행위를 표현하므로 자연스럽다고 할 수 있다. 이 표현 기법의 단점으로는 적용될 규칙을 모두 체크해야 된다는 비효율성이 있고, 이의 실행은 기존의 언어로 된 프로그램의 실행과 유사하다는 것이다.

#### \* Semantic Network

의미망(Semantic Network)은 객체(Object)간의 관계를 강조하여 지식을 표현하는 기법이다. 의미망은 보통 Node와 Arc로 구성이 되는데, 노드는 보통 객체(Object), 개념(Concept), 상황(Situation) 등을 나타내고 Arc 는 그들간의 관계를 나타내게 된다. 이러한 표현의 장점은 계층적 구조로 관계를 표현하여 속성의 성질을 전달받을 수 있다는 것이다. 단점은 각 노드의 관계가 고유의 개체를 나타내는지 또는 그 부류를 지칭하는 지를 구별하기가 힘들며 상황에 따라 해결책을 강구하는 전문가의 접근방법을 표현하지 못한다는 것이다.

#### \* Frame(Schema) Representation

프레임 표현방법은 객체의 묘사에 중점을 두어 지식을 표현 하는 기법이다. 각 프레임은 프레임의 이름과 일련의 Slots 을 가지고 있는데 이 슬롯은 객체의 성질을 나타내거나 또는 다른 객체와의 연관(link)을 나타낸다. 이러한 링크는 하나의 프레임에서 다른 프레임으로의 정보의 전달이 가능하게 해준다. 이를 보통 Inheritance 라고 한다. 이러한 슬롯으로서 Meta- slots 이라는 것이 있는데 이러한 것은 다른 슬롯을 만드는 것에 관하거나, 또는 슬롯이 불러지고 만들어질 때 무엇을 할 것인가에 대한 정보를 포함한다. 슬롯중에서 Demon이라는 것이 있는데 이는 Interpreter 에서 감지를 하고 있다가 불러 질 때 갖고 있는 함수가 실행이 된다. 프레임은 Context의 사실뿐만 아니라 규칙을 표현하는 데도 이용된다.

#### \* Logic Based Representation

논리를 기초한 지식의 표현은 해당영역에 대한 지식을 논리로 표현되는 문장으로 나타난다. 이러한 논리는 보통 1 차의 Predicate 라는 논리로 표현되는데, 이는 정리 증명에 기초한 추론과정과 연결된다. 논리기초한 언어는 수량을 조절하는 문장을 사용할 수 있는데, 이의 대표적인 것으로는 Prolog를 들 수 있다. 이런 표현의 장점은 알려진 사항과 어떻게 그 지식이 사용되는가가 명확하다는 사실이다.

#### 1.4 제어 전략(Control Strategy) 방법

문제풀이라는 의미는 초기에 알려진 사실 또는 조건으로부터 구하고자하는 목표상태에 이르기까지의 풀이과정(Solution Path) 을 문제공간(Problem Space) 내에서 탐색하는 것을 뜻한다. 해를 찾기 위한 탐색을 문제풀이전략(Problem Solving Strategy) 이라고 하는데 이 방법에는 Blind Search처럼 문제 공간을 모두 탐색하는 Weak Method 와 해당문제에 대한 경험적 평가함수(Evaluation Function) 를 이용하여 탐색공간을 줄이는 Strong Method가 있다. 전문가 시스템에서는 탐색효율을 위하여 후자의 방법을 이용하며, 이러한 풀이전략은 전문가 시스템용 도구의 기법을 이용하여 컴퓨터에 내장되게 된다. 현재 KBES에서 개발되고 사용하는 문제풀이 제어전략들은 다음과 같다.

#### \* 전진추론(Forward Chaining)

전진추론은 Bottom-up, Data-driven 또는 Antecedent-driven 이라고도 하는데, 이 방법은 알고 있는 사실(Facts) 로부터 목표상태(Goal State)까지 도달하는 것으로 알고 있는 모든 사실들이 시스템에 입력되어 그 사실에 적합한 가장 좋은 결론을 도출

해낸다(그림 1.3참조). 이 방법은 입력되는 데이터는 적고 많은 가정(결론)이 있는 문제에 적합하다. 이 방법의 단점은 모든 조건들에 대한 가능한 사실들을 입력으로서 요구하는 것인데 때로는 이러한 사실들이 미지수인 경우도 있다. 때로는 풀이과정 중에 발생하는 사건(Event)에 의해 문제풀이 전략이 조절될 수 있는데 이러한 형식의 전진 추론을 Event- driven이라고 한다.

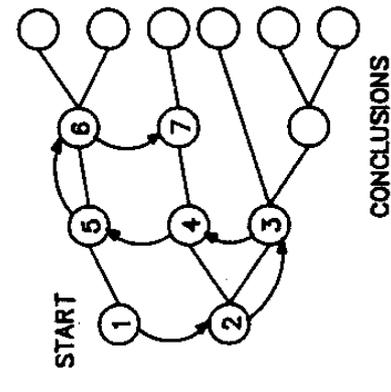
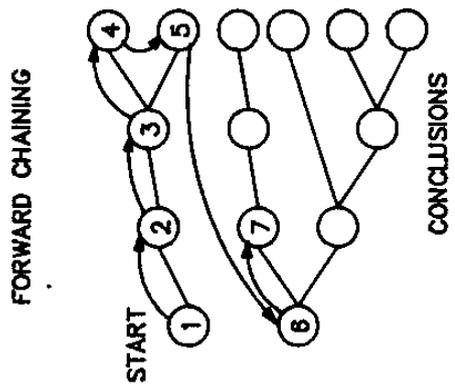
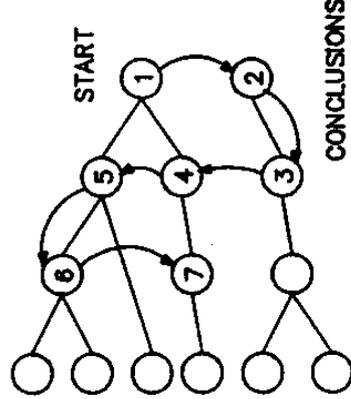
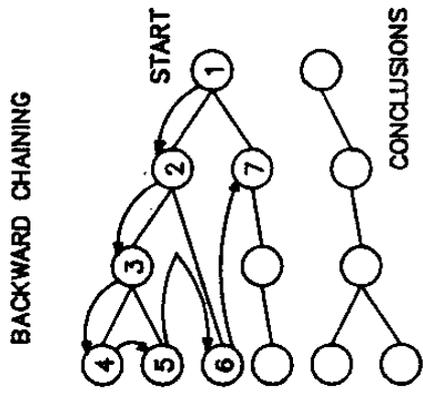
#### \* 후진추론(Backward Chaining)

Backward Chaining 은 Top-down, Goal-driven, Consequence-driven, 그리고 Hypothesis-driven 이라고도 한다. 이 방법은 가정(결론) 으로부터 시작하여 이 가정을 뒷받침해주는 데이터(Facts) 가 Context에 있는지를 체크해본다. 만약 지금까지 알려진 사실로서 이 가정을 지원해주지 못하면 필요한 사실들을 다시 Sub-goal로 생각하여 이 과정을 반복한다. 만약 가정이 증명되지 못하면 지식베이스에서 정해 놓은 순서에 따라 또다른 가정에 대하여 고려하게 된다(그림 1.3참조).

#### \* Means-Ends Analysis

이 방법은 현재의 상태와 목적상태의 차이점이 그 차이를 줄이는 가장 관련된 인자(Operator)를 찾는 데 이용된다. 이 Operator가 직접적으로 현재의 상태에 적용될 수 없다면 문제의 상태는 해당하는 Operator가 적용할 수 있도록 Sub-goal을 설정하면서 변하게 된다. means-ends 방법은 전진추론과 후진 추론 모두를 이용하게 되며 이 방법은 여러가지의 상태와 차이를 줄이고자 하는 인자간의 차이점이 우선적으로 정의가 될 수 있는 문제에만 적용할 수 있다.

#### \* Problem Reduction



**DEPTH--FIRST SEARCH**

**BREADTH--FIRST SEARCH**

그림 1.3 Forward chaining 과 Backward chaining

문제분할방식은 문제를 작은 몇개의 부문제(Sub-problems)로 나누어 해결을 하게 된다. 분리된 문제들은 AND-OR 그래프로 표시되는데, AND 노드는 일련의 계속되는 노드를 가르키는 Arc를 가지는데, 이들 노드들은 AND 노드가 참이되기 위해서는 모두 해결이 되어야 한다. OR 노드는 계속되는 노드중의 하나만 해결되어도 참 값을 가진다. 이러한 구조는 해당 문제에 대한 여러가지의 해가 있다는 것을 의미한다. 대부분의 경우에는 후진추론이 AND-OR그래프를 풀기위하여 사용된다. 이러한 기법은 크고 복잡한 문제의 해를 구할 때에 유용하다.

#### \* Plan-Generate-Test

기본적인 generate-test 전략은 모든 가능한 해를 만들고 이들을 대상으로 목표상태를 만족시키는 해가 있는지를 발견 할 때까지 각 해를 체크한다. plan-generate-test 방법은 가능한 해의 수를 탐색공간에서 일찍부터 절단함으로써 제한 시킨다. 이러한 절단은 plan 단계에서 행해지는데 이 때는 입력된 데이터에 의한 제한조건이 형성되어 데이터에 어긋나는 해는 버려지게 된다.

#### \* Backtracking

Backtracking은 부적당한 탐색의 길(Path)을 만났을 때 이전의 결정들을 되물리고 반복하는 것이다. 이러한 Backtracking 전략은 해를 찾는 길을 정의하는데 있어 추측이나 약산이 요구되는 경우에 이용된다.

#### \* Hierarchical Planning and Least Commitment Principle

계층적 계획의 원리는 계속되는 개념적계층으로 계획을 설립하는 것이다. 예를들면 복잡한 시스템의 설계에서는 설계공간이 Sub-system의 단계로 나누어지는데 높은 단

계의 사항들은 낮은 단계에 있는 세부사항의 개념적인 것들이다. 이러한 방식으로 시스템의 문제를 나누다 보면 최종적으로는 거의 독립적인 작은 시스템에 관한 문제가 된다. 각 Sub-system에 대한 해는 여러가지가 될 수 있다. 각 Sub-system에 대한 여러 변수를 결정하는 데 필요한 충분한 정보가 유용하지 않을 경우도 있고, 한 개의 작은 문제에 대한 해가 다른 작은 문제에 대한 결정에 영향을 받을 수도 있다. 이러한 연관성을 최소화하기 위하여 다른 시스템에서의 결정을 늦추는 게 중요하다. 이는 Least Commitment Principle이라고 불려진다. 변수의 결정은 문제공간에 대한 더 많은 정보가 얻어 질때까지는 행해지지 않는다.

#### \* Agenda Control

Agenda는 문제의 풀이기가 수행하여야 할 작업의 순서에 관한 리스트이다. Agenda 제어전략은 Agenda안에 있는 일들에 관해 우선권을 부여하는 일에 관한 것이다. 물론 우선권이 높은 일이 먼저 실행된다. 이러한 양식의 제어는 문제의 어떤 부분에 특별한 관심을 집중시키는 게 필요한 복잡한 문제를 다루기 위해서 사용된다. Agenda는 역시 여러가지의 독립적인 지식 원(Knowledge Source)이 서로 정보를 교환하게 되는 시스템에서 이용된다.

### 1.5. 응용분야

#### 1.5.1 Expert System이 적합한 분야

전문가 시스템의 정의는 사람의 전문성을 필요로 하는 의사 결정문제를 해결할 수 있는 컴퓨터 프로그램으로 일컬어지지만 그러한 문제의 범위는 사실 너무 광범위하다. 그러나 문제풀이에 대하여 잘 기록이 된 과정이나 알고리즘이 있는 의사 결정문제와

그렇지 않은 부분의 문제와는 구분을 할 수 있다. 후자의 영역이 전문가 시스템이 요구되는 분야라 할 수 있다.

이 영역의 문제는 그들을 풀기위해서 필요한 데이터의 불충분함과 정보에 대한 불확실성으로 특성이어질 수 있다. 보통 규칙으로 표현되는 경험을 사용하여 전문가는 이러한 데이터에 대한 차이를 좁힐 수 있고, 그리하여 해결책을 강구하게 된다. 전문가 시스템으로 이러한 분야의 문제를 해결하려고 한다면, 이는 전문가처럼 똑같은 기술을 사용하여 전문가를 모방하던지 또는 상응한 결과를 주는 의사결정과정을 적용하여야 한다. 그러므로 전문가 시스템은 데이터를 표현하고 조절하며, 불충분하며 불확실한 데이터를 다루어야 하며 그리고 경험적 규칙이나 주관적 판단을 할 수 있어야 한다. 실용적인 전문가 시스템을 구성하기 위해서는 전문가의 지식이나 방법론을 추출하는 것이 필요하지만 하상 가능한 것이 아니다. 오늘날의 전문가시스템에 대한 기법으로는 아주 좁은 분야의 전문성이 있는 문제에 대해서만 가능하다.

전문가 시스템의 구축여부나 적당한 문제에 대한 기준에 대해서는 많은 사항들이 있다. 첫째는 전문가 시스템이 가치가 있으려면 그의 능력이 그 해당분야의 전문가에 비교될만 하여야 한다. 둘째는 실행속도문제로서 기존의 빠른 계산속도와 전문가와 같은 시스템의 능력을 최대한 이용하여야 한다. 셋째는 전문가가 회귀한 분야에서 당장 어떤 결론을 요구하는 많은 분야가 있는 데, 이때는 전문가 시스템이 비전문가에게 필요한 결과를 주어 도와줄 수 있다. 넷째는 전문가라도 어떤 의견을 교환하고 정보와 데이터를 공유할 수 있는 지능을 가진 보조기구가 필요하다고 인식되는 분야의 문제에 적합하다. 마지막으로 현실적인 이유로 많은 문제에서는 전문가가 부족하며 또한 빠른 시간안에 훈련될 수 없기 때문에 전문가 시스템이 요구되는 절실한 이유가 된다.

전문가 시스템의 적용분야는 Derivation문제로부터 Formation 문제까지의 범주를

나누어진다. Derivation 문제에서는 해를 설명하는 부분으로서 문제의 조건이 제시되어 있는데, 이들은 얻어진 사실들로부터 해를 구하도록 규칙을 적용하는 데 있어 기준이 된다. Formation 문제에서는 문제의 조건이 해가 만족 시켜야 하는 성질의 형태로 주어진다. 실제 생활에서는 대부분의 문제는 이 극단적인 두 부류의 중간에 위치하게 된다.

\* Derivation 문제의 영역 예

o Interpretation : 주어진 데이터는 그들의 의미를 파악하기 위하여 분석되어진다. 예로는 Dipmeter Data를 분석하여 지질의 패턴이나 성향에 대한 정보를 추출하는 Dipmeter Advisor 나 광물질 탐사를 위한 PROSPECTOR[참고문헌 1] 등이 있다.

o Diagnosis : 이 문제는 데이터의 분석으로부터 시스템의 상태를 알아내는 부류의 문제이다. 이러한 부류의 문제로는 MYCIN[참고문헌 3] 과 같은 의학적인 진단문제를 들 수 있다.

o Monitoring : 신호를 계속적으로 분석하여 검사되어야 할 시스템의 상태에 따라 요구되는 변화가 이루어지는 문제이다.

\* Formation 문제의 영역 예

Formation 문제는 Generate-and-Test 방식의 예로 설명할 수 있는데, 이는 시스템의 한 부분에 의해 가능한 한 개의 해가 만들어지고, 다른 한 부분에 의하여 그것의 적합성이 시험된다. Formation 문제는 보통 두 개의 부류로 나눌 수 있는데, 이는 Constraint Satisfaction 과 Optimization 문제이다. Constraint 만족문제는 해가 일련의 제한조건을 만족하면 되는 것이고, 최적문제는 가장 좋은 해를 구하려고 하는 문제이다. 이러한 부류의 문제의 예로는 다음 사항을 들 수 있다.

o Planning : 어떠한 목표를 달성하기 위하여 요구되는 행동의 계획을 설정하는 문제이다. 이의 대표적인 예로는 MOLGEN이 있다.

o Design : 디자인은 어떠한 기능을 수행하기 위하여 물리적인 시스템을 선정하는 문제이다. 가장 성공적인 설계문제의 예로는 R1[참고문헌 1] 인데, 이는 VAX 컴퓨터의 형상을 설계하며 필요한 구성요소를 선택하고 그들의 배치와 연결문제도 고려한다.

### 1.5.2 구조분야의 적용 예

지난 십수년동안 많은 문제의 영역에서 전문가 시스템이 구축되었다. 이러한 전문가 시스템은 사용된 데이터베이스의 형태나 그들의 추론기관 및 지식의 표현방식에 따라서 구분을 할 수 있지만 구조공학분야에 대해서는 표 1.2과 같다. 구조공학분야라면 기계공학,지반공학, 건축공학등의 여러 분야를 망라하게 되는데, 본 연구에서는 전문가 시스템의 개발에 관련된 구조공학의 모든 분야를 포함한다. 구조공학이라면 구조시스템의 역학적원리를 실생활에 적용하는 분야라 할 수 있는데, 구조공학자의 정상적인 교육에는 재료, 강도와 서어비스 기능에 대한 구조시스템의 분석, 구조부재의 설계등을 포함하게 된다. 실무적인 구조공학에는 교육에서 제시되지 않지만 경험을 통하여 습득할 수 있는 많은 분야의 주제를 다루게 된다.

구조공학의 분야는 다음과 같은 분야로 세분할 수 있다.

1.) 재료문제 : 재료의 종류에 따라 시멘트, 철, 목재, 플라스틱, 흙, 그리고 복합재료등에 관한 연구의 영역

2.) 구조해석 : 하중, 정적해석, 동적해석 및 진동, 유한요소해석등의 분야

표 1.2 구조공학 분야의 전문가시스템

분류	이름	용도	언어	개발사	컴퓨터 기종
재료	WA	사용되는 재료나 용접의 형상에 따라 올바른 용접과정을 선택.	LOTUS 1-2-3	Stone & Webster	IBM PC
문제	WDA	용접의 결점을 진단하고 잘못된 용접이 재발되지 않도록 도움을 제공.	EXSYS	Stone & Webster	IBM PC
구조해석문제	SRAS	건물의 지진위험을 예비 평가.	DECISION FACTOR	Stanford Univ.	IBM PC
	DAPS	충격하중에 의한 방호건물의 피해 평가	EXSYS	US Air Force	IBM PC
	SACON	MARC라는 범용 구조해석 프로그램의 데이터 준비.	EMYCIN	Stanford Univ.	DEC 20
	SPERIL I	지진 피해를 평가.	C	Purdue Univ.	PDP 11/45
	SPERIL II	지진 피해를 평가.	PROLOG	Purdue Univ.	VAX 11/780
	CSUCL	구조물의 파괴모드, 피해정도 평가 추가하중에 대한 자문.	EXSYS	Minnesota Univ.	IBM PC
	FACS	FEM 프로그램 가이드.	PRISM	Rensselaer Poly. Univ.	IBM Minicomputer
	PLASHTRAN	MSC/NASTRAN 프로그램의 사용자 자문.	LOOPS	Berkeley	XEROX 1108
	CDA	항공기를 위한 sandwich Panel의 설계 자문	PLEXSAR	Lockheed	VAX
코드체크문제	AASHTO	단순지지된 고속도로 상의 교량 평가.	FORTRAN	Lehigh Univ.	CDC machine
	AMUBC	코드 사용률 자문.	PROLOG1	Sydney Univ.	SUN 3
	SPEC	건물의 부재 설계시 기준 체크.	Framekit	Carnegie-Mellon	VAX 11/750
구조시스템 설계	RETWALL	옹벽의 선택과 예비설계.	BUILD	Sydney Univ.	SUN 2
	BDES	교량의 결정, 모델링, 해석과정 자문.	PASCAL	Duke Univ.	IBM PC
	WISER	풍하중의 영향을 자문.	KEE	Houston Univ	Symbolics
	HI-RISE	상업용 및 주거용 건물의 예비설계.	PSRL	Carnegie-Mellon	VAX 11/750
	LOW-RISE	상업용 건물의 구조계획, 예비설계, 평가.	OPS5, C, LISP	Carnegie-Mellon	VAX 11/750
	ALL-RISE	건물의 예비설계.	SRL	Carnegie-Mellon	VAX 11/750
	FLODER	평면계획에 있어서 프레임의 그리드 결정	OPS5, LISP	Carnegie-Mellon	VAX 11/750
	HI-COST	건물의 구조시스템의 관한 견적 제공.	Framekit Rulekit	Carnegie-Mellon	VAX 11/750
	DENSITY	구조설계과정을 지원하는 통합된 시스템.	SRL	Carnegie-Mellon	VAX 11/750

3.) 코드체크 : 각 사용재료에 따른 법규의 만족도를 검토하는 문제와 관련된 것으로는 설비요건등에 관한 코드가 있다.

4.) 구조시스템 : 건물, 교량, 터널, 옹벽, 기초, 탱크등의 구조적인 문제에 관한 부분

5.) 기타 지원시설, 시설유지, 검사, 컴퓨터이용 도면작성등의 여러 영역이 있다.

지난 기간동안의 컴퓨터를 이용한 구조물의 설계는 구조해석 및 도면화에 치중되어 왔다. 그러나 전문가 시스템의 무한한 가능성은 기존의 컴퓨터 도움으로는 불가능했던 설계과정에 대한 분야에 보조기구로서의 역할을 가능하게 하였다. 역학적인 성질에 대하여 전문가와 상담하여 이러한 지식을 초보자에게 전달해주는 프로그램을 개발하려는 생각은 더욱 훌륭한 엔지니어를 가능케하고 구조시스템에 대한 신뢰를 높일 수 있을 것이다.

#### \* 재료문제의 영역

재료문제에 적용된 전문가 시스템으로서 두 가지 예가 이 절에서 제시되는데, 이는 모두 Stone and Webster Engineering에서 개발되어 상용적으로 이용되고 있다. 이들은 그 성질이 진단에 속하며 용접과 그 결점에 대한 전문성을 포함하고 있다.

#### o WA-Welding Advisor

많은 자료, 용접장비의 다양함, 복잡한 법령기준들 때문에 적절한 용접과정을 선택하는 것이 어렵게 되며, 더우기 필요한 용접물품에 대한 평가가 이루어지기전에, 즉 가격이 정해지기 전에 이러한 과정이 선택되어야 한다. 이 사항에 관한 전문가 시스템은 사용되는 재료나 용접의 형상에 따라 올바른 용접과정을 선택하는데 도움을 준다. 현재 이 시스템은 ASME 기준에 따른다. 이는 매크로기능을 광범위하게 사용할 수

있게한 LOTUS 1-2-3으로 구성되었으며, 매크로로서 코딩된 약 150개의 규칙과 평가를 위한 계산공식들을 포함하고 있다. 이 작업의 전문성은 Stone and Webster Engineering 의 용접전문인인 Bill Hathaway 와 인공지능 연구팀의 Gavin Finn 과 Martin Rooney 에 의하여 개발되었다.

#### o WDA-Weld Defect Advisor

구조물의 건설과정에서 발생하는 용접의 어려움으로 인하여 유발되는 시공연도에 의한 용접결점과 구조적인 요인에 의한 용접의 불충분한 사항을 진단하는 게 이 시스템의 목적이다. 이 시스템은 위의 사항에 대하여 잘못된 용접이 재발하지 않게 하기 위하여 도움을 준다. 시스템은 후진추론을 사용하며 EXSYS 라는 셸에 의해 구축되었으며, IBM-PC에서 실행이 가능하다.

#### \* 구조해석분야

본 절에서는 구조해석분야에 적용된 9 개의 전문가 시스템 에 대하여 설명하는데, 그 기능은 위험, 파괴분석으로부터 유한요소프로그램의 효율적인 사용에 대한 사항들을 포함한다. 이 분야는 기존의 프로그램과 대부분이 공유를 요하기 때문에 개발하는데는 진단부류의 문제보다는 많은 시간을 요하게 된다.

#### o SRAS-Seismic Risk Analysis System

이 시스템은 기존건물의 위험분석과 안전성에 관심이 있는 Stanford 대학의 연구진에 의하여 개발되었는데, 목적은 지반 의 운동, 구조물의 붕괴, 그리고 발생가능한 피해에 대한 사회적 영향등에 있어 전문가의 판단요소와 경험을 모델링하자는 것이다. 결과로는 기존의 건물의 지진위험을 예비평가하는데 있어 도움을 주는 전문가시스템인데 그 자문은 건물의 지진에 대한 여러 평가에 있어 유용하도록 계획되었다. 이 시

시스템은 DECIDING FACTOR 라는 셸에 의하여 구축이 되었으며 IBM-PC, XT, AT에서 실행이 가능하며 검증의 단계에 있다.

o DAPS - Damage Assessment of Protective Structures

DAPS는 지하벙커와 같은 방호건물이 강한 폭발에 대하여 생기는 피해를 평가하기 위하여 미국공군에 의하여 개발된 시스템이다. 방법은 피해의 직접적인 원인에 대한 요인과 피해의 정도를 나타내는 정도에 대하여 경험을 추출하였는데 이는 구조물의 기능이나 수리의 가능성에 대하여 판단을 하게한다. 이러한 구조물의 피해평가에 대한 데이터의 주관성으로 말미암아 시스템은 부정확한 피해요인이나 단계로서도 추론을 할 수 있게하고, 그의 진단에 대하여 확신성을 부여할 수 있게 하였다. DAPS는 EXSYS 라는 셸을 사용하였으며, 후진추론을 하며 IBM-AT에서 실행이 가능하다. DAPS의 특징이라면 Fuzzy Logic 을 사용하여 추론을 하는 능력이며 이는 계속 기능을 확대하고 있다.

o SACON - Expert System For Operating An Analysis Program

SACON 은 MARC라는 범용의 구조해석프로그램의 데이터의 준비를 도와주는 전문가 시스템인데, MARC의 사용자가 이에 익숙해지는 과정을 빨리하기 위하여 개발되었다. SACON 은 구조해석문제에 대하여 가장 좋은 모델링을 할 수 있게 도움을 준다. 이 시스템은 EMYCIN에 의하여 제공된 후진추론을 사용하며 약 170개의 규칙이 140여개의 자문변수와 더불어 지 식베이스를 구성하고 있다. SACON 은 1979년, 기존의 EMYCIN이라는 셸의 적용가능성을 시험하기 위하여 개발되었는데, EMYCIN이 의학분야외에도 적용이 가능하다는 판단이 나온 이후에는 개발이 중지되었다.

o SPERIL-I - Seismic Damage Analysis

강한 지진이후에는 건물은 그 피해정도가 분석되어지는데, 어떤 건물인 경우에는 그

피해의 정도가 시각적으로 판단하기 쉽지만 대부분의 건물은 피해의 정도가 분명하지 않아 건물수리를 하기 위해서는 전문가에 의하여 피해의 사실이 평가되어야 한다. SPERIL-I은 그러한 건물을 위하여 필요한 전문가 시스템이다. 이는 시각적인 데이터와 지진의 가속도등의 모든 가능한 정보로부터 가장 좋은 결론을 유도해낸다. 다른 지진피해 평가시스템처럼 확실성의 정도를 나타내준다. SPERIL-I은 Purdue대학에서 개발되었는데, 후진추론을 이용하며 C 언어로 코딩되었고 PDP 11/45 에서 실행이 가능하다.

#### o SPERIL-II - Seismic Damage Analysis

강한 지진이후에는 건물의 피해가 평가되어 건물이 수리될 것인지 또는 아예 무너뜨릴 것인지를 결정하여야 한다. 이 시스템은 SPERIL-I 이 발전된 것으로서 공학자의 이해와 의사결정을 돕는다. SPERIL-I과 다른 사항으로는 지식베이스의 차이로서 SPERIL-II 는 Meta-rule 을 사용하며 이러한 Meta- rule에 기초하여 규칙을 그룹화하여 각 그룹에서 전진추론이나 후진추론이 가능하다는 것이다.

#### o CSUSL-Concrete Structures Under Severe load

심한 하중을 받고 있는 방폭대피소와 같은 구조물의 해석이 극한강도에 기초하지 않게 설계된 기존의 건물보다 해석의 기법에 의하여 더 유리한 결과를 얻게 되는 것은 심각한 문제이다. 해석기법으로는 유한요소법이 보통 이용되는데, 이 방법의 단점은 작업의 수행중에 포함되는 비용과 재원인데, 정확도가 중요시되는 부류의 구조물의 해석을 수행하고자 하는 경우에는 더 간단한 것이 요구된다. 즉 이런 구조물은 하중이 없는 상황에서의 해석을 수행할 수 있을 뿐만아니라 하중이 가해진 이후에 대해서도 진단을 할 수 있는 시스템이 요구된다. 다른 진단용 전문가시스템에서와 같이 하중이 가해진 이후의 문제의 처리는 성질상 판단을 요구한다. 이 시스템은 Minnesota 대학

의 T. Krauthammer에 의하여 개발되었는데 전진추론을 이용하며, 시스템의 목적은 구조물의 파괴모드를 결정할 뿐만아니라 피해의 정도를 매기고, 그 구조물에 작용될 추가하중에 대한 자문을 해 주는 것이다.

#### o FACS - Finite Element Guide

항공기의 구조시스템의 설계를 하기 위해서는 개념화, 모델링 그리고 설계의 과정을 거치게 된다. 개념적인 단계에서는 기존의 막강한 FEM 을 이용하지 않게되는 이유는 기존의 전처리장치가 있다하더라도 기학적으로 표현되기보다는 수치적인 실체가 너무 복잡하게 다루어지기 때문이다. 그러므로 FEM에서 요구하는, 크기를 설정한 부재를 나타내는 방법은 너무 복잡하므로 비행기의 각 구성요소를 빨리 정의할 수 있는 기학적 정의와 이러한 정의를 FEM 프로그램에 맞게 모델링하는 전문성을 적용할 수 있는 전문가시스템이 요구된다. 이렇게함으로써 FEM 구축하는데 소모되는 긴 시간을 절약할 수 있을 것이다. FACS는 Rensselaer Polytechnic대학의 B. Gregory 와 M. Shephard 에 의해 개발되고 있는데, 이는 CAD 시스템에서 제공하는 구조체의 기학적 묘사외에도 해석모델에 관한 매뉴얼이나 전문지식을 포함함으로써 그 이상을 제공해준다. FACS 는 VM/CMS 운영체제를 가진 IBM 마이크로 컴퓨터에서 실행이 가능하다.

#### o PLASHTRAN - Two-Dimensional FEM Technique Consultant

FEM 패키지의 많은 능력에 부수되어 나타나는 것이 데이터를 준비하는 복잡한 과정이다. 이러한 해석프로그램을 사용하는 데는 다른 어려움외에 설계자는 설계->해석->결과체크->재설계->재해석등의 반복과정을 만족한 설계결과가 얻어질 때 까지 반복해야 한다. 이러한 일련의 과정이 쉬워질 때는 사용자가 해석프로그램의 능력뿐만 아니라 그 문제에 관한 전문가가 되었을 때이다. 그러나 이런 상태에 이르기까지는 많

은 시간을 요구하게 되므로 PLASHTRAN 은 MSC/NASTRAN 프로그램의 사용자를 돕기 위하여 개발되었다. 이는 설계문제를 모델링하고 이해하는 데 있어 공학자를 돕고, 교육시키기 위한 모델을 제시한다. PLASHTRAN 은 Object Oriented 언어인 LOOPS 로 구축되었는데, 이는 전진추론을 이용하므로 사용자의 입력으로부터 시작을 하게된다. PLASHTRAN 은 현재 하중의 경우, 구조요소, 재료 및 그들의 성질, 쉘구조물에서 비선형의 유무, 벽체구조의 배근방법등에 관한 자문을 해 주는 능력을 가진다.

#### o CDA - Composite Design Assistant System

항공공학은 재료와 그들의 성질, 해석기법, 그리고 경험등의 지식으로 표현되는 설계과정의 요소들을 포함하게 되는데, 이 분야에서는 새로운 재료, 해석기법이 나타나면 설계기법이 변하게 된다. 그러나 이러한 지식의 유입은 공학자가 이해하기에는 너무나 많으므로 가장 접근하기 쉬운 자료핸드북이나 설계 매뉴얼을 이용하게 된다. 이러한 기법에 따라가기 위해서는 새로운 상황을 보관하고 공학자를 돕는 운영시스템이 필요하게 된다. CDA 는 항공공학자를 돕는 전문가 시스템으로서 항공기를 위한 Sandwich Panel의 설계에 대한 것이다. CDA 는 PROLOG로된 PLEXSAR 을 이용하여 구축되었는데, Sussex 대학의 POPLOG라는 환경에서 실행이 가능하다.

#### \* 코드체크분야

본 절에서는 코드체크를 위한 3 개의 전문가시스템을 제시한다. 코드체크분야의 전문가시스템의 적용은 조건이나 규준을 정확하게 표현하고 이용할 수 있게 한다. 코드의 사용은 구조공학의 모든 분야에 사용되므로 아주 유용한 분야라할 수 있다.

#### o AASHTO - Bridge Rating Systems

교량설계는 구조공학에서 많이 다루는 문제로서 작업을 돕기 위하여 여러 종류의 프로그램이 개발되어있다. 그러나 이러한 시스템을 사용하려면 우선 사용자가 충분히 익숙해져야 하는 속성이 있다. 교량설계를 할 때 어려움은 설계의 제반기준이 너무 많아 산발적으로 주어졌고, 이전의 많은 경험적 데이터가 존재한다는 것이다. 이런 연유로 인하여 교량설계자는 특정한 설계를 할 경우에는 많은 옵션에 제약되어 더 좋은 해결책이 있는데도 불구하고 간단한 해결책을 선정하게 된다. 이 시스템은 교량설계에 관련된 모든 가능성 있는 입력과 설계를 하기위한 데이터베이스로 지식을 담고있다. 이 시스템은 RC덱크와 PS I-Beam 을 가진 단순지지된 고속도로의 교량을 평가한다. 또한 AASHTO의 교량평가법령과 전문지식, 그리고 의사결정에 관한 경험등을 포함한다. 이는 CDC 머시인상에서 실행이 가능하며 FORTRAN 으로 개발되었다.

#### o AMUBC - Australian Building Code System

기준은 일상의 공학자에게 지침서가 될뿐만 아니라, 많은 전문지식을 담고있다. 그러나 그 양은 많고 체계적으로 표현하려면 많은 노력을 기울여야 한다. 그러므로 특정한 기준을 이해하고 사용하려면 상당한 경험이 있어야한다. 이 시스템은 코드를 사용하는데 있어 공학자를 돕기 위한 목적으로 개발되었다. 이 시스템은 PROLOG 1셀로 구축되었으며, 8088/8086 마이크로 컴퓨터의 MS-DOS에서 실행이 가능하며 약 700 라인의 코드와 40K의 기억용량을 요한다.

#### o SPEX

기준과 시방서는 설계자가 코드에 적합한지를 검토하는데 사용된다. 해석이나 설계 변수의 선택에 있어 설계기준을 사용할 수 있는 많은 프로그램이 있지만 이러한 기준은 수시로 변하거나 다시 작성되므로 기준을 사용하는 과정과 기준의 내용을 독립시키는게 바람직하다. SPEX는 구조요소 설계를 위한 지식기반 기준체크 시스템이다.

이는 여러가지 지식의 재원을 사용하게 되므로 Blackboard 시스템을 이용하였으며, 최적을 위해서는 FORTRAN 언어를, 작업과정과 지식베이스를 이용하기 위해서는 Framekit를, 그리고 철골부재의 단면을 저장한 데이터베이스의 운영을 위하여 INGRES를 각각 사용하고 있다. SPEX는 Carnegie-Mellon 대학에서 일관된 건물의 설계와 건설 계획을 위하여 복합 프로그래밍 환경의 사용을 시험해보기 위하여 연구되고 있는 한 부분이다.

#### \* 구조시스템 분야

본 절에서는 구조시스템에 적용된 9 개의 전문가 시스템을 설명한다. 여기서 설명된 대부분의 시스템은 건물의 구조설계에 관련된 것이며, 이러한 시스템은 상용적인 목적보다는 연구목적으로 개발되었으며 설계지식과 설계과정의 표현에 더욱 관심이 주어졌다. 다른 시스템들은 옹벽 및 교량의 설계, 그리고 바람의 조건등에 대한 설계이다. 전문가 시스템이 구조설계를 위해 사용될 가능성은 설계문제의 복잡성에 따르는데, 대안이 적은 설계문제에서는 전문가 시스템방법이 선택과정에서 도움을 줄 수 있으며, 복잡한 설계문제에서는 현재의 기법으로는 구조물의 설계모델에 관한 이해를 증진시킬 수 있다.

#### o RETWALL - Retaining Wall Design System

기초공학자는 토질의 역학적 거동과 관련하여 어떤 구조물의 설계원리를 적용하게 된다. 그러나 토질의 복잡한 성질때문에 기초공학에서 설계문제를 풀기 위해서는 상당한 양의 지식이 필요하다. RETWALL 전문가 시스템은 옹벽구조물의 분야에서 이러한 경험을 제공하기 위하여 개발되었다. 이 시스템은 사용자의 입력에 따라 적절한 옹벽의 선택과 이 구조물의 예비설계를 수행하는데 자문을 해준다. 이 시스템은 BUILD

라는 셸에 의하여 구축되었는데, 이는 후진추론을 이용하며 Quentus Prolog로 되어있다.

o BDES - Bridge Design System

교량설계는 대안이 많은 분야의 문제인데, 교량설계라 할 경우에는 스패의 형태, 지간, 재료성질등을 포함하게 되는데, BDES는 North Carolina에서의 고속도로 교량의 결정, 모델링, 해석과 정을 돕기위하여 개발되었다. 이 시스템에서의 입력형태는 사용자가 교량의 기하학적 형상을 그래픽으로 표현하고 교량의 기능, 교량이 있는 주변환경을 정의한다. BDES는 Duke대학에서 J. Welch에 의해 Pascal언어로 구축되었는데 전진추론 규칙기반시스템이다.

o WISER - Design Modification of High-Rise Buildings for Serviceability

고층건물의 설계에는 더욱더 경제적이고, 보다 강하고 가벼운 건물의 재료를 이용하려는 경향이 있다. 이러한 건물의 새로운 거동형태는 바람의 영향에 대해서는 반드시 검토되어야 하는데, 이는 이전의 무거운 재료를 이용한 건물의 거동과는 거동이 다르기 때문이다. WISER 은 바람의 동적효과를 경험적인 지식베이스와 이론적인 근거를 바탕으로하여 측정하는데 있어, 공학자를 돕기위하여 개발되었다. 이 시스템은 LISP로 된 KEE 환경에서 개발되고 있는데 Symbolics 3640에서 실행된다.

o HI-RISE - Preliminary Design System for High-Rise Buildings

HI-RISE 는 형태가 사각형인 상업용 또는 주거용인 건물의 예비설계를 위한 전문가 시스템이다. 건물의 공간계획, 즉 Grid가 주어지면 HI-RISE 는 몇 개의 주요한 조건을 만족시키는 구조시스템을 택하게 된다. 건물의 예비설계를 하기 위해 서는 구조시스템 및 그의 구성요소들에 대한 이해가 있어야 하는데 HI-RISE 는 기존의 지식을 내장하여 이를 지원해주게 된다. 이는 VAX 상에서 실행되는 PSRL언어에 의하여

개발되었는데, PSRL은 프레임 및 규칙기반 추론을 가능하게 한다.

o LOW-RISE - Design System for Low-Rise Industrial Buildings

LOW-RISE는 산업용 건물의 구조계획, 예비설계, 그리고 평가를 돕는다. 계획이라는 것은 수평 및 수직프레임의 선택과 이를 사용자의 입력에 따라 적절히 배치하는 것을 말한다. LOW-RISE 는 OPS5, LISP 그리고 C언어를 사용하여 구축되었는데, 제어전략은 HI-RISE 와 비슷하다. 여기서 OPS5는 프레임의 선택 및 배치를 위하여 사용되었고, LISP는 해석과 같은 알고리즘에, 그리고 C 는 데이터베이스와의 정보교환에 이용되었다.

o ALL-RISE - Design System for Preliminary Design

ALL-RISE는 HI-RISE가 계속된 것으로 예비설계에 Grid의 입력을 요구하고, 설계의 계층적인 접근을 한다. ALL-RISE는 카네기 멜론대학에서 개발된 프레임기초 언어인 SRL 로 구성되었으며 이 시스템의 개발목적은 설계조건을 만족시키면서 설계대안을 종합하는 여러가지 방법에 관한 연구의 일환이었다.

o FLODER - Design System for Floor Framing Planning

FLODER는 주어진 평면계획에 대하여 프레임의 Grid를 정하는 전문가 시스템으로서 이 시스템의 목적은 전문가 시스템내의 Grid를 만드는 지식을 조사하여 건축적인 요소와 구조적인 사항이 구조물의 설계시 고려하고자 하는 것이다. 이는 HI-RISE의 한 부분으로 개발되었는데, OPS5와 LISP로 구축되었는데, 기본적인 OPS5로 되어있고 알고리즘은 LISP로 코딩되었다.

o HI-COST - Cost Estimating System for Preliminary Design

HI-COST 는 건물의 구조시스템에 관한 지식기반 견적시스템이다. 이는 구조시스템

의 대안을 평가하는데 있어 참고가 되고, 전문가 시스템과 데이터베이스와의 인터페이스를 위한 KADBASE의 사용을 설명하기 위하여 개발되었다. HI-COST 는 건설전문가의 지식베이스와 비용데이터베이스를 이용하여 건설비를 계산한다. 이는 Carnegie-Mellon 대학에서 개발된 프레임과 규칙기반언어인 Framekit와 Rulekit를 이용하여 구축되었다.

#### o DESTINY - An Integrated Design System

건물의 설계는 건축, 구조공학 그리고 기계공학등의 여러가지 분야의 협력으로 이루어지는데, 여기는 구조시스템의 지식뿐만 아니라 다른 여러 지식들이 수반된다. 그러므로 구조설계과정의 지원을 위해서는 통합된 컴퓨터 시스템이 요구되는데 DESTINY는 이러한 요구로 Blackboard 방법을 이용하여 현재 개념적인 구성을 하여 개발의 초기 단계에 있다.

#### \* 기타의 분야

이 절에서는 이상에서 언급하지 않은 4 개의 전문가 시스템에 대하여 설명한다. 하나는 건물의 유지에 관한 것이고, 두개는 기계공학 그리고 내번째는 컴퓨터를 이용한 교육용 프로그램이다.

#### o DAMP

건물에 미치는 물의 영향을 진단하는 표준적인 보편적인 방법은 건물의 검사자를 부르는 것이다. 건물의 검사자의 일을 경감시키기 위하여 DAMP라는 전문가 시스템이 개발되었는데, 이는 습기에 의해 유발된 피해에 대한 원인과 가능한 대책을 자문해 준다. DAMP는 사용자와의 인터페이스를 위하여 간단한 프롬프트와 대답으로 수행하며, 후진추론방법을 이용하고 약 150 개의 규칙이 내장되어 피해의 원인과 대책을 알

려준다.

#### o Pump-Pro

Pump-Pro는 펌프의 역학, 기계공, 그리고 펌프의 동작에 익숙하지 않은 사람을 돕기 위하여 개발된 진단용 프로그램으로서 Stone and Webster Engineering 사가 연구하였는데, 전진추론방법을 이용한다. 이 시스템은 Martin Rooney 에 의하여 개발된 MAID Service 라는 전진추론언어를 사용하여 구축되었는데 이는 마이크로 컴퓨터에서 실행이 가능하다.

#### o VA - Vibration Analysis

산업용 진동기기에 의한 진동문제의 진단은 구조공학자들이 접하게 되는 문제인데, 이 시스템의 입력은 스펙트럼 데이터로 주어져야 하므로 초보자를 위한 것이 아니다. 팬의 형상과같은 다른 데이터들은 스펙트럼에 따라 요구된다. 이는 VAX 11/780에서 실행되는 Mini-MYCIN과 IBM-PC에서 실행되는 EXSYS로 각각 구축되어 사용이 가능하다.

#### o ICAI - Moment Instruction Program

예비설계는 구조공학에서 다루기 힘든 분야로서, 이는 질적인 문제에 속한다. 그러므로 이 단계에서는 구조요소의 성질이 알려지지 않았으므로 유한요소 프로그램이 거의 이용되지 못한 다. 이 단계에서의 작업은 제안된 형상에 대해 수치해석보다는 경험적 방법에 의존하게 되는데, 구조설계에 경험이 없는 사람을 돕기 위하여 이 시스템이 개발되었다. 이는 전진추론 규칙기반 시스템인 GEPSE 라는 셸에서 구축이 되었고, C 언어를 이용하며 IBM-PC에서 실행이 가능하다.

### 1.6 전문가 시스템의 기대

전문가 시스템에 대한 기대는 현재 이용가능한 기술의 발달에 비추어 상당히 과장되어 있다. 이러한 과장된 기대로는 다음의 예를 들 수 있다.

○ 전문가 시스템은 현재 전문가가 다룰수 있는 모든 문제를 해결하기 위하여 개발될 수 있다.

○ 전문가 시스템은 지식공학자(Knowledge Engineer)가 가끔 전문가와 접촉하며 개발한다.

○ 전문가 시스템은 빠른 시간안에 원형(Prototype) 이 구성되어 확장될 수 있다.

○ 전문가 시스템은 현재의 모든 소프트웨어대한 문제의 해결책이다.

현재 사용가능한 기법으로 지식이 쉽게 표현되거나 변경될 수 있고, 이러한 지식이 수치일 필요가 없는 높은 단계의 프로그래밍 환경이 제공된다는 것이 가능한 전문가 시스템의 설명에서 이러한 기대가 일어나게 된다. 전문가 시스템을 설계하거나 구성하는데 있어 현재 사용가능한 기법들은 물론 이러한 특성을 가지지만, 다음의 각 항에서 언급하는 바와 같이 한정된 능력을 가지고 있다.

1. 전문가 시스템은 현재 전문가가 다룰수 있는 모든 문제를 해결하기 위하여 개발될 수 있는 주장에 대해서는 전문가 시스템의 기법을 사용하기에 적절한 문제를 풀기 위하여 전문성을 요구하는 분야이지 전문성이 있다고 모두 구성할 수 있는게 아니다. 인간의 전문가에 의하여 해결이 가능한 많은 문제는 현재의 전문가 시스템의 기법으로는 쉽게 풀리지 않는다. 전문가 시스템을 적용하기 위하여 적절한 문제를 선택하는 기준으로는

1) 문제가 좁은 전문 분야이어야 하고, 많은 상식의 지식을 포함하지 않아야 하며,

- 2) 문제는 전문가의 견지에서 너무 쉽거나 너무 어렵지 않아야 하며,
- 3) 문제가 명확하게 정의되어야 하고 그리고
- 4) 문제에 관한 전문가의 협력이 있어야 한다.

2. 전문가 시스템은 지식공학자(Knowledge Engineer)가 가끔 전문가와 접촉하며 개발한다는 사항에 대해서는 실제로는 많은 전문가 시스템이 이 일에 종사하는 일련의 팀에 의하여 개발되고 있다. 이러한 팀의 일부는 지식의 표현기법과 이러한 표현들을 내장하는 도구를 잘 아는 지식공학자를 포함한다. 다른 일부는 전문가로서 지식을 제공하게 된다. 성공적인 개발을 위해서는 팀의 두 부분의 협력이 요구된다. 지식공학자는 전문가와 정기적으로 또는 필요할 때마다 접촉하여야 한다. 이는 지식공학자가 일반적으로 문헌등으로부터 필요한 지식을 올바르게 추출하였다고 볼 수 없기 때문이다.

3. 전문가 시스템은 빠른 시간안에 원형(Prototype)이 구성되어 확장될 수 있다는 이 설명은 한정된 상황아래서는 맞을 수 있지만 모든 상황을 고려할 때는 옳다고 볼 수 없다. 만약 시스템이 해결하려는 문제가 명확히 알려지고, 그리고 Prototype을 위하여 선정된 도구가 원하는 문제풀이의 형태에 적합할 경우에는 전문가 시스템은 빠른 시간안에 원형이 구성될 수 있다. 그러나 대부분의 경우에는 풀고자 하는 문제가 애매 모호하게 정의되고 원형을 위해 선정된 도구는 값싸고 PC수준에서 실행이 가능한 것들이다. 전문가 시스템은 지식과 기능의 확장이 용이하여야 한다. 그러나 지식의 표현에 관한 현재의 기법으로는 시스템의 확장이나 유지를 위하여 전문가 시스템을 개발하기 위하여 사용된 도구나 언어에 충분히 익숙한 사람을 필요로 한다.

4. 전문가 시스템은 현재의 모든 소프트웨어대한 문제의 해결책이라는 이 사항은 해결 가능한 문제의 형태에 대한 깊은 고려도 없이 많은 다른 영역에서 전문가 시스템을 개발하기 위해 제안된 많은 의견이나 제약에서 나타나는 기대일 뿐이다. 많은 경우에

는 전문가 시스템이 이러한 문제들을 풀 수 있을 것이라는 기대를 가지고 소프트웨어의 문제가 거론된다. 그러나 전문가 시스템은 제한된 상황에서만 유용하며, 소프트웨어 문제의 성질이나 소프트웨어의 기술의 현재 상태에 대해서는 상당히 고려해 보아야 한다.

## 2. Expert System 구축용 도구

### 2.1 도구의 종류

전문가 시스템 개발용 도구(tools) 라는 것은 전문가 시스템을 구축하기 위한 작업을 간편화시켜 주는 프로그램시스템이다. 전문가 시스템의 구축을 위하여 사용되는 프로그래밍언어로는 보편적으로 알려져 있는 기존의 FORTRAN, PASCAL, C 등의 문제 중심의 언어(Problem Oriented Language)와 소위 인공지능형 언어라는 LISP, PROLOG의 심볼처리언어(Symbol Manipulation Language)가 있다. 문제 중심의 언어는 특별한 문제의 부류에 적합하게 설계되었는데, 예를들면 FORTRAN이 수치계산을 행하기에 편리한 기능을 가지므로 과학, 수학, 통계처리분야에서 많이 이용되고 있는 것을 볼 수 있다.

심볼처리언어는 인공지능의 적용을 위하여 개발된 언어로서 LISP는 리스트구조(List Structure)의 형태로 된 심볼을 처리하기 위한 조직(Mechanism)으로 되어 있다. 여기서 리스트는 항목들이 괄호(Parenthesis) 로 둘러싸여 있는데, 이러한 항목들은 심볼이거나 다른 리스트가 될 수도 있다. 리스트구조는 복잡한 개념을 쉽게 나타낼 수 있는 방법이다. 비록 PROLOG 가 조금의 각광을 받고 있는 것은 사실이지만 전문가 시스템의 구성을 위하여 가장 많이 사용되고 있는 언어는 역시 LISP이다. 몇개의 전문가 시스템에서 FORTRAN, PASCAL, C언어 등의 문제 중심의 언어를 사용한 예가 있지만 인공지능분야의 작업에서는 역시 LISP, PROLOG와 같은 심볼처리언어가 더 적당하다고 할 수 있다.

리스트의 선호도는 언어가 가진 몇가지 특징에 기인한다고 볼 수 있는데, 그 특징으로는 쉽고 다양한 심볼처리, 기억장소 운영의 자동화, 에디터 및 디버깅작업을 도와주

는 편의시설, 프로그램 코드와 데이터의 일관된 처리등을 들 수 있다. LISP 언어는 그 언어만 전용적으로 빨리 효율적으로 처리하도록 설계된 LISP Machine을 포함하여 컴퓨터의 종류에 따라 실행할 수 있는 여러 계통의 다양한 종류가 있는데, 그 중에서 가장 주된 것은 INTERLISP 와 MACLISP 이다. 현재는 모든 리스프를 통합하는 Common LISP 가 LISP 언어의 기준이 되고 있다.

리스프언어와 같은 프로그램 언어를 이용하면 전문가 시스템의 구축을 요구사항에 따라서 아무런 제약없이 할 수 있지만 지식베이스를 구성하기 위한 지식 및 추론기관 등을 어떻게 표현하는지에 대한 세부적인 사항에 대해 이해할만큼 사용자가 리스프에 충분히 익숙해져야 한다. 이러한 과정을 간단하게 하기 위하여 전문가 시스템의 구축을 돕는 것으로서 지식의 표현방법 및 추론기관을 미리 구현하여 사용자는 단지 해당 분야의 전문지식만을 내장함으로써 시스템을 구성할 수 있게 해 놓은 것이 도구라 할 수 있다. 근래의 전문가 시스템의 활발한 연구는 이러한 전문가 시스템의 개발을 위한 개발도구의 보급이 큰 도움이 된 것은 누구나 부인할 수 없는 사실이다. 이러한 전문가 시스템 개발용 tool은 그 도구로 가능한 작업의 부류에 따라 다음과 같이 나누어 볼 수 있다.

- o Expert Systems Shells(Skeletal System)
- o High-Level Programming Language
- o Mixed Programming Environments
- o Rule Induction Tools

### Expert Systems Shells

셸에 관한 최초의 작품은 혈청감염질환을 진단하는 전문가 시스템인 MYCIN

(Buchanan, Shortliffe 1984)[참고문헌 3]을 들 수 있다. 이 연구의 일부분으로서는 지식베이스를 추론기관과 독립시키려는, 즉 모듈화하려는 노력이 있었는데 이 노력의 결과로 감염질환을 진단하는데 사용된 지식베이스를 제거하는 것이 가능하게 되었다. 이를 보통 Empty MYCIN 또는 Essencial MYCIN 이라고 불리우는 EMYCIN이다. 이러한 EMYCIN을 사용하는 경우에는 사용자는 지식의 획득과 이를 프로그램 형태로 저장하려는 노력만 기울이면 가능하게 되었다. 구조공학에서의 이러한 노력은 한 예로서는 SACON 을 예를 들 수 있는데, SACON의 지식베이스는 유한요소의 구조 해석 프로그램인 MARC를 사용하는 데 관한 지식이 포함되었다.

표 2.1 Expert System shells

Name	Knowledge representation, inference, and control	Implementation
EMYCIN ( MYCIN )	O-A-V triplets If-then rules Backward chaining Certainty factors	INTERLISP DEC PDP-10
EXPERT ( CASNET )	A-V pairs If-then rules Forward chaining Confidence factors	FORTRAN Various computer
KAS (PROSPECTOR)	O-A-V triplets If-then rules Forward and Backward chaining Certainty factors	INTERLISP Xerox 1100

Expert system shells는 먼저 전문가 시스템을 구축하기 위한 도구로서 개발하기 위한 것이 아니라 적용분야에 적합한 전문가 시스템을 개발한 후 이 전문가 시스템에서 사용하는 기능을 바탕으로하여 다른 분야에서도 적용 가능한 전문가 시스템의 도구로 만들어진 것이다. 그러므로 이런 형태의 도구는 먼저 개발된 전문가 시스템의 적용분야와 유사한 경우 빠른 시간에 전문가 시스템의 개발이 용이하다. 그러나 유사하지 못

한 분야에서는 제어 기관의 고정, 지식 표현의 제약, 설명기능의 제한으로 인한 시스템 개발의 비효율성이 단점으로 되어 있다. 여기에 해당하는 도구로는 표 2.1에 나타난 바와 같다.

### High-Level Programming Language

지난 수 년동안에 지식을 표현하기 위하여 규칙을 사용하는 수 많은 도구들이 상품화되어 가격이나 능력, 그리고 Hardware의 능력에 따라 구입하여 사용하는 것이 가능하다. 대형이고 복잡한 시스템은 Mainframe 이거나 Minicomputer, 또는 LISP머신 등을 필요로 하며, 가격은 보통 1000달러에서 6만달러까지의 범위이다. 작은 규모의 셀은 보통 100 달러부터 수 백달러의 가격으로 구입이 가능한 데, 이는 보통 개인용 컴퓨터에서 사용이 가능하다. 이러한 Software는 개발하는 회사가 새로운 기법이나 사용자의 요구에 따라 계속 변경, 확장시켜 나가므로 급속히 그 능력이 증대된다. 다음의 표 2.2은 개인용 컴퓨터에서 사용이 가능한 도구들을 간추려 요약한 것이다.

이러한 도구들이 가지고 있는 장점은 다음과 같다. 대부분이 가격이 저렴하며 보급되어 있는 대부분의 소형 컴퓨터에서 사용이 가능하다. 또한 컴퓨터에 대한 지식이 거의 없는 사용자도 조금의 시간만 투자하더라도 쉽게 원하는 도구를 이용할 수 있다. 이러한 도구를 사용하면 빠른 시간에 전문가 시스템의 구축을 위한 시험(Prototype)을 해 볼 수 있으므로 노력을 절약할 수 있다. 이러한 장점들로 인하여 상용적으로 보급된 많은 도구들이 대학, 회사 및 정부기관에서 이용되고 있는 데, 이러한 도구를 이용할 경우에는 다음의 각 항의 설명과 같은 단점을 염두에 두어야 한다.

## ⌘ 2.2 High-level Programming Language

Name	Knowledge representation, inference, and control	Implementation
OPS5	O-A-V triplets If-then rules Recognize-act cycle with conflict resolution	FranzLISP, MacLISP VAX 11/780
M.1	A-V pairs If-then rules Certainty facts Backward chaining Depth-first search	PROLOG IBM PC
INSIGHT2+	A-V pairs If-then rules Forward and Backward chaining Certainty factors	PASCAL IBM PC, DEC Rainbow, Victor9000
S.1	O-A-V triplets If-then rules Multiple objects with inheritance Backward chaining Depth first search Certainty factors	LISP DEC-VAX, Xerox 1100 Symbolics 3600
EXSYS	O-A-V triplets If-then rules Forward and Backward chaining certainty factors	C Various computers

1. 제어 기관의 고정 - 제어 과정은 대부분이 전진 또는 후진 추론으로 제한되어 있다. 어떠한 규칙을 적용할까 또는 언제 추론과정을 그만둘까 하는 문제는 사용자의 입장에서서는 조절이 불가능하다.
2. 지식 표현의 제약 - 추론의 지식을 규칙으로 표현하게 되는데, 이는 어색할 때도 있으며 한편으로는 규칙의 구성을 지나치게 복잡하게 만들 우려도 있다. 즉 때로는 간단한 생각을 상당히 많은 규칙으로 나타내야 할 경우도 있다.
3. 설명 능력의 제한 - 보통 설명능력은 일반적이지 못하고 산문적으로 표시한 출력에 의한 방법으로 해결해야 한다.
4. 시스템 개발의 비효율성 - 이 도구는 전형적으로 개발단계에서와 전문가 시스템의 이용단계에서는 다른 Software를 쓸 경우가 대부분인데 이는 시스템의 개발하는 입장에서는 불편하다. 예를들면 Editing이나 Debugging은 실행단계에서도 지식베이스에 접근이 가능하도록 해 두면 사용하기가 편리할 것이다.

결론적으로 전문가 시스템의 구축을 하기위한 목적으로 도구를 사용하는 경우에는 원하는 전문가 시스템의 원형을 시험해 본 다던지 또는 규모가 크지않은 소형의 전문가 시스템을 구축하고자 하는 경우이다. High-level programming languages는 비록 shells보다는 사용자가 처리해야 할 일이 많이 있지만 이런 형태의 도구는 적용 범위가 상당히 광범위하다. 또한 특별한 분야의 시스템을 구축할 때 그 분야에 필요한 기능을 추가 및 보완할 수 있는 유연성(flexibility)을 가지고 있다. 특히 추론 기관의 조절, conflict resolution, 컴퓨터 그래픽과 설명기능을 추가할 수 있다.

### Mixed Programming Environments

위에서 언급한 도구들이 제한적인 이유로 사용하기가 곤란할 경우에는 보다 고급의 언어로서 LISP와 같은 언어를 이용할 수 있다. 보통 Programming Environment 라 불리는 프로그래밍 tool은 시스템에서 사용가능한 명령어, 에디터, 사용자와의 인터페이스, 복합적인 지식의 표현등의 전문가 시스템을 개발하는 데 일을 편하게 해주는 여러 기능등을 가진다.

이러한 기능으로서 대표적인 것으로는 디버깅, 입, 출력지원, 설명기능, 지식베이스의 에디터가 있다. 디버깅을 지원해주는 것으로는 추적기능을 들 수 있는데, 이 기능은 실행단계에서 사용되었던 규칙등의 이름을 나열하는 방식으로 사용자에게 정보를 제공하게 된다. 또다른 기능으로는 미리 프로그램중에 실행을 중단시킬 수 있는 기능(Break)을 두어 그 당시의 지식베이스의 상태를 조사해 볼 수 있게 한 것도 있다. 입,출력지원기능으로서는 프로그램의 실행과정에 지식을 획득하게 한 것을 보면 EMYCIN에서 지식베이스에서 해당지식을 발견할 수 없으면 필요한 정보를 사용자에게 요구할 것을 예로 들 수 있다. 또한 컴퓨터 그래픽스를 이용하여 윈도우/메뉴 방식을 제공하여 입력시 요구되는 선택항목에 대해 오류를 피하고 빠른 시간안에 입력을 할 수 있게 하였고, 그래픽으로 정보의 이해에 효율을 기하고 있다. 대부분의 전문가 시스템은 특정한 결론에 도달하게 된 경위에 대하여 사용자에게 설명을 할 수 있는 기능을 제공하고 있는데, 어떠한 것은 이러한 기능이 내장되어 있지 않아 전문가 시스템을 구축하는 자가 이를 고려하여 프로그램을 하여야 한다. 위에서 언급한 세가지의 기능외에 추가되는 것으로는 지식베이스를 편집하기 위한 에디터가 있다. 이 기능은 지식베이스에 있는 지식들의 문장을 체크하는 것을 들 수 있는데, 이는 사용자가 문법에 어긋난 규칙이나 명령문에 대해 에디터가 이를 발견하여 틀린 사항에 대해 설명을 해 주는 것을 말한다.

이런 프로그램 환경은 인공지능, 컴퓨터 그래픽스, 객체 지향적 프로그램등의 전산 과학의 여러 분야의 접근을 포함하므로 Hybrid 도구라고 불리워진다. 프로그램 환경은 전문가 시스템내의 지식을 표현하기 위하여 프레임의 데이터구조로 정보를 구성하는데, 이 프레임은 정보를 저장하고 불러내는 데이터베이스의 역할을 한다. 프레임이라는 것은 특정한 개념이나 물체를 나타내는 것으로서 각각의 프레임은 고유의 이름을 가지고, 수치적인 값이나 함수를 값으로 하는 슬롯 또는 Facet 이라하는 속성을 가진다.

프로그래밍 환경과 전문가 시스템용 도구와의 차이는 다음의 사항으로서 구별할 수 있다.

1. 지식표현의 유연성 - 프로그램 환경은 규칙, 프레임등의 복합적 지식표현이 가능하다.
2. 논리적 추론과정에 있어 추론과정의 조절 - 사용자가 추론의 방법을 제어할 수 있다.
3. Conflict Resolution 을 조절이 가능하다 - 현재의 상태에 적용가능한 규칙이 여러개 있을 때, 그 규칙들이 어떻게 관리될 것인가를 조절할 수 있다.
4. 사용자의 편리한 이해를 위하여 컴퓨터 그래픽스나 설명 기능을 추가할 수 있다.

수 많은 도구의 종류와는 대조적으로 프로그래밍 환경은 상당히 비싸고, 특별한 Hardware를 요구한다. KEE, ART 그리고 Knowledge Craft 등과 같은 큰 규모의 복합형 도구는 LISP Workstation이나 SUN 3등의 Supermicro에서 가능하며, 가격은 6만 달러정도이다. 물론 이러한 상황은 계속 변하여 이러한 기능을 가진 도구가 IBM

PC-AT 등에서도 실행이 가능하게 하려는 노력은 계속되고 있다. 게다가 이러한 시스템은 그 사용법을 익히는 데 일정한 교육이나 상당한 기간의 실습을 거쳐야 한다. 다음 표 2.3는 프로그래밍 환경에 속하는 사항들을 비교, 설명하고 있다.

### Rule Induction Tools

앞의 두 절에서 언급한 전문가 시스템용 도구와 프로그래밍 환경외에 예제를 기초로 하여 규칙을 만들어 내는 즉, 지식의 획득을 돕는 Rule Induction Tool 이 있다. Microcomputer 에서 사용가능한 상용 소프트웨어는 가격면에서 보통 200 내지 1200 달러 정도이며, 대형 시스템인 TIMM인 경우에는 PC, Mini computer, Mainframe에서 사용가능하지만 1000에서 50000 달러 정도까지 한다. 작은 시스템으로는 널리 보급된 EXPERT EASE가 있다. 현재 자료조사된 도구를 소개하면 표 2.4와 같다.

### Æ 2.3 Mixed Programming Environment

Name	Knowledge representation, inference, and control	Implementation
KEE	Frames with slots Object-oriented programming If-then rules Multiple objects and extensive inheritance Forward and Backward chaining	Zeta/CommonLISP TI Explorer Xerox 1100 Symbolics 3600 LMI Lambda
LOOPS	Frames with slots User-definable inheritance Lattice of objects Forward and Backward chaining	INTERLISP Xerox 1100
ART	O-A-V triplets Logical links between objects, Multiple objects and inheritance Forward and Backward chaining Certainty facts	ZetaLISP, VaxLISP LMI Lambda DEC-VAX TI Explorer
Knowledge Craft	OPS-5 rule with frames User-definable inheritance Object-oriented programming Forward and Backward chaining Agenda	Common LISP DEC-VAX TI Explorer Symbolics 3600

### Æ 2.4 Rule Induction Tools

Name	Organization	Implementation
Expert/Ease	Intelligent Terminals	PASCAL IBM PC, DEC Rainbow, Victor9000
TIMM	General Research Corporation	FORTRAN IBM, DEC, Prime
Rulemaster	Radian	C VAX, SUN workstation
ACLS	Southdata Software	PASCAL IBM PC, APPLE II
ETS	Boeing Computer Service	INTERLISP-D Xerox 1100

## 2.2 공학문제에 적합한 전문가 시스템용 도구의 조건

전문가 시스템을 구축하고자 할때 가장 먼저 고려할 사항은 적용하고자 하는 분야에 대한 문제를 명확하게 이해하여야 한다. 그 다음 주어진 문제에 가장 적합한 전문가 시스템 도구를 선정하거나 개발해야 한다. 그리고 나서 획득된 지식을 전문가 시스템 도구가 제공하는 언어로 표현해야 한다. 기존에 있는 도구를 선정할 것인지, 아니면 새로운 도구를 만들 것인지는 각각 장단점을 지니고 있다. 도구를 선정하면 전문가 시스템의 구축의 시간과 경비를 줄일 수 있으나 도구에서 제공하는 기능이 한정되어 있어 적용분야에 대해 적절히 사용할 수 없다. 그러나 도구를 개발하면 개발자의 의도에 의해 적용 분야에 가장 알맞는 도구를 개발할 수 있으나 많은 개발시간이 요구된다.

공학분야의 전문가 시스템을 구축하기 위한 도구는 공학이 다루어야 할 분야가 광범위하므로 유연성이 좋아야 한다. 다음은 이상적인 공학용 도구가 지니야할 사항을 언급한다.

(1) 도구(tool) 내부에서 수치적 계산이 가능해야 하고, 도구에서 연산기능을 제공해야 한다. 만약 도구내에서 제공할 수 없을 경우 외부함수(external function)라도 사용할 수 있어야 한다.

(2) 여러가지 지식표현 형태를 지니고 있어야 한다. 적용 분야에 알맞는 지식표현 형태를 선정할 수 있는 함으로써 규칙의 갯수, 지식베이스의 크기, 그리고 지식베이스의 관리가 용이하다.

(3) 여러가지 추론형태를 가지고 있어, 문제영역에 따라 추론형태를 선택하여 사용

할 수 있어야 한다. 예를 들어 진단(diagnosis)이나 조사(interpretation)에서는 후진추론이 타당하고 설계(design)나 계획(planning)은 전진추론이 적당하다.

(4) 다른 program이나 다른 programming언어로 쓰여진 전문가 시스템과의 연결이 가능해야 한다. 이 조건은 구조설계문제에서는 가장 절실한 문제로서 기존에 작성된 프로그램과 복합된 시스템을 구축할 수 있는 도구이어야 한다.

(5) 설명기능을 지니고 있어야 한다. 추론한 결과의 신뢰도를 제공하기 위해서는 설명기능이 있어야 한다.

(6) 여러가지 활용할 수 있는 환경을 지니고 있어야 한다. 에러를 찾아주는 디버거(debugger)와 에디터(editor)등 여러가지 기능을 갖추고 있으면 편리하다.

(7) 모든 지식은 불확실성을 가지고 있으므로 uncertainty에 대해 처리할 수 있는 기능이 있어야 한다.

(8) 작업을 각 단계별로 분리하여 문제를 해결해야 할 방법이 필요하다. 즉, A라는 작업을 시행할 때는 B라는 작업에 사용되는 지식이 필요하지 않는다면 작업을 분리함으로써 지식 베이스의 탐색공간이 상당히 줄어들어 추론시간을 감소시킬 수 있다.

(9) 사용자의 편의를 위해 window/menu양식을 제공하고 그래픽 라이브러리를 구축하여 primitives를 제공하여 언제든지 사용할 수 있어야 한다.

### 3. K-CLIPS의 개발

#### 3.1 개요

CLIPS 는 NASA/Johnson Space Center의 Artificial Intelligence Section 에서 만든 전진추론 (forward chaining)을 지닌 Production System이다. 이 시스템의 개발 목적은 범용 컴퓨터에 서의 LISP의 떨어지는 기능을 보완하고 현재 LISP으로 개발된 도구와 하드웨어의 높은 가격에 비해 보다 싼 가격으로 전문가 시스템용 도구를 개발하고, 또한 타 언어와의 연계성을 강화하기 위한 것이다. CLIPS의 특성을 요약하면 표 3.1과 같으며 기타 자세한 사항은 부록 1의 K-CLIPS 사용법을 참조하기 바란다.

표 3.1 CLIPS의 특성

Knowledge Base	Facts	A-V
	Relation	rule
		variable
Uncertainty	수행 못함	
Inference	Recognize-Act Cycle	
	Forward Chaining	
	Depth First Search	
	Breadth First Search	
	Meta-Rule, Agenda	
User-Interface	Edit --- Micro EMACS editor	
	Trace	
	Explanation	
	Windows	
	Other language interface	

K-CLIPS는 기존의 CLIPS의 기능을 분석, 검토하여 단점을 보완하고, 인공지능의 전문가 시스템 구축에 대한 기술의 축적과 계속적인 연구로 독자적인 도구를 개발하려는 목적으로 현재 KAIST 토목공학과에서 개발한 도구이다.

기존의 CLIPS의 기능을 보완한 사항은 다음과 같다.

### (1) 지식 표현기법의 확장

현재의 CLIPS에서는 지식을 Attribute-Value 형태의 심볼리스트(symbol list)로 표현하고 있다. 그러나 대부분의 공학적인 문제는 지식들이 계층적인 구조를 지니고 있는데, 일반적인 패턴인식의 해결을 위해 이런 계층 구조는 모두 규칙(rule)으로 표현해야 하므로 많은 규칙이 요구된다. 그러므로 계층적인 구조를 표현하기에 용이한 프레임 형식의 지식표현 기법이 추가되었다. 그렇게 함으로써 규칙의 수를 줄이고 지식을 효율적으로 관리할 수 있다.

### (2) 함수를 정의하는 기능개발

전문가 시스템에 사용된 모든 규칙(rules)과 사실(facts)은 도구가 제공하는 명령어로 쓰여 진다. 이런 기존의 명령어로 이루어지는 함수를 사용자가 원하듯대로 정의할 수 있게 함으로서 동일한 작업의 반복 및 고차원의 명령어를 형성하여 사용할 수 있다. 그렇게 함으로 K-CLIPS가 제공하는 명령어의 수를 증가시키고 지식베이스의 구축을 보다 체계적으로 할 수 있다.

### (3) 사용자와의 인터페이스 기능의 확장

전문가시스템의 장점은 인간의 추론능력을 모방할수 있는 기능이 뛰어난 특징을 가지고 있는데, 이러한 장점은 사용자가 컴퓨터의 추론과정에 대한 대한 정보를 제공할 수 있고 제공받을 수 있게 하는 인터페이스 기능을 요구하게 된다. K-CLIPS에서는 인터페이스 기능의 원활화를 위해 사용자에게 정보를 제공할 수 있게 하는 윈도우기능과 사용자로부터 원하는 입력자료를 얻고자 할 경우 사용할 수 있는 메뉴형식의 입력방식을 확장하였다.

#### (4) 그래픽 라이브러리 함수의 추가

사용자와의 인터페이스로 가장 중요한 요소는 지금의 데이터에 대한 정보를 그림으로 제시하는 것이 최고이다. K-CLIPS에서는 규칙상에서 직접 그래픽을 할 수 있도록 점, 선, 면, 문자등의 도형에 관한 함수를 내장시켰다.

### 3.2 K-CLIPS의 지식표현 구조

K-CLIPS의 기본적인 구조는 주된 지식베이스와 사용자로 하여금 지식베이스를 조절하도록 해주는 일련의 보조모듈로 그림 3.1과 같이 되어 있다.

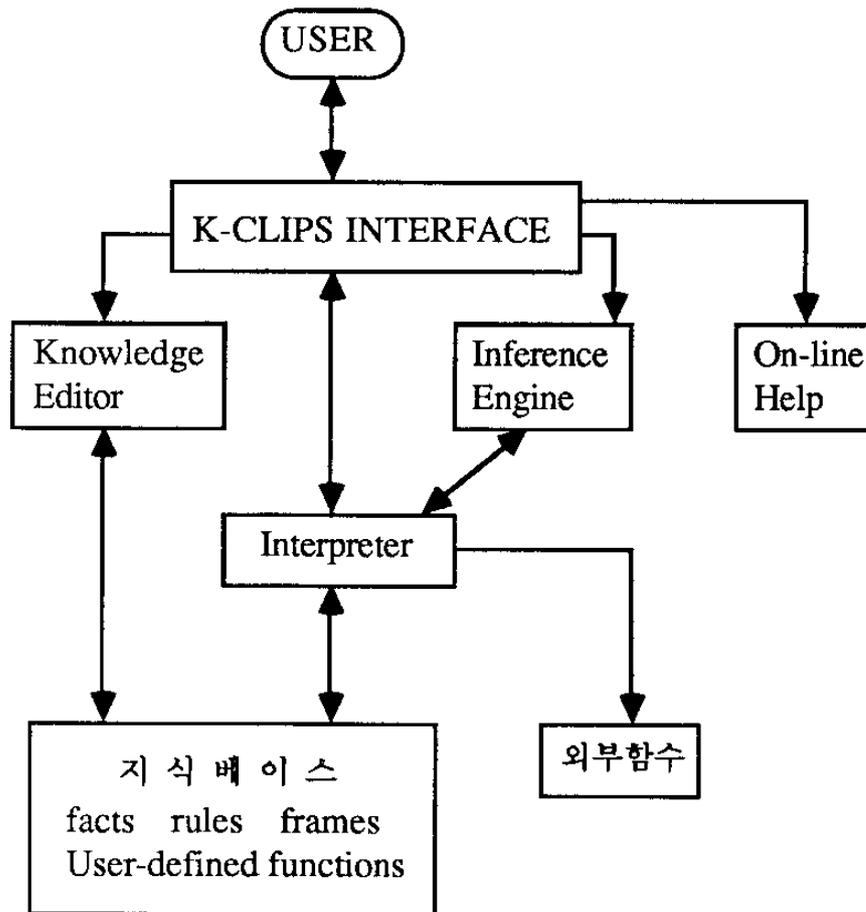


그림 3.1 K-CLIPS 의 구조

지식베이스에 구축될 지식은 4개의 지식표현 구조로 표현될 수 있다. 이는 다음과 같다.

- o Facts
- o Rules
- o Frames
- o K-CLIPS 구문으로 이루어진 User-defined-functions

그러므로 지식기반 전문가시스템을 개발하는 과정은 지식베이스내에 4가지 형태로 획득된 지식을 표현하는 것이라고 할 수 있다. 사용자는 사실(facts) 및 프레임(frames)으로 static한 지식을 표현하고 이를 조절하기 위한 규칙 및 함수를 설정함으로써 특정분야의 문제해결을 지적으로 해결할 수 있는 프로그램을 작성할 수 있다. 이들의 각각에 대한 세부적인 설명은 부록 1을 참조하기 바라며 3.2.1 - 3.2.4 절에서 간략히 그 개념을 설명한다. 그림 3.1에서 지능형 편집기(knowledge editor)는 사용자로 하여금 지식베이스에 있는 지식을 불러내어 수정, 추가, 삭제 및 재배열 등을 추론과정의 어떤 단계에서 가능하게 해준다.

### 3.2.1 사실(Facts)

추론과정에서 static한 지식의 저장은 working memory에 보관하게 되는데, 이는 K-CLIPS에 사용되는 사실들의 집합으로 이루어지며 이러한 사실들에 대하여 규칙이 적용된다. 이러한 메모리에 하나의 사실을 추가하는 명령은 assert이고 제거하는 명령은 retract이다. 그리고 여러개의 사실을 한번에 추가하는 명령에는 deffacts가 있다. 여기서 정의되는 사실들은 리스트(list) 형식으로 표현된다.

### 3.2.2 규칙(Rules)

지식베이스에는 K-CLIPS에 정의되는 규칙들이 정의된다. 규칙은 다음과 같은 형태로 정의된다.

### 3.2.3 Frames

K-CLIPS에서는 fact-lists라는 데이터 베이스이외에 프레임 형태의 지식표현이 가능하도록 설계 구현하였다. 프레임 형태의 지식표현 수단은 Minsky가 처음으로 제한한 것으로 프레임 구조는 프레임이라 불리는 노드 (node) 와 이들 프레임 들간의 상속관계를 나타내는 계층구조로 된다. 가장 윗부분의 노드는 일반적인 개념을 표현하며 아래로 내려 갈수록 보다 세부적인 경우를 표현한다. 프레임 구조에서 각 노드가 나타내는 개념은 속성 (attribute) 과 속성값(values of attribute) 에 의해 정의되며 보통 이 속성들은 슬롯 (slot) 으로 표현된다. 각 슬롯에서는 슬롯의 값이 변할 때 자동적으로 실행을 할수 있는 'demon'이라는 프로시저 (proccdure) 를 정의할 수 있다. 그러나 K-CLIPS에서는 프로시저 기능을 수행하지 않고 단지 계층적인 상속만 가능하도록 구현되었다.

#### 3.2.3.1 클래스의 정의

K-CLIPS에서는 노드에 해당하는 역할을 하는 것을 클래스(class)라 부른다. 클래스는 define class 함수에 의해 정의되며, 클래스를 정의하면 실제 노드가 만들어 지고 K-CLIPS의 지식 베이스 속에 등록된다. 또한 define class는 상위클래스(superclass)와 하위클래스(subclass)의 관계를 결정하여 준다. 클래스 정의의 예를 들면 다음과 같다.

```
(define class frame (beam column))
```

여기서 하위 클래스 beam은 상위 클래스 frame의 슬롯과 슬롯값을 상속 받는다. 클래스가 정의된 지식 베이스는 그림 3.2와 같이 linked-list 형태로 내부에 구성되어 있다. 그리고 보다 빠르게 클래스를 찾기위해, hashtable이라는 array에 정의한 클래스의 번지 (address)인 포인터(pointer)를 기입하고 있다. 그렇게 함으로써 이미 정의된 클래스는 직접 그 클래스를 찾을 수 있다. hashtable의 구조는 그림 3.3과 같이 표현된다.

### 3.2.3.2 클래스의 슬롯

슬롯의 정의 기능을 사용하여 클래스의 슬롯을 만들 수 있다. 클래스가 하나의 객체 (object)를 나타낸다면 슬롯은 속성을 나타내고 또한 여러개의 속성값을 가질 수 있다. K-CLIPS에서의 클래스의 슬롯은 지식 베이스에 저장, 제거 될 수 있고, 또한 슬롯의 속성값을 변화시킬 수 있다. 이러한 작업을 가능하게 하는 함수가 create\_slot, remove slot, add slot, modify 이 있다. 형식은 다음과 같다.

(1) (create\_slot <class> ^<slot 1> <value1> ^<slot 2> <value2> )

<class>의 슬롯인 <slot1>, <slot2>의 값 <value1> <value2>을 지식 베이스에 기입한다.

(2) (remove\_slot <class> ^<slot> &optional <value>)

<class>의 슬롯인 <slot>의 값을 제거한다. 만약 <value>을 쓰지 않으면 <slot>의 모든 값이 제거 된다.

(3) (add\_slot <class> ^<slot> <value> )

<class>의 슬롯이 여러 개의 값을 가질 경우 이 함수를 사용한다.

(4) (modify <class> ^<slot> <value1> => <value2> )

<class>의 슬롯의 값을 <value1>에서 <value2>로 변화시킨다.

### 3.2.3.3 클래스 패턴문장

클래스와 슬롯들이 지식 베이스에 정의되면 이것들을 규칙의 조건부에 표현하고 매치(match) 해야 할 방법이 필요하다. 이때 클래스, 슬롯으로 구성된 지식 베이스는 보통의 지식 베이스와는 다르게 계층구조를 가진다. 이러한 계층 구조에는 클래스와 클래스, 클래스와 슬롯의 관계 등이 있을 수 있다. 이런 관계를 표현하여 규칙의 패턴으로 사용되는 문장을 클래스의 패턴문장이라고 한다. K-CLIPS에서는 다음과 같은 2개의 문장으로 정의하여 사용할 수 있게 하였다. 여기서 <class2> <value>에 패턴 변수가

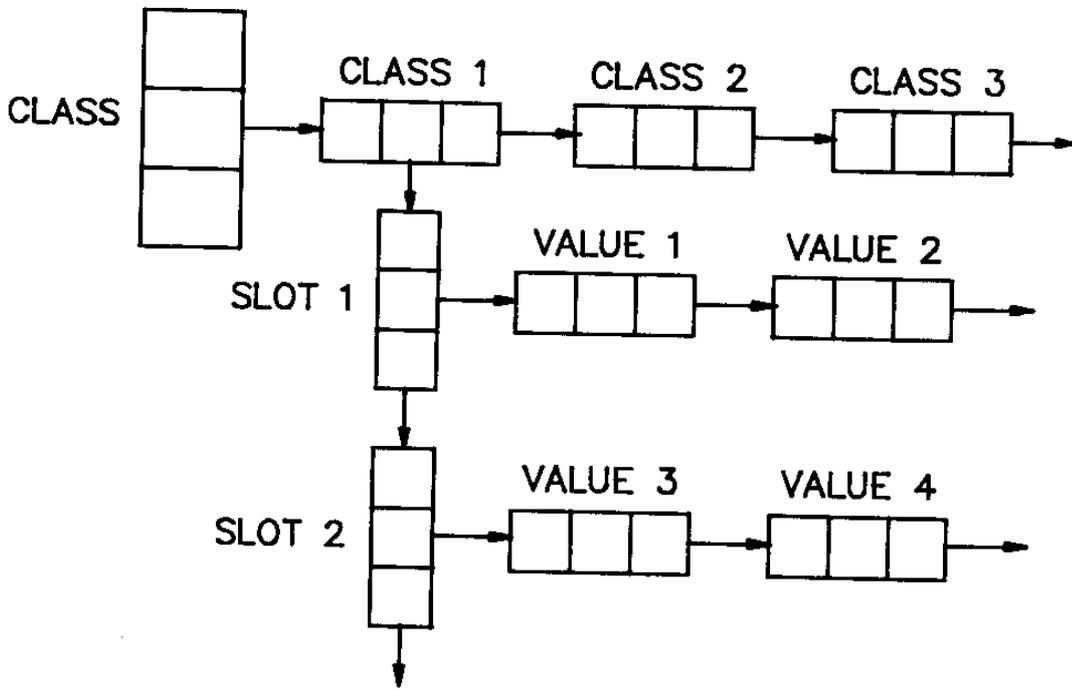


그림 3.2 지식베이스의 Linked-List

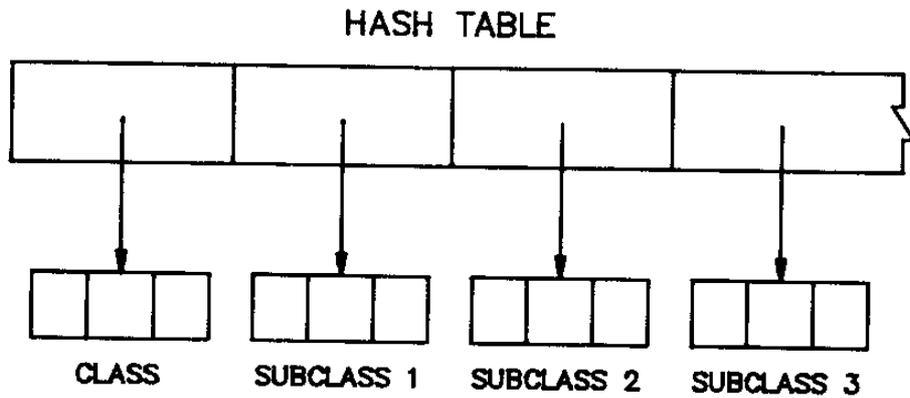


그림 3.3 HASHTABLE 의 형태

들어갈 수 있다.

(1) (SUBCLASS\_OF <class1> <class2> )

<class1> 이 <class2>를 하위 클래스로 가지는 것을 서술한다.

(2) (SLOT\_VALUE <class> <slot\_name> <value> )

<class>의 한 <slot\_name>의 속성값이 <value>임을 서술한다.

#### 3.2.3.4 클래스의 Predicate

요구하는 조건에 만족하는 클래스와 슬롯 또는 슬롯값이 지식 베이스에 존재하는 하  
는가를 체크하여 TRUE 와 FALSE 의 값으로 되돌려 주는 것을 클래스의 predicate 이라  
고 한다.

K-CLIPS 에서 사용할 수 있는 것으로는 다음과 같은 3 개의 문장을 사용할 수 있다.

(1) ( slot\_of <class> <slot\_name> )

<slot\_name>이 <class>의 슬롯 인 것을 서술한다.

(2) ( subclass\_of <class1> <class2> )

<class1>이 <class2>를 하위 클래스로 가지는 것을 서술한다.

3) ( slot\_value <class> <slot\_name> )

<class>의 슬롯 <slot\_name> 의 속성값이 존재하면 return 이 속성값이고 그렇지 않으면  
FALSE 가 된다.

위의 기능들을 설명하기 위해 다음과 같은 간단한 예를 들어 보았다.

```
KCLIPS> (defrule Move-Block "rule1"
```

```
  (SLOT_VALUE Monkey ^at ?x1 ^on floor ^energy ?h1)
```

```
  (test (>= ?h1 3))
```

```
(defrule <rule-name> "this is example"
  (declare (salience 10))
  <pattern> ....
  =>
  <action> ....
)
```

defrule은 규칙을 정의하는 명령이고 <rule-name>은 규칙의 이름을 나타낸다. ""은 코멘트를 표시한다. =>의 앞부분은 LHS(left hand side)라 하며 뒷부분을 RHS(right hand side)라고 부른다. 규칙의 LHS에 있는 <pattern>들은 임의의 리스트 형태가 될수 있고 패턴변수(pattern variable)를 포함할 수 있다. 그리고 데이터베이스에 <pattern>과 매치되는 존재하면 규칙의 RHS에 위치한 <action>이 수행되는 데, 여기에서 사용될 수 있는 작업은 새로운 사실들의 추가, 기존 사실의 삭제, 결과의 출력, 입력자료의 요구와 함수들의 요구등이 있다. 예를 들어보면 다음과 같다.

```
(defrule determine-engine-state "rule1"
  "rem <- (query phase)
  (not (working-engine ?))
  =>
  (retract ?rem)
  (printout "What is the working state of the engine : " crlf)
  (bind ?response (read))
  (assert (working-state engine ?response))
)
```

determine-engine-state는 규칙의 이름이며 ?rem은 사실 (query phase)의 기억장소를 나타내는 패턴변수(element variable)이며 ?response는 단순한 패턴변수이다. 규칙의 수행도중 매치되는 지식베이스 원소에 해당되는 패턴변수에는 원소의 값이 바인딩(binding)된다. RHS의 action들은 하나씩 수행되며 위의 retract는 데이터베이스 (query phase)라는 사실을 제거시키고 assert는 새로운 사실을 추가한다.

```

(SLOT-VALUE Block ^at ?x2)
(SLOT-VALUE Banana ^height high ^at ?y1)
(test (neq ?y1 ?x1))
=>
(bind ?value (- ?h1 3))
(create-slot Monkey ^energy ?value ^at ?y1)
(create-slot Block ^at ?y1)
(fprintout t "Monkey move the block from " ?x1
           " above which a banana is hanging" crlf
           "Current energy =" ?value crlf))

```

이 규칙은 Monkey 와 Banana 문제에서 Monkey 가 상자를 바나나가 매달린 장소에 운반하는 규칙이다.

### 3.2.4 User Defined Functions

K-CLIPS는 사용자가 내장하고 있는 기본함수 (basic function)를 바탕으로 보다 복잡하고 고차원적인 함수를 구축할 수 있다. 한번 정의된 함수는 제거하지 않는 이상, 내장된 함수와 동일하게 사용될 수 있다. 즉 이 함수는 규칙의 RHS에서 사용이 가능하며 이전에 정의된 함수를 사용하여 새로운 함수를 구축할 수 있다.

예를 들어 함수 square 는 변수 x 의 제곱으로 정의한다.

```
( defun square (x) (* x x) )
```

그 다음 함수 square 는 언제든지 사용할 수 있다.

```
( square 2.5 )
```

변수 x 에는 2.5 이 대응되어 리턴되는 값은 6.25 이다.

K-CLIPS에서 한번 정의한 함수는 다른 사용자 정의함수에 사용될 수 있다. 예를들어

```
(defun sum_of_squares (x y)
  (+ (square x) (square y)))
```

이 함수는 x,y 두 변수를 제공하여 합하는 함수이다.

```
(sum_of_squares 2 4)
```

계산된 결과의 값은 20.0으로 평가된다.

또한 이 함수는 규칙과 함께 사용된다. 예를들면,

```
(defun my-assert (x y)
  (str-assert (str-cat x "+" y)))
```

```
(defrule my-assert-rule
  (data ?x ?y)
  =>
  (my-assert ?x ?y))
```

그러면 사실이 (data user define)이면 fact-list에는 (user + define)인 사실이 하나 assert 된다. 또 다른 예로는 앞서 정의한 sum\_of\_squares 함수를 사용한 규칙을 보기로 하자.

```
(defrule Sum-of-Squares-Rule
  (data ?x ?y)
  =>
  (bind ?eval (sum_of_squares ?x ?y))
  (fprintout t "*** Sum of Squares " ?x " , " ?y " = " ?eval crlf))
```

이 규칙이 수행되기 위해서 사실 ( data 3.0 4.0 )을 assert 하고 (run) 을 하면

```
** Sum of Squares 3.0 , 4.0 = 25.0
```

의 리턴값을 가진다.

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.

defun 함수 이외에, 사용자가 정의한 함수명의 리스트를 출력하거나 선택된 함수의 포맷을 출력하거나 또는 정의한 함수가 더이상 필요하지 않을 경우에 제거할 수 있는 함수를 K-CLIPS에는 제공되고 있다. 아래에 언급하는 명령어로 사용한다.

(displayfunc) : 정의된 함수의 이름들을 프린트 한다.

(remove\_defun <function-name>) : <function-name> 에 입력된 함수를 함수의 리스트에서 제거한다.

(ppfunc <function-name>) : 정의한 <function-name> 의 내용을 프린트 한다.

### 3.3 확장된 인터페이스 함수

전문가시스템에서 사용자와 직접적으로 연결되는 인터페이스 부분은 시스템 개발자가 신중히 고려해야 할 사항중 하나이다. 그러나 기존의 전문가시스템도구에서는 단지 텍스트 타입의 입출력기능만 제공하여 전문가시스템 사용자와의 정보의 전달에 있어 매우 불편한 점을 느끼게 되었다. 이런 단점을 보완하기 위해 많은 전문가시스템도구에서는 원활한 인터페이스를 위해 윈도우 및 메뉴 형식의 툴박스를 제공하기 시작하고 있다. 그러나 현재까지는 전문가시스템도구에서 그래픽기능을 제공하지 못하는 실정이다. K-CLIPS에서는 전문가시스템 사용자의 이해의 폭을 증가시키기 위해 시스템 구축에 활용할 수 있는 윈도우 및 메뉴형식의 구축함수와 그래픽처리가 가능한 함수를 제공해 주고 있다. 메뉴함수는 전문가시스템 개발자의 의도에 맞는 정보를 얻고자 할 경우나 또는 사용자가 편리하게 정보를 시스템에 제공하고자 할 경우에 사용하는 입력함수이다. 윈도우함수는 전문가시스템의 사용자의 사용자에게 추론된 결과나 정보를 원하는 크기의 윈도우에 텍스트 형식으로 제공하는 출력함수이다. 그리고 마지막으로 그래픽함수는 출력을 제공하는 함수로서 추론된 결과는 그림으로 보여주거나 사용자의 이해를 돕기위해 그림으로 설명해 준다. 위에서 언급한 인터페이스 함수는 현재 PC상에서 사용하도록 개발되었다.

#### 3.3.1 비디오 모드

본 연구에서 개발된 윈도우, 메뉴, 그래픽 함수의 설명에 앞서 PC상의 비디오모드에 대한 사항을 먼저 알아야 한다.

사용자가 이용하는 PC의 Video Adapter의 종류를 먼저 파악해야 한다. Monochrome Display Adapter (MDA)는 단지 텍스트만의 출력이 가능하며 그래픽 출력이 가능한 Adapter로는 Color Graphics Adapter (CGA), Enhanced Graphics Adapter (EGA), Hercules Monochrome Graphics Adapter가 있다. 본 연구에서 사용한 Adapter로는 Enhanced Graphics Adapter (EGA)를 이용하였다.

각 Adapter에 따라 화면의 크기, 화면의 resolution, display type(컬라 또는 흑백)이 달라지게 된다. PC상의 비디오 모드는 두가지로 나누는 데

1) 텍스트 모드(text mode) : 텍스트 모드의 PC화면은 cell로 이루어져 있는데 각 cell은 하나의 attribute와 character를 가진다. attribute는 문자의 색깔과 밝기를 결정하는 것이고 character는 ASCII문자를 말한다. 그림 3.4 처럼 텍스트모드의 화면의 상부좌측의 좌표는 (1,1)이고 x축은 오른쪽으로 갈수록 + 이고, y축은 아래로 갈수록 + 이다. 현재 연구에서 사용되는 텍스트 모드의 화면의 크기는 80 column과 25 line을 가진다.

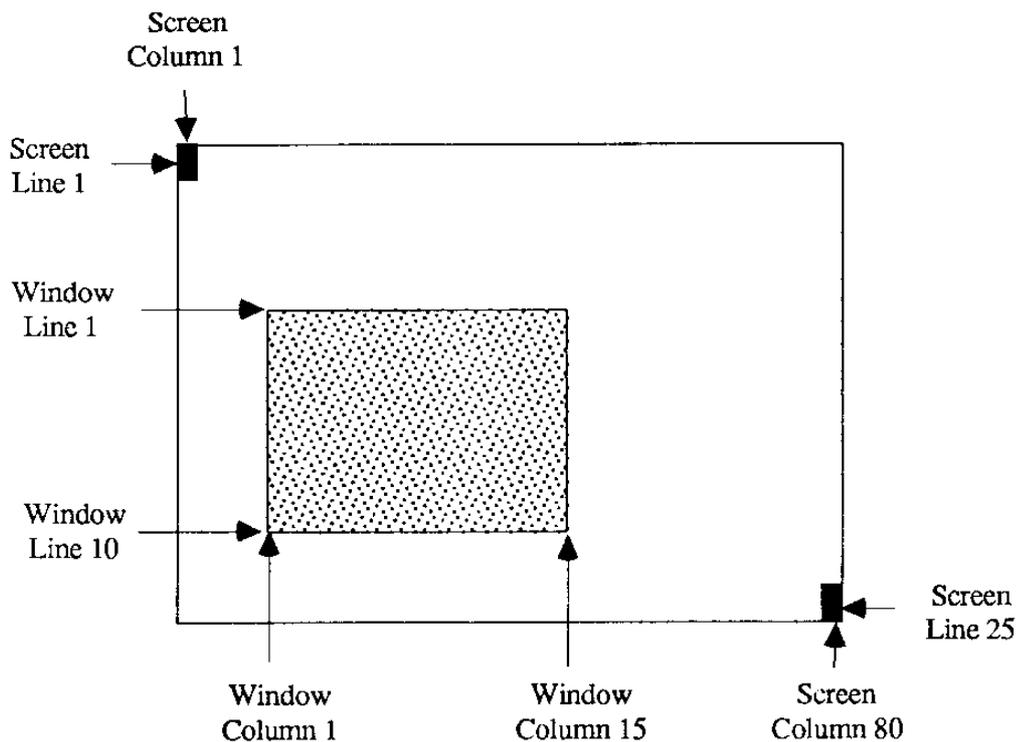


그림 3.4 Window in 80 \* 25 Text Mode

2) 그래픽 모드(graphics mode) : 그래픽 모드의 PC화면은 pixel로 구성되는데 각 pixel는 화면상에 하나의 점으로 나타난다. pixel의 수는 시스템에 연결된 비디오 어댑터(video adaptor)의 종류에 따라 달라진다. 그림 3.5 처럼 화면의 상부좌측의 좌표는 (0,0)이고 x,y의 좌표값은 텍스트모드에 사용된 것과 동일하다. 현재 연구에서 사용된 pixel의 수는 640\*350 이다.

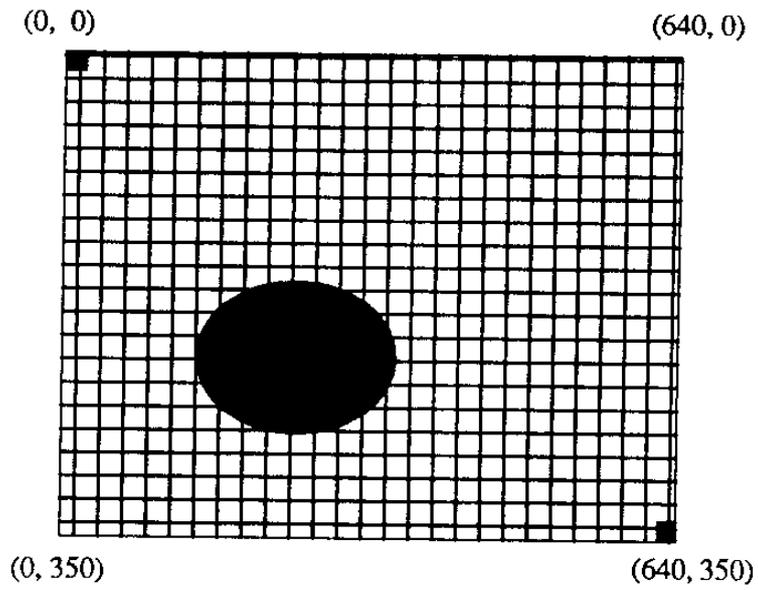


그림 3.5 Window in 640 \* 350 Graphic Mode

본 연구에서 개발된 윈도우와 메뉴함수는 텍스트 모드를 사용하고 있으며, 그래픽 함수는 그래픽 모드에서 정의되어 사용된다.

### 3.3.2 윈도우 함수

K-CLIPS는 사용자에게 필요한 정보를 편리하게 전달할 수 있는 윈도우를 제공한다. 즉 정보를 필요로 할 경우 화면상에 알맞은 크기의 윈도우를 열어 사용자가 참조하도록 하고 불필요할 경우 열린 윈도우를 닫게 하는 기능을 제공한다. 이런 기능은 아래의 함수에 의해 구성된다.

#### 1) window\_open 함수

이 함수는 처음 윈도우를 선언할 경우에 먼저 사용해야 한다. 여기서는 열고자 하는 윈도우의 번호와 크기를 결정한다.

(window\_open <window\_id> <left> <top> <right> <bottom>)

<window\_id>는 윈도우의 번호를 선언하는 것으로 이 번호는 앞으로 이 윈도우를 사용할 경우 이 번호만 호출하면 윈도우와 연결이 된다. <left><top><right><bottom>은 윈도우의 크기를 결정하는 값을 표시한다.

#### 2) window\_close 함수

이 함수는 열려진 윈도우 중 원하는 윈도우를 닫고자 할 경우 사용한다.

(window\_close <window\_id>)

<window\_id>를 지정하지 않고 (window\_close)만 사용한 경우는 열린 윈도우는 모두 닫히게 된다.

#### 3) window\_select 함수

이 함수는 선언된 윈도우중 원하는 윈도우를 다시 화면상에 나타내고자 할 경우 사용한다.

(window\_select <window\_id>)

#### 4) window\_move 함수

이 함수는 선언한 윈도우의 위치를 변경하고자 할 경우에 사용된다.

(window\_move <window\_id> <x> <y>)

<x>, <y>는 윈도우의 상부좌측의 좌표를 나타낸다.

#### 5) cprintout 함수

이 함수는 윈도우에 정보를 프린트하는 함수로서 printout과 유사한 기능을 가지고 있다.

(부록 1 참조)

(cprintout <item>...<item> [crlf])

#### 6) window\_remove 함수

윈도우를 더 이상 열 필요가 없을 경우에는 윈도우가 차지하는 스택을 줄이기 위해 정의된 윈도우를 메모리에서 제거하는 함수이다. 이 함수에 의해 선언된 사용한 윈도우는 다시는 화면상에 나타나지 않으며 나타내고자 할 경우는 재선언이 필요하다. 즉 window\_open를 다시 사용해야 한다.

(window\_remove <window\_id>)

위 함수를 사용한 간단한 규칙을 만들면

```
(defrule test-window
```

```
  (WINDOW INDEX ?windiw-index)
```

```
  (WINDOW TITLE ?title)
```

```
  (MESSAGE ?string)
```

```
=>
```

```
(window_open ?window index 10 10 25 25)
```

```
(window_title ?title)
```

```
(cprintout "This Window Meassage ." crlf ?string )
```

(pause)

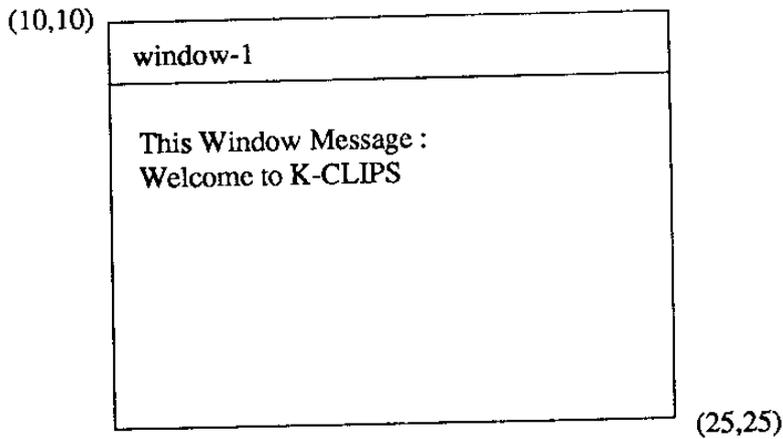
(window\_close ?window index))

(assert (WINDOW INDEX 1))

(assert (WINDOW-TITLE window-1))

(assert (MESSAGE "Welcome to K-CLIPS"))

이 규칙이 수행되면 화면상에는 다음과 같은 윈도우가 구성된다.



### 3.3.3 메뉴 함수

메뉴함수는 사용자의 입력을 편리하고자 할 경우 사용되는 것으로 다음과 같은 포맷을 가진다.

```
(make_menu <menu-id> <x> <y>
```

```
< " argument-1 " >
```

```
< "argument-2 " >
```

```
.....
```

```
< "argument-n " > )
```

여기서 <menu-id>는 사용하는 메뉴의 번호를 언급해야 하며, <x>,<y>는 메뉴

의 상부좌측의 좌표점을 나타낸다. 그리고 <"argument-1"> ...<"argument-n"> 는 메뉴에 사용될 항목별 문자열을 말한다.

예를들면 부재로 사용할 요소를 결정할 경우 다음과 같이 정의한다.

```
(make_menu 1 20 20
  " Column Element "
  " Beam Element "
  " Wall Element "
  " Brace Element "
  " Slab Element ")
```

여기서 첫번째의 상수는 메뉴의 번호를 지칭하고 두번째, 세번째 상수는 메뉴의 위치를 나타낸다. 그리고 그다음 문자열(string)은 원하는 항목을 나타내고 있다. 이 함수를 수행하면 화면에는 아래의 그림과 같이 나타난다. 각 항목을 결정할 경우는 Down키와 Up키를 사용하여 항목을 결정한 후 리턴키를 사용하면 원하는 값을 입력할 수 있다. 이 함수와 윈도우 함수와 복합하여 사용하면 더욱 편리한 시스템을 구축할 수 있다.

(20,20)

Column Element
Beam Element
Wall Element
Brace Element
Slab Element

### 3.3.4 그래픽 함수

여기서 제공하는 함수는 TURBO-C에서 제공하는 그래픽 함수를 기본으로하여 구축한 것으로 함수의 명칭은 동일하게 사용하였다. 현재 사용할 수 있는 함수를 크게 분류하면 다음과 같다.

#### 1) 조절 함수

- a) `initgraph` : 그래픽모드를 선언하는 함수로 가장 먼저 선언한다.
- b) `closegraph` : 그래픽모드에서 끝난 후, 텍스트모드로 되돌아가는 함수.

#### 2) 선언 함수

- a) `setcolor` : 사용할 색을 정의하는 함수.
- b) `setlinestyle` : 사용할 선의 종류와 두께를 선언한다.
- c) `setfillstyle` : 해칭선의 색과 종류를 선언 한다.
- d) `settextjustify` : 문자의 위치의 결정한다.
- e) `settextstyle` : 사용할 문자의 종류와 크기를 결정한다.

#### 3) 그래픽 함수

- a) `arc` : 원호를 그리는 함수.
- b) `circle` : 원을 그리는 함수.
- c) `drawpoly`, `fillpoly` : 연속적인 선을 그리는 함수.
- d) `ellipse`, `fillellipse` : 타원을 그리는 함수.
- e) `line`, `linereel`, `lineto` : 직선을 그리는 함수.
- f) `moveto`, `moverel` : 현재 좌표점에서 지정하는 위치로 옮기는 함수.
- g) `rectangle` : 사각형을 그리는 함수.
- h) `bar`, `bar3d` : 해칭선을 가지는 사각형을 그리는 함수.
- i) `pieslice` : 파이모양을 그리는 함수.
- j) `sector` : 타원의 일부분을 나타내는 함수.

#### 4) 문자 함수

- a) `outtext` : 현재 좌표점에서 문자를 프린트하는 함수.

b) outtextxy : 지정된 좌표점에서 문자를 프린트하는 함수.

### 3.3.4.1 조절함수

#### 1) initgraph 함수

(initgraph <color> )

initgraph 함수는 그래픽 모드를 시작할 때는 반드시 사용해야 하는 함수로서 앞으로 사용할 그림의 색을 선정할 수 있게 하였다. <color>는 표3.2에 언급된 값 중하나를 선택해야 한다.

표 3.2 Type of color

Numeric Value	Symbolic Name
0	BLACK
1	BLUE
2	GREEN
3	CYAN
4	RED
5	MAGENTA
6	BROWN
7	LIGHTGRAY
8	DARKGRAY
9	LIGHTBLUE
10	LIGHTGREEN
11	LIGHTCYAN
12	LIGHTRED
13	LIGHTMAGENTA
14	YELLOW
15	WHITE

#### 2) closegraph 함수

(closegraph)

closegraph 함수는 그래픽 모드를 빠져나올 경우에 사용해야 한다.

### 3.3.4.2 선언함수

#### 1) sector 함수

(setcolor <color>)

sector 함수는 initgraph 함수에서 선정된 색을 바꿀 경우에 사용되는 함수로서 <color>는 새로이 결정될 색이 지정된다. 표 3.2를 참조하라.

#### 2) setlinestyle 함수

(setlinestyle <linestyle> <thickness>)

setlinestyle 함수는 선의 형태 (line style)과 선의 굵기 (line thickness)을 결정하는 함수로서 <linestyle>에서는 표 3.3에 나타난 선의 형태중 하나를 결정하는 해야 하며, <thickness>에는 표 3.4에 나타난 선의 굵기 중 하나를 결정해야 한다.

표 3.3 Line-type 의 종류

Name	Value	Description
SOLID LINE	0	solid line
DOTTED LINE	1	dotted line
CENTER LINE	2	centered line
DASHED LINE	3	dashed line

표 3.4 Line의 Thickness

Name	Value	Description
NORM WIDTH	1	1 pixel wide
THICK WIDTH	3	3 pixels wide

#### 3) setfillstyle 함수

(setfillstyle <pattern> <color>)

setfillstyle 함수는 해칭에 사용할 형식과 해칭선의 색을 선정하는 함수이다. <pattern>에는 해칭의 타입을 결정하는 변수로서 표 3.5에 표시된 값중 하나를 선택해야 한다. <color>는 해칭선의 색상을 결정하는 변수로서 앞서 언급한 표3.2를 참조하여 선택해야 한다

표 3.5 해칭의 종류

Name	Value	Description
EMPTY FILL	0	fill with background color
SOLID FILL	1	solid fill
LINE FILL	2	fill with
LTSLASH FILL	3	fill with ///
SLASH FILL	4	fill with //, thick lines
BKSLASH FILL	5	fill with \, thick lines
LTBKSLASH FILL	6	file with \
HATCH FILL	7	light hatch fill
XHATCH FILL	8	heavy cross-hatch fill
INTERLEAVE FILL	9	interleaving line fill
WIDE DOT FILL	10	widely spaced dot fill
CLOSE DOT FILL	11	closed spaced dot fill

#### 4) settextjustify 함수

(settextjustify <horiz> <vert>)

settextjustify 함수는 문자의 text justification을 결정하는 함수로서 default 는 수평에 대해서는 LEFT-TEXT이고 수직에 대해서는 TOP-TEXT 이다. <horiz>와 <vert> 에서 사용할 변수값은 표 3.6에 나타난 값 중 하나이어야 한다.

표 3.6 Text의 Justify

Name	Value	Description
LEFT TEXT	0	horiz
CENTER TEXT	1	horiz and vert
RIGHT TEXT	2	horiz
BOTTOM TEXT	3	vert
TOP TEXT	4	vert

### 5) settextstyle 함수

(settextstyle <font> <direction> <charsize>)

settextstyle 함수는 문자의 형태와 문자가 쓰여지는 방향과 문자의 크기를 결정하는 함수이다. <font>에는 문자의 형태가 결정되어야 하는 데, 여기서 활용할 수 있는 문자의 형태는 표 3.7에 언급한 것 같이 하나의 8\*8 bit mapped 폰트와 여러개의 stroked 폰트를 사용할 수 있다. <direction>에는 문자가 쓰여지는 방향을 결정하는 변수로 수직인지 수평인지를 결정해야 한다. 그리고 <charsize>는 각 문자의 크기를 결정한다.

표 3.7 Text의 Font

Name	Value	Description
DEFAULT FONT	0	8*8 bit-mapped font
TRIPLEX FONT	1	stroked triplex font
SMALL FONT	2	stroked small font
SANSSERIF FONT	3	stroked sans-serif font
GOthic FONT	4	stroked gothic font

표 3.8 Text의 Direction

Name	Value	Description
HORIZ DIR	0	left to right
VERT DIR	1	bottom to top

표 3.9 Text의 Size

Size	Description
charsize = 0	stroked font만 사용 (* 4)
charsize = 1	8*8 pixel rectangle
charsize = 2	16*16 pixel rectangle

### 1.3.4.3 그래픽 함수

#### 1) arc 함수

(arc <x> <y> <stangle> <endangle> <radius>)

arc 함수는 원점이 (<x>,<y>)이며 반경이 <radius>이고, 시작점의 각도 <stangle>에서 종점의 각도가 <endangle>인 원호를 그리는 함수이다. 원호의 각도는 반시계방향으로 계산된다. 3시방향이 0도이며, 12시방향이 90도를 가리킨다.

#### 2) circle 함수

(circle <x> <y> <radius>)

circle 함수는 원점이 (<x>,<y>)이며 반경이 <radius>인 원을 그리는 함수이다. 원의 표시하는 선의 굵기는 조절이 가능(setlinestyle 함수 사용)하나 선의 형태는 직선으로 고정되어 있다.

#### 3) drawpoly 함수

(drawpoly <x1> <y1> <x2> <y2> ... <xn> <yn>)

drawpoly 함수는 현재의 선의 색상과 형태로 (<x1> <y1>) 위치에서 (<xn>,<yn>) 위치까지 연속적인 직선을 그리는 함수이다.

#### 4) ellipse 함수

(ellipse <x> <y> <stangle> <endangle> <xradius> <yradius>)

ellipse 함수는 타원을 그리는 함수로서, 원점은 (<x>,<y>)에 있고 수평축 반경은 <xradius>이고 수직축 반경은 <yradius>를 지닌다. 시작점의 각도 <stangle>이며 종점의 각도가 <endangle>이다. 호의 각도는 반시계방향으로 계산된다. 3시방향이 0도이며, 12시방향이 90도를 가리킨다.

#### 5) line 함수

(line <x1> <y1> <x2> <y2>)

line 함수는 현재 선정된 선의 색상, 형태, 굵기를 사용하여 절점 (<x1>,<y1>)과 절점 (<x2>,<y2>)사이를 잇는 직선을 그리는 함수이다. 이 함수를 사용하면 현재의 위치는

변화가 없다.

6) linerel 함수

(linerel <dx> <dy>)

linerel 함수는 현재의 위치에서 상대적인 거리 (<dx>,<dy>) 만큼 선을 잇는 함수이다.

7) lineto 함수

(lineto <x> <y>)

lineto 함수는 현재의 위치에서 절점 (<x>,<y>) 까지 직선을 잇는 함수이다.

8) moveto 함수

(moveto <x> <y>)

moveto 함수는 현재의 위치를 절점 (<x>,<y>)로 옮기는 함수를 말한다.

9) moverel 함수

(moverel <dx> <dy>)

moverel 함수는 현재위치에서 상대적인 거리 (<dx>,<dy>)만큼 현재위치를 변경하는 함수이다.

10) rectangle 함수

(rectangle <left> <top> <right> <bottom>)

rectangle 함수는 현재의 선의 색상,굵기,형태로 직사각형을 그리는 함수이다. <left>,<top>)은 직사각형의 왼쪽상단을 가리키며, <right>,<bottom>은 오른쪽 하단을 가리킨다.

11) bar 함수

(bar <left> <top> <right> <bottom>)

bar 함수는 rectangle 함수와 유사하나 사각형내부가 선정된 색의 해칭이 나타난다.

12) bar3d 함수

(bar3d <left> <top> <right> <bottom> <depth>)

bar3d 함수는 bar의 3차원적인 표현을 하는 함수로서 <depth>에는 두께의 크기가 결정되어야 한다.

13) fillellipse 함수

(fillellipse <x> <y> <xradius> <yradius>)

fillellipse 함수는 ellipse 함수와 동일하나 타원내부에 선정된 색의 해칭이 존재하는 것이 다르다.

14) fillpoly 함수

(fillpoly <x1> <y1> <x2> <y2> ... <xn> <yn> )

fillpoly 함수는 drawpoly와 동일하나 구성된 다각형 내부에 선정된 색의 해칭이 존재하게 된다.

15) floodfill 함수

(floodfill <x> <y> <border>)

floodfill 함수는 위치 (<x>,<y>)에서 <border> 에 결정된 색상으로 둘러싸인 부분을 fill시키는 함수이다.

16) pieslice 함수

(pieslice <x> <y> <stangle> <endangle> <radius>)

pieslice 함수는 arc 함수와 유사하나 원호와 원호의 끝점과 원점을 잇는 부채꼴 내부에는 선정된 색의 해칭이 나타난다.

17) sector 함수

(sector <x> <y> <stangle> <endangle> <xradius> <yradius> )

sector 함수는 fillellipse 함수와 동일하나 타원의 부분을 표시할 수 있다.

1.3.4.4 문자함수

1) outtext 함수

(outtext <textstring> )

outtext는 현재의 지점에서 선언된 justification, 폰트, 방향, 크기를 사용하여 입력된 문자열인 <textstring>를 화면에 표시하는 함수이다.

2) outtextxy 함수

(outtextxy <x> <y> <textstring>)

outtextxy는 지정된 좌표점 (<x>, <y>) 에서 선언된 justification, 폰트, 방향, 크기에 따라 입력된 문자열 <textstring>를 화면에 표시하는 함수이다.

#### 4. 참고 문헌

##### A. 인공 지능에 관한 문헌

1. P.Harmon and D. King, Artificial Intelligence in Business : Expert System, John Wiley and Sons, New York, 1985.
2. Hayes-Roth F, Waterman D.A, Lenat D.D., Building Expert System, Addison-Welsey, 1983.
3. D.A.Waterman, A Guide to Expert System, Addison-Wesley, US, 1986.
4. B.G.Buchanan and E.H.Shortlife, Rule-based Expert Systems, Addison-Wesley, US, 1984.
5. S.Weiss and C.Kulikowski, A Practical Guide to Designing Expert Systems, Rowman and Allandheld, Totowa,NJ, 1984.
6. C.Towmsend and D.Feucht, Design and Programming Personal Expert Systems, TAB, 1986.
7. L.Brown, R.Farrel, and E.Kant etc., Programming Expert System in OPS5, Addison-Welsey, 1986.

##### B. 전문가시스템 도구에 관한 문헌

1. F.Chehayeb and J.Conner, GEPSE-A Computer Environment for Engineering

- Problem Solving, Research Report R86-11, Dept. of Civil Engineering, MIT, 1985.
2. 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.
  3. Mark H. and Richer, An Evaluation of Expert System Development Tools, Expert System, July, Vol.3, No.3, 1986.
  4. Phillip J.L., William J.G., and Del D., Expert System Tools for Civil Engineering Applications, Expert Systems in Civil Engineering, 1986.
  5. Joseph C. Giarrantano, CLIPS User's Guide, 1987.
  6. Sobelman and Krekelberg, Advanced C : Techniques and Applications, QUE, 1985.
  7. M.Waite, S.Prate, and D.Martin, C Primer Plus, Howard W.Sams & Co.,Inc., 1984.
  8. B.W.Kernigham and D.M.Ritchie, The C programming Language, Prentice-Hall, 1978.
  9. Herbert Schilt, C : The Complete Reference, McGraw-Hill, 1987.

#### C. 전문가시스템에 관한 문헌

1. D.Sriram, M.I.Maher and S.J.Fenves, Knowledge-based Expert System in

- Structural Design, Computers and Structures, Vol.20, No.1-3, pp. 1-9, 1985.
2. D.Sriram, DENSTINY : A Model for Integrated Structural Design, Artificial Intelligence, Vol.1, No.2, pp.109-116, 1986.
  3. M.L. Maher, Expert System for Civil Engineering : Technology and Application, ASCE.
  4. M.L. Maher and S.J.Fenves, HI-RISE : A Knowledge-Based Expert System for the Preliminary Structural Design of High Rise Building, 1985.
  5. M.F.Rooney and S.E.Smith, Artificial Intelligence in Engineering Design, Computers and Structures, Vol.16, No.1-4, pp.279-288, 1985.
  6. Furuta,H.. Tu,K.S. and James,T.P., Structural Engineering Applications of Expert Systems, Journal of Computer Aided Design, Vol.17, No.9, pp410-419, 1985.
  7. W.J.Rasdof and G.C.Salley, Generative Engineering Database-Toward Expert Systems, Computers and Structures, Vol.20, No.1-3, pp11-15, 1985.
  8. S.J.Fenves and J.H.Garrett, Knowledge Based Standard Proceeding, Artificial Intelligence, Vol.1, No.1, pp.3-14, 1986.
  9. C.K.Soh and A.K.Soh, Example of Intelligent Structural Design System, Journal of Computing in Civil Engineering, Vol.2, No.4, October, 1988.
  10. M.S.Jones and V.E.Saouma, Prototype Hybrid Expert System for R/C Design,

Journal of Computing in Civil Engineering, Vol.2, No.4, April, 1986.

11. M.R.Wigan, Engineering Tools for Building Knowledge-based System on Microsystem, Microcomputers in Civil Engineering, pp.52-68, 1986.

12. D.Sriram, Knowledge-Based Approaches For Structural Design, Topics in Engineering Vol.1 edited by C.A. Brebbia and J.J.Connor, Computational Mechanics, 1987

13. M.L.Maher, Problem Solving using Expert System Techniques, Expert System in Civil Engineering edited by C.N.Kostem and M.L.Maher, April 1986

14. C.K.Choi and E.D.Kim, BUILD-A : A Three Dimensional Analysis Program for Building Structure, Proceedings of the Second International Conference on Civil and Structural Engineering Computing : CIVIL-COMPO 85, London, England, 1985.

15. C.K.Choi and E.D.Kim, A Preliminary Model of I-BUILDS : An Intelligent Building Design System, Proceedings of Second International Conference on Application of Artificial Intelligence in Engineering, Knowledge Based Expert System in Engineering : Planning and Design, Edited by D.Sriram and R.A.Adey, Computational Mechanics Publications, Boston,USA, 1987.

16. 한국과학기술원, 인공지능을 이용한 고도의 구조해석/설계용 전문가시스템의 개발 (I,II), 과학기술처, 1988-89

## **PART II : 건물설계 전문가 시스템**



## 1. 서론

### 1.1 연구의 목적

지난 수 십년 동안의 컴퓨터 과학과 관련분야의 기술은 그 양과 질에 있어 급속한 발전을 거듭하여 왔다. 이러한 고도의 기술은 컴퓨터의 소형화, 가격의 저렴화 추세를 가져왔고, 이러한 컴퓨터의 보급확대로 누구나 쉽게 컴퓨터 사용이 가능하여 해당분야의 작업능률을 향상시킬 수 있게 되었다. 그러나 이러한 하드웨어의 경향과는 달리 소프트웨어는 늘어나는 사용자의 요구를 만족시키지 못하고 있으며 이는 곧 각 분야에서 기존의 경험자들이 소프트웨어 개발을 위한 노력을 할 필요성을 나타낸다.

우리나라 건축분야에서의 컴퓨터 이용현황에 관하여 그 예를 들면 현재 대기업체에서 개인사무소에 이르기까지 초대형, 대형, 중형, 소형 및 개인용 컴퓨터등 수많은 종류가 보급되어 각 분야에서 광범위하게 이용되고 있다. 건축구조분야에서의 컴퓨터의 이용은 주로 구조물의 해석, 각 구성부재의 설계, 적산 및 견적, 시공관리, 그리고 각 작업에서 얻어진 결과들을 도면화하는 작업등이다. 실제 건축구조물의 해석과 설계분야의 컴퓨터 사용은 발달된 이론을 근거로 많은 프로그램이 개발되어 있으므로 현재는 적산, 시공관리 그리고 도면화의 작업등에 많은 시간과 노력이 요구된다. 후자의 분야에도 작금에는 CAD 시스템이 도입되어 건설산업을 고도화시키는데 크게 이바지하고 있다.

그러나 지금까지는 건축구조분야를 비롯한 각 분야에서의 컴퓨터 사용효율을 증대시키고자 하는 노력은 대부분이 기존의 절차식 언어(Procedural Language)에 기초하여 수치적인 계산을 위주로 하는 양적인 문제에 치중하여 왔다. 구조분야의 한 예를 들면 구조해석, 부재의 최적설계, 도면화 작업등 주로 정해진 일의 순서에 의한 사항을

컴퓨터에 의존하는 (Routine Design 이라고 칭함) 경향이였다. 그러나 최근에 들어서는 지식을 기반으로 하여 경험적 문제의 해를 구하는데 그 학문적 바탕이 마련되고, 실용화를 위한 연구결과가 기대를 모으고 있다. 이 방법은 직감에 의존하거나 또는 수치적으로 정식화(Formulation) 가 어려운 분야의 문제 즉, 기존의 전문가의 경험에 의존하게 되는 문제 (예를들면 판단, 감시, 창조적인 설계(Creative Design))의 분야에 적합하다.

지식기반 시스템은 인공지능의 여러분야에서 가장 활동적인 연구가 진행되는 분야로서, 이는 문제해결의 단계에서 지적인 능력을 나타내는 컴퓨터 프로그램이다. 이러한 지적인 능력은 특정한 분야에서의 문제해결전략에 대한 지식과 문제풀이과정에서 필요한 여러가지 일을 수행할 수 있는 방법에 관한 지식을 컴퓨터에 내장함으로써 가능하게된다. 여기서 지능이라는 말은 지식을 적용, 조절 및 획득하고 결론에 이르는 추론과정을 나타내는 능력으로 정의된다. 이러한 정의는 인간의 행위를 잘 표현한다고 볼 수 있는데, 컴퓨터에 의한 이러한 기능은 기존의 알고리즘식 언어로는 구현하기가 상당히 어려웠던 영역이다. 사람이 일을 수행하는 능력은 문제해결의 과정에서 사용하는 지식인데, 이는 대부분이 그 분야의 전문가에게 국한되어 있다. 만약 이러한 지식을 전문가로부터 얻어내어 컴퓨터에 저장할 수 있다면, 특정분야에서 전문가가 문제를 해결하듯이 컴퓨터에 의해서도 어려운 문제를 해결할 수 있을 것이다. 이러한 개념으로 전문가 시스템이라는 지식베이스 시스템은 출현한지 얼마 되지는 않지만 각 분야에서의 역할은 보조기능이 아닌 전문가의 위치에서 문제해결과정을 통하여 질적 및 생산성을 향상시키는데 일익을 담당하게 될 높은 가능성을 가지므로 각 분야에서 활발한 개발이 진행되고 있다.

공학분야에서의 설계과정에서 전문가시스템이 지원가능한 단계를 예를들기 위하여

설계단계를 분석하면 다음과 같다.

1. 문제영역의 명시
2. 예비설계
3. 분석(해석 또는 세부설계)
4. 평가

설계자는 이러한 일련의 과정을 반복하여 설계에 대한 최적의 해를 구하게 된다. 그러나 이러한 설계과정에서 컴퓨터의 사용은 주로 해석과 세부설계분야의 수치적인 단계에서만 주로 이용되어 왔고, 또한 이러한 분야의 심오한 이론들은 정리되고 프로그래밍되어 현실적으로 많이 이용되고 있는 실정이다. 그러나 예비설계 또는 평가과정에서의 지식들은 체계화내지 정식화가 어렵기 때문에 오랜 세월의 경험을 가진 전문가의 지식에 의존할 수 밖에 없다. 이러한 연유로 인공지능의 한 분야로 새로운 문제풀이의 접근방법이 시도되어 소위 ill-structured 지식의 범주까지 컴퓨터로 대신하려는 지식기반 시스템의 등장에 이르게 되었다. 이러한 지식기반 전문가 시스템(Knowledge-based Expert System)은 주로 경험에 의존하는 문제의 해결에 적합하며 그 실효성은 몇 가지 연구사례가 입증하고 있다[참고문헌 A.1.2]. 그러므로 현재 컴퓨터를 고속의 전자계산기로 이용하는 패턴을 이러한 질적문제를 다루는 보다 높은 차원의 보조기구로서 이용하여 작업의 효율을 높이는 게 필요하다.

## 1.2 연구범위

본 연구는 건물의 구조설계 과정에서 일어나는 제반 작업과정을 기존의 언어(FORTRAN)로 프로그래밍된 일관설계용 패키지인 BUILDS의 기능 즉, 효율적인 3차원적 구조해석, 최적화 기법을 이용한 구조설계, 적산, 견적 그리고 도면화 프로

그럼을 전문가 시스템 에서 최대한 이용하고, 이러한 구조설계 과정에서 필요로 하는 지식 즉, 건물의 구조시스템의 선정, 구조물의 모델링, 입력의 이해와 필요한 데이터의 처리, 해석 및 설계결과에의 분석, 검토, 그리고 데이터의 수정등의 기존의 경험적 지식을 저장하여 BUILDS의 사용자를 지원해 주는 전문가 시스템을 구축한다.

그리고, 전문가의 경험적 지식을 요구하는 분야의 문제, 즉 비정형화된 문제를 해결하기 위하여 전문가 시스템을 구축할 경우, 원형(prototype)을 설정하거나 보다 빨리, 쉽게하기 위하여 보통은 전문가 시스템 구축용 도구(tools)을 이용하게 된다. 이러한 경우에는 해당 도구가 가지는 제한성에 많이 영향을 받게 되는데, 대부분의 기존의 도구는 규칙의 지식의 표현으로 비결정적인 문제해결의 접근에 용이한 반면, 수치해석면에서는 효과적으로 대처하지 못하며, 더구나 기존의 FORTRAN, C등의 알고리즘 언어와 복합적인 시스템을 구축하기가 어렵게 되어있다. 이를 극복하기 위하여 본 연구에서는 지식을 규칙이나 알고리즘과정으로 쉽게 표현할 수 있는 전문가 시스템 구축용 도구인 K-CLIPS(KAIST-C Language Integrated Production System)를 사용한다. K-CLIPS는 기존의 LISP언어에 비해 C언어가 가지는 이식성(portability), 수치처리등의 장점을 이용하여 C언어로 되어있다. 또한 현재 생산되어 보급되는 고급언어들은 혼합된 형태로 사용이 가능하기 때문에(예를들면 C의 주프로그램과 FORTRAN의 부프로그램을 언어의 전환없이 각각 컴파일하여 연결할 수 있다.) 이미 개발되어 있는 프로그램과 복합된 통합시스템을 구축할 수 있게 하여준다. 본 연구에는 전문가 시스템 개발용 도구인 K-CLIPS를 사용하여 건물의 일관된 설계시스템인 I-BUILDS를 구축한다.

## 2. I-BUILDS의 개요

### 2.1 서론

건축가가 건물을 설계할 때는 여러가지의 설계원리, 구성부재의 조립, 시공등의 여러부분들과의 상호연관 관계를 고려하여 건물의 형태, 재료, 안정성, 방화 및 위생시설등의 법적 제한요소들을 만족하는지에 항상 관심을 두게 된다. 구조체는 이러한 건물이라는 유기체의 한 부분으로서 건물의 계획시 중요한 요소가 되며, 이를 구조기술자가 기능적인 공간에 부가물로서 계획이 된 이후에 처리하도록 하여서는 곤란하다. 특히 고층건물은 중력과 같은 수직력과, 바람 및 지진과 같은 수평력의 자연적인 제약과 건물 실내외의 온도, 습도차이, 공기의 압력등과 같은 환경적 제약조건을 만족할 수 있게 부재를 적절하게 배치, 접합하여 외력을 효과적으로 지반에 전달할 수 있게 하여야 한다.

구조적인 관점에서 보면 건물이 고층화됨에 따라 수평변위가 커지기 때문에 구조재료의 강도보다 구조물의 강성이 중요시된다. 건물의 강성은 주로 구조시스템에 좌우되며 이러한 구조시스템의 효율은 보통 사용된 재료의 양(예, 단위 면적당 무게)으로 평가된다. 이러한 연유로 건물에 가장 적절한 구조형식은 최소의 재료의 사용으로 최대의 강성을 가질 수 있는 것이라 할 수 있다. 구조형식을 결정하는 주요요인으로는 공사비, 설비비, 유지비등의 경제적인 면과 상부구조의 결정에 영향을 주는 지반조건, 스패수, 스패크기, 부재, 부재의 접합조건, 높이와 폭의 비, 방화시설, 해당지역의 법규, 건축재료의 유용성등이 있으며, 또한 사회적, 경제적, 문화적, 기술적 수준등도 포함된다[참고문헌 B.22-25].

그러나 지금까지의 공학설계에 컴퓨터를 활용하려는 노력을 분석해보면 주로 컴퓨

터가 가지는 특성, 즉 빠른 연산속도, 컴퓨터 그래픽스에 의한 시각적 영상효과, 그리고 많은 양의 데이터 저장에 중점을 두어왔기 때문에 위에서 언급한 바와 같은 여러 측면에서의 경험적 판단에는 적용하기가 어려웠고, 다만 세부단계에서 물량추정에 의한 경제성 문제에 국한된 결과를 바탕으로 설계자는 자신의 경험에 의존, 그 적절성을 판단하여 설계를 완성하였다. 이러한 문제는 인공지능의 기법의 발달로 인간의 경험에 의존한 지식 즉, 개념과 추론에 의한 결과의 도출과정등도 컴퓨터에 의한 설계과정의 한부분으로서 도입하게 됨으로서 자동화된 설계시스템을 구현하기 위한 기술적 토대를 마련하게 되었다.

일반적으로 지식을 습득해나가는 과정은 주로 정립된 이론에 근거하여 이루어지므로 처음으로 실제적인 업무에 나서는 사람은 경험의 부족상태에 있다. 구조공학분야에서도 처음에는 구조역학의 기본적인 이론에만 익숙해져 있기 때문에 여러조건이 주어지는 구조물의 실제설계, 특히 적절한 구조시스템을 선정하는 문제에는 많은 어려움이 뒤따른다. 이러한 분야는 그 이전의 축적된 경험을 요하는 부분이므로 지식의 체계화로 전문가 시스템의 도입이 필요함을 적절히 설명한다고 볼 수 있다.

본 연구는 위와 같은 제반 설계조건, 구조요소의 역학적 특성등에 관한 기존의 이론적 지식과 이 분야의 전문적인 경험적 지식을 바탕으로 자동화된 건물설계의 일관시스템을 개발하려는 노력의 하나로서, 특히 건물의 설계과정이 건축가와 건축주, 구조공학자, 설비자등의 여러분야의 참가자들이 상호 충분한 정보교환을 거듭하게 됨으로써 다방면의 전문가들의 협력에 의하여 진행이 되므로 이러한 일련의 설계과정을 일관된 모델로 정립하는데 그 목적이 있다. 이러한 전문가시스템을 이용함으로써 짧은 시간에 사람이 생각해 볼 수 있는 가능성보다 더 많은 때로는 가능한 모든 후보의 정보를 체크해 볼 수 있으므로 건물의 설계에 커다란 도움을 줄 수 있다.

## 2.2 건물설계의 일관시스템(BUILDS)

구조물을 완성하기 위해서는 보통 구조계획에 의한 기본설계를 바탕으로 구조해석, 부재의 설계, 도면화, 적산, 시공등의 여러과정들이 통하여 이루어진다. 이러한 작업들은 모두가 각각 독립적인 것이 아니라 서로가 연계되어 있으므로 단일 목적을 행하는 현재의 각 소프트웨어를 일관된 하나의 시스템으로 통합하는게 필요하다. 이러한 시스템은 컴퓨터와 사용자간의 대화내용을 최소화하여 사용자는 단지 간단한 초보적인 입력만으로 일련의 설계과정을 제어할 수 있어 시스템의 깊은 지식에 관한 이해를 할 필요가 없다. 건축물의 구조해석에서 도면화까지의 일련의 작업을 검토해보면 개인적인 관점에 따라 다소 차이는 있겠지만 보통은 다음 그림 2.1과 같이 나타낼 수있다.

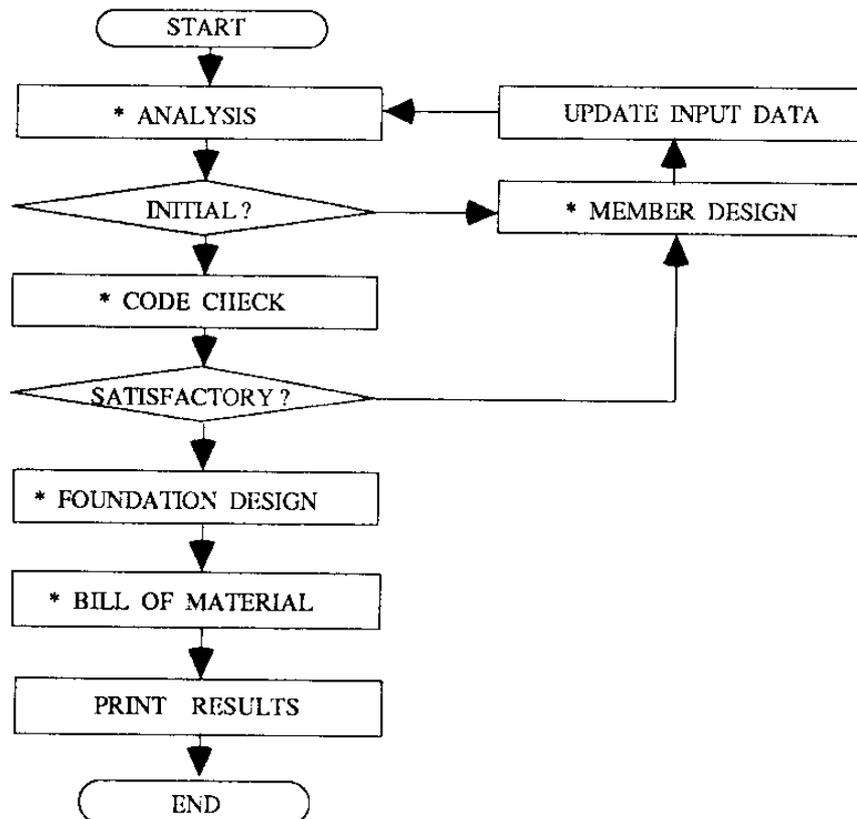


그림 2.1 건축 구조설계의 흐름도

즉 처음에는 개략적인 초기설계에 대한 입력데이터를 작성하여 구조해석을 수행하고 그 결과에 따라 부재의 설계가 끝나면 타당성을 판단하기 위하여 해석결과가 설계기준이 요구하는 조건이나 기타의 제한사항들을 위반하는 지를 검토한다. 만약 만족스럽지 못하다면 해당하는 부재의 성질을 수정, 변경하여 다시 반복적인 작업을 하게 되고, 만족한다면 상부구조의 설계는 끝이 나게된다. 그러면 기초설계를 행하여 최종적으로 건적과 도면화작업이 이루어지게 된다. 이러한 실제적인 건축구조물의 설계모델로 각 기능을 수행하는 Software를 구성함으로써 설계작업의 능률을 크게 향상시킬 수 있다.

BUILDS 프로그램은 건물구조의 해석에서부터 도면화작업을 위와같은 설계모델로 수행할 수 있게 구성된 건물설계의 일관시스템 (an integrated BUILDing Design System)이다. BUILDS에 의한 건물설계는 건축구조물의 형상의 선정으로부터 시작된다. 여기서 건물은 각 층에서 강하게 연결된 임의의 방향으로 된 평면프레임의 조합체로 모델링된다. 이러한 건물에 대한 3 차원적 선형해석이 가해지는 하중에 대해 수행되어 건물의 거동이 구해진다. BUILDS-A에 의한 건물의 거동이 구해지면 부재의 설계가 BUILDS-C/S에 의하여 행하여 지는데 이 과정은 최적의 해가 이루어질 때까지 반복된다. 이 두개의 부시스템은 최적이론의 실제적인 이용에 바탕을 두어 빠른 시간에 최적해에 접근해 가도록 하는 기법을 이용한다. 부재의 설계가 끝나게 되면 BUILDS-F에 의하여 기초의 최적설계가 이루어진다. 그리고나서 BUILDS는 BUILDS-B에 의하여 건적이 행하여지고 이 조건이 만족되면 BUILDS-D에 의하여 도면화 작업이 이루어져 건물의 전체작업은 완성된다. 만약 비용이 요구조건에 위반될 경우에는 구조해석-부재설계-건적의 과정이 새로운 구조물의 모델에 대하여 반복된다.

BUILDS는 한국과학기술원 토목공학과에서 개발된 프로그램으로서 건물구조를 설

제한 경우에 컴퓨터의 도움을 구조공학자에게 실제적인 도움을 줄 수 있게 하기 위한 목적으로 개발되었다 [참고문헌 B.12-16]. 이 시스템은 그림 2.2 와 같이 그 기능에 따라 여러개의 부시스템으로 구성되어있다. BUILDS는 각각 다른 기능을 가진 위와같은 부시스템을 유기적으로 연결해 놓은 설계의 일관시스템을 위한 소프트웨어이다.

이 시스템은 시스템 전체를 일관하여 자동적으로 수행할 수 있는 것을 주 목적으로 하고 있으나, 목적에 따라서는 각 부 시스템만을 사용할 수 있다. 즉 그림 2.1의 흐름도에서 \*로 표시한 단계에서 사용자가 결과를 검토하고 필요에 따라 데이터를 수정하여 재설계를 할 수 있게 하는등 전체시스템을 단계별로 사용할 수 있게 하여 에러에 의하여 전체과정을 반복하는 등의 손실을 사전에 방지하여 효율성을 기한다. 각 부시스템간의 정보교환은 다음 그림 2.3과 같다.

#### 구조해석 프로그램 (BUILDS-A)

건물구조의 해석은 BUILDS시스템 중에서 가장 중요한 부분이 다. 프로그램의 특징으로는

- 1)개구부를 갖는 전단벽 요소와
- 2)건물의 새로운 구조해석의 기법으로서 건축물의 건설과정에 따른 사하중의 점진적 재하의 특성을 등을 고려하였고, 그리고 건물의 특성을 이용하여 입력데이터의 간략화 및 최소화, 마지막으로 컴퓨터 그래픽스를 이용하여 입력 및 출력데이터를 그림으로 도시하였다.

여기에 덧붙여 동력학적인 해석을 수행할 수 있는 능력의 프로그램을 내장하였는데 이를 열거하면 다음과 같다.

B	-----	BUILDS-A : Analysis of building structure
U	-----	BUILDS-C : RC member design
I	-----	BUILDS-S : Steel member design
L	-----	BUILDS-F : Foundation design
D	-----	BUILDS-B : Bill of material/cost estimation
S	-----	BUILDS-D : Drafting

그림 2.2 BUILDS의 부시스템

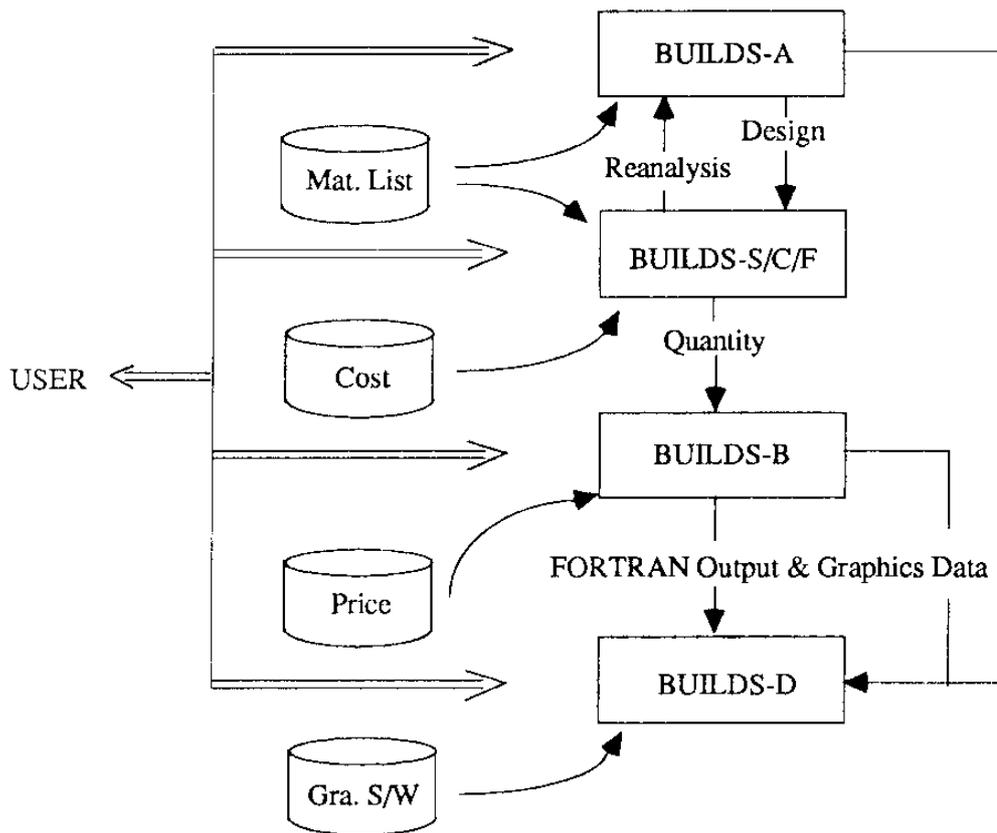


그림 2.3 BUILDS의 부시스템간의 연계

○ 동가정적해석법 : 우리나라의 내진규준에 의한 지진하중을 간단한 입력으로만으로 산정하여 해석을 할 수 있다.

○ 동적해석법 : 동적해석법이 요구될 경우에도 간단한 입력만으로도 필요한 모드수를 자동으로 선택하여 각 모드의 형상을 구하고 해당모드에 대한 결과를 중첩하여 원하는 결과를 구할 수 있다.

○ 모드 중첩법 : 동력학적 이론에 의한 일반적인 동적해석을 원할 경우에는 모드 형상과 진동수, 모드중첩법에 의한 시각력해석, 그리고 모드중첩법에 의한 응답스펙트럼 해석이 가능하다.

○ 직접적분법 : 동적하중에 대하여 가장 강력하고 상세한 해석 방법인데 앞의 방법을 사용할 수가 없거나 건물의 비선형해석을 할 경우에 이용된다.

#### 철근 콘크리트 부재의 설계 프로그램 (BUILDS-C)

현재 우리나라의 철근콘크리트 설계는 주로 설계자의 경험과 지식을 토대로 하여 이루어지고 있으며, 구조물을 설계함에 있어 많은 시간과 노력이 요구되고 있는 실정이다. 따라서 본 설계시스템의 개발의 목적은 구조물의 설계를 수행함에 있어 소요되는 많은 인원과 시간, 비용등의 과대한 지출을 막고, 설계의 초기단계에서 개략적인 사용단면 및 가격등을 축적된 지식을 활용하여 결정하고, 필요에 따라서는 최종결과까지도 컴퓨터에 의하여 수행할 수 있도록 하는데 있다.

이를 위해서 프로그램의 개발에서 채택한 설계방법은 허용응력설계법을 근간으로하여 철근콘크리트 구조물을 설계하며 이론과 기존의 설계도의 검토를 통하여 실제 현장에서 사용하고 있는 단면의 종류를 산정하여 각 단면에 대한 저항능력과 단면재원 등에 대한 데이터베이스를 구축한다. 이러한 자료를 이용하여 필요한 관계식을 구성해

됨으로서, 실제 부재의 설계과정은 연속해의 결정과정과 불연속해의 결정과정의 두 단계로 분리된다. 연속해의 결정과정에서는 산출된 관계식으로부터 부재의 응력조건에 따라 반복되는 작업없이 직접 어느 특정단면을 결정하게 되며, 불연속해 결정은 앞의 연속해 결정과정에서 산출된 단면을 중심으로 모든 제한 조건을 만족하는 범위에서 비용이 가장 적게드는 단면을 선택한다. 즉 주어진 단면형상에서 응력조건에 맞게 철근량을 결정하는 현재의 설계방법과는 달리 주어진 응력조건에 대해 단면형상 및 철근량을 동시에 결정하는 실용적이고 경제적인 설계를 수행하게 되며, 필요이상의 재료 및 비용을 방지할 수 있는 추가적인 효과를 준다.

#### 철골부재의 설계 (BUILDS-S)

BUILDS-S는 BUILDS-A에 의한 건물구조의 해석결과에 따라 현재의 대한건축학회의 강구조 계산규준을 적용하여 철골부재의 최적설계를 수행할 수 있도록 구성되어있다. 기존의 최적설계기법에 관한 이론과 방법은 대단히 많은 설계변수를 갖는 고층건물의 설계에는 적용이 어려우므로 BUILDS-S에서는 설계시에 요구되는 여러 설계조건들을 건물전체 Level의 제한조건 (건물의 변위, 고유진동수등) 과 각 부재 Level의 제한조건 (부재의 응력, 단면의 크기)으로 나누어 적용시키는 새로운 최적화 알고리즘에 따라 철골건물의 최적설계를 수행한다. 특히 건물 전체 Level의 제한조건을 만족시키기 위하여 사용되는 민감도해석과 규격표내에서 불연속 최적해를 효율적으로 구할 수 있게 하기위한 새로운 관계식을 구하여 사용함으로써 실용적이고 경제적인 설계시스템이 되었다.

#### 건물 기초의 설계 (BUILDS-F)

이 프로그램은 건물의 설계시 가장 필수적이고 복잡한 문제중의 하나인 기초설계에 대하여 프로그래밍하였으며, 이를 BUILDS-A와 BUILDS-D에 연계시켰다. 프로그램의 특징으로는 과거의 기록, 실험, 토질기술자의 판단등에 의하여 기초의 적합한 형태를 설정한 다음, 얇은 기초의 경우에는 판용법을 사용하여 알고리즘을 개발하였고 깊은 기초의 경우에는 파일기초에 대하여 파일의 배치에서 안정성을 검사하는 과정까지를 전산화하여 사용자가 여러형태의 기초설계를 컴퓨터로 행할 수 있게 하였다.

본 프로그램에서 개발한 기초의 형태는 다음의 네가지이다. 얇은 기초로는 독립확대 기초, 복합기초, 전면기초이고, 깊은 기초는 파일기초에 대한 것이다. 여기서 독립확대기초는 가장 널리 사용되는 얇은 기초의 형태로서 지반이 안정할 때 가장 경제적이다. 복합기초는 독립확대기초를 설치하고자 할 때, 편심이 발생하거나 두 개의 기초가 매우 근접하거나 겹치는 부분이 생길 경우에 사용된다. 전면기초는 독립확대기초의 저면적의 합이 건물전체 평면적의 반이상이 넘을 경우이거나, 지반의 지지력이 너무 작아 다른 종류의 기초로는 상부구조물의 하중을 지지하기가 어려울 때, 강성지반에 높은 지반응력집중을 줄이고자 할 때, 불균등 지반일 때, 지반이 공동(Cavity)이나 정의하기 어려운 압축성 Lense의 흙을 포함하고 있을 때, 건물의 주위에 하중을 분포시키고자 할 때, 양압력에 저항하고자 할 경우에 사용된다. 이와같은 얇은기초에 대해서는 판용법을 적용하여 단면설정에서 철근배치에 이르는 상세설계까지 하였다.

파일기초는 깊은 기초의 형태 중에서 가장 많이 사용되며 많은 침하가 예상되거나 지표근처의 지층이 상부구조물을 지지 할 수 없는 경우, 또는 기초에 무시할 수 없는 사력(Inclined Forces) 이 작용하는 경우나 흙의 침식가능성이 있을 때 사용된다. 본 프로그램에서는 단파일 해석시에 수직하중에 대해서는 Load-transfer 방법을 사용하고 수평하중에 대해서는 지반반력계수를 이용한 해석법을 사용하였다. 그리고 3 차원 균

파일을 위해서는 유한요소법을 사용하여 설계와 해석을 동시에 할 수 있다.

본 프로그램에서 개발한 프로그램은 현재까지의 단편적 프로그램과는 달리 건물기초의 일반적인 기초형태의 해석과 설계가 가능한 종합적 프로그램이다. 또한 BUILDS-F는 BUILDS-A와 연결되어 있으므로 기초에 관한 전문적인 지식이 없는 구조공학자 입지라도 쉽게 기초를 설계할 수 있으며 이미 시공된 기초의 해석시에도 이용할 수 있다. 따라서 BUILDS-F는 실무에 상당히 유용하게 사용될 수 있어 기초의 해석 및 설계시 드는 비용과 시간을 상당히 절약할 수 있다.

#### 견적 및 적산 프로그램 (BUILDS-B)

BUILDS-C/S/F의 각 부시스템에서 최적설계가 끝나고나면 설계된 구조시스템에 관한 견적을 수행하게 된다. 우선, 필요한 자재표 (Bill of Material)가 설계결과에 따라 작성되고, 해당공사를 완성하는데 필요한 인원, 자재의 종류와 단가를 기억시켜 놓은 Database를 이용하여 견적을 완성한다. 또한 이 시스템에서는 대상건물의 모델링에서 해석, 설계, 적산과정에 이르는 모든 과정에 대한 구조계산 보고서를 작성한다.

#### 도면화 프로그램 (BUILDS-D)

본 프로그램에서는 철골건물만(철근콘크리트 건물은 차후에 추가 예정)의 시공에서 요구되는 구조도면을 Plotter에 의하여 자동출력할 수 있도록 BUILDS-D를 개발하였다. 이는 BUILDS내에서 해석과 설계과정에서 나타나는 정보를 필요에 의해 도면화시킬 수 있도록 하였다. 자동출력되는 도면의 종류로는 도면의 목차역할을 하는 도면명 목록표, 기초평면도, 기초와 기둥의 연결을 위해 사용되는 기초판의 재원과 전체 건물 평면내에 위치하는 기둥열과 기초판 배열도, 기초와 기초판의 설치형상을 자세히

나타낸 기초의 기초판 상세도, 건물내에 쓰이는 기둥, 보, 거더에 대한 형강 제원을 나타내는 부재일람표, 각 층들의 층별평면도, 각 프레임의 입면도, 마지막으로 기둥열 내에 사용되는 형강제원이나 접합위치등이 기재되는 기둥열상세도로 현재 8 가지의 도면이 작성된다.

BUILDS-D에서는 사용자의 요구에 따라 도면의 크기를 지정함으로써 KS에서 규정하는 A0 - A6, B1 - B6크기의 도면을 출력할수 있도록 하였다. 또한 모든 도면을 연속해서 출력할 수 있지만 필요시 2 층 평면도 혹은 기둥열 계획도등 특정도면들만을 출력시킬 수있게 하였다. 실제의 도면에서와 같이 중심선이나 치수선에 대해서는 가늘게 작도하도록 3 가지 종류의 펜을 사용하고 있다. 매번 공사마다 요구되는 수 십장의 도면을 컴퓨터 그래픽을 통하여 출력하므로 시간의 절약(A0 도면작성시 약 5분)은 물론이며 깨끗하고 정확한 도면을 얻을 수 있다. 그리고 프로그램의 지속적인 개발로 지식을 축적시킬 수 있으며 작업표준화와 대외신용도를 높일 수 있게 한다. 더욱이 BUILDS내에서 해석과 설계와 연계되어 있어 입력자료의 오류를 체크해 볼 수 있다.

### 2.3 I-BUILDS의 설계모델

건물의 설계라는 작업은 공간상에 있는 하중을 비용, 기하학적 형태 또는 설계에 관한 기준등을 만족하면서 기초나 지반에 안전하게 전달시키는 구조물을 창조하는 것이다. 최적설계의 결과는 이러한 하중을 적절한 수준의 안정성과 기능을 만족시키는 정도로 전달하는 능력을 가진 구조물에 대한 세부사항을 마련하는 것이다. 설계는 주어진 조건을 만족시키는 결과를 얻기 위하여 일련의 과정을 반복하는 작업이라 할 수 있다. 이러한 설계의 일반적인 진행과정은 다음과 같다.

1 단계 : 문제 인식의 단계 해당분야의 문제가 제시된다.

2 단계 : 문제의 정의 단계 설계의 목표, 물리-기능적 특성, 가격, 질적요구도, 수행 정도등의 제한조건을 제시한다.

3 단계 : 종합단계 개념화, 문제의 수학적 모델링등으로 자료를 종합한다.

4 단계 : 분석 및 최적화 설계조건에 합당한 지에 대한 공학적인 분석을 통하여 최적의 해를 추구한다. 조건이 위배되면 이전의 단계부터 반복된다.

5 단계 : 평가 설계의 결과에 대하여 최적의 것을 선택하기 위하여 질, 신뢰성, 수행능력등의 평가인자로서 비교, 평가한다.

6 단계 : 결과의 제시 도면화, 자료의 목록등으로 문서화하여 자료를 제시하거나 데이터베이스로 저장한다.

이러한 설계과정은 개인적인 기호에 따르지만 I-BUILDS에서는 위에서 언급한 설계의 과정에 따라 그림 2.4와 같이 5가지의 단계로 나눈다.

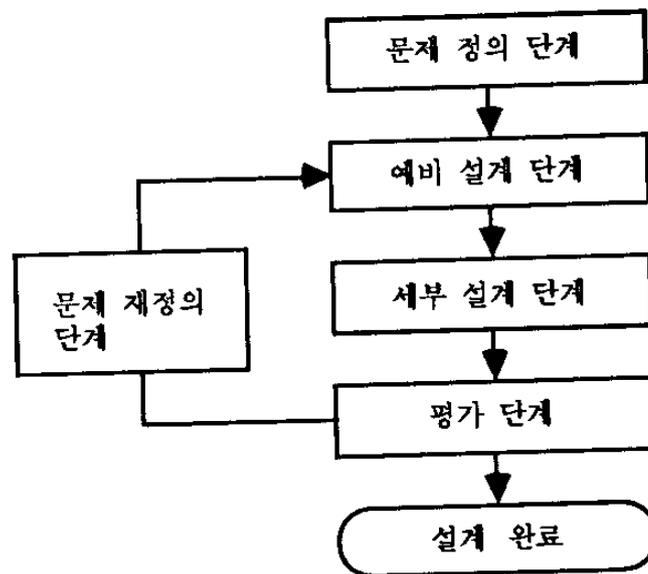


그림 2.4 설계과정의 모델

### 문제의 정의/재정의 단계

문제의 목표 및 해가 가져야 할 조건들을 제시하는 단계로서 앞에서 언급한 1, 2 단계에 해당된다. 건물의 설계는 각각 다른 전문성을 가진 여러사람들 즉, 건축주, 건축가, 구조공학자, 전적가동의 협력으로 이루어지게 된다. 이러한 모든 사람들이 관련하는 건물의 계획은 상당히 복잡한 문제이므로 본 연구의 내용은 주로 구조공학자들이 건물의 설계에 관련하게 되는 제반과정을 다룬다. 즉 문제의 범위는 구조시스템에 관한 설계분야에 국한되며, 그 조건은 요구되는 입력데이터로서 정의된다.

### 예비설계 단계

건물의 형상에 대한 후보들을 제시하고, 그 중에서 가장 적절한 것을 선택하는 것으로 앞에서 언급한 3단계에 해당된다. 이 단계에서의 일은 건물의 설계에 대한 모든 조건들이 종합, 분석되어 가능한 안이 만들어지는데, 설계과정에서 가장 창조적인 과정이다. 이 안들 중에서 설계자의 직감과 판단에 의거하여 가장 좋은 것이 채택된다.

### 세부설계 단계

예비설계에서 선택된 안을 검토해보기 위해서는 이 안을 확대하여 세부적인 사항까지를 결정하여 모든 설계변수에 대하여 고려해 보는 것이 바람직한데, 이는 앞에서 언급한 4단계에 해당된다. 건물설계에 대해 예를들면 선정된 건물의 형상에 대해 초기의 부재성질을 가정하고, 가해지는 설계하중에 대한 건물의 거동 및 구성요소의 부재력을 알아보는 것이 이에 해당된다고 할 수 있다. 이 사항이 만족이 되면 해석의 결과에 따라 부재의 세부사항 (철근의 배치등)이 결정되어진다.

### 평가 단계

앞에서 언급한 과정들이 완성되고 나면 선택된 안에 대한 효율성 및 최적성, 이외의

요구조건등을 기준으로 평가가 이루어지는데, 결과가 만족스러우면 이를 제시하며 끝나게 된다. 이는 앞에서 언급한 5, 6 단계에 해당된다. 만약 평가가 불만족스러우면 예비설계의 단계에서 다른 안에 대하여 검토하던지 또는 문제를 재정의하던지 하여야 한다.

건물의 설계의 과정은 일반적 공학분야에서의 설계과정과 같이 계층적방식으로 진행된다, 이는 개괄적인 개념을 설정하고 차츰 세부적인 사항으로 근접해가는 방법인데, 앞에서 언급한 건물의 설계모델을 더욱 더 세분화시키면 그림 2.5와 같다. 그림 2.5를 보면 예비설계와 평가단계에서는 주로 질적 및 해석적모델에 관한 것이고 세부설계의 단계는 양적 또는 수치적 모델에 관한 것이라 특정지어 질 수 있다. 예를 들면 예비설계에서는 건물의 형상을 결정하는 문제 즉 건물의 구조시스템의 결정, 그 프레임의 평면배치등의 사항이고, 세부설계는 부재의 선택, 크기등의 구조해석, 부재의 최적설계문제에 관한 것들이다. 위에서 언급된 설계모델에 따른 구체적인 설계작업의 진행은 다음과 같다.

#### 문제의 정의 모듈

이 모듈에서는 건물의 기본형상을 결정하는 건물의 높이, 폭, 길이, 그리고 각 방향으로의, 스패의 수와 크기등의 공간배치(space allocation)가 사용자로부터 UNDERSTAND-GIVEN-PROBLEM KU(knowledge unit)로 직접 입력된다.

#### 예비설계 모듈

초기에 주어진 데이터를 바탕으로 건물의 구조의 기본이되는 평면프레임의 구조시스템을 MENU DB에서 선택하고, GENERATE-AND-SELECT KU에서 건물의 2-

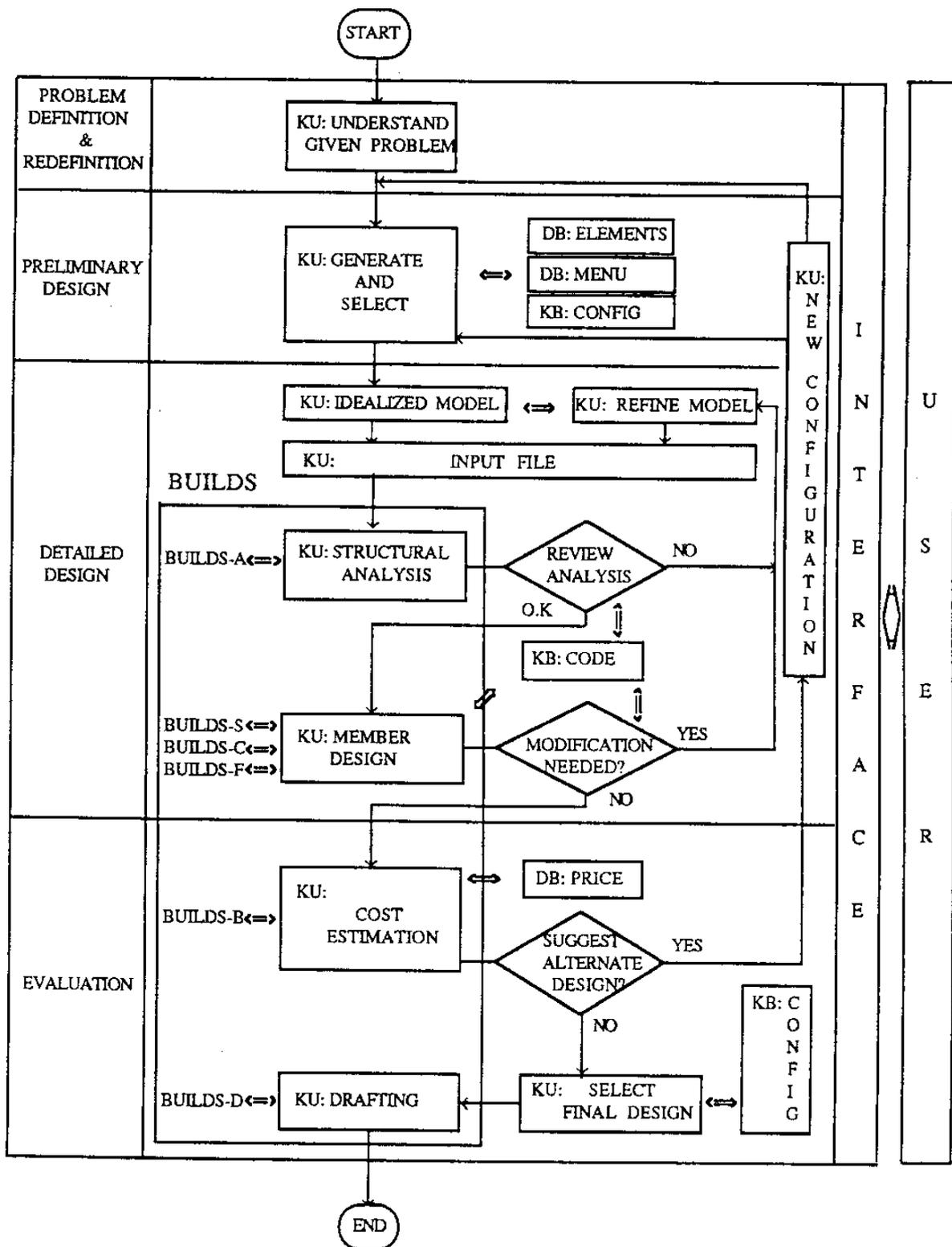


그림 2.5 I-BUILDS의 설계과정

D 프레임의 평면배치에 관한 지식베이스인 CONFIG.KB를 참고하여 3차원적인 형상으로 조합된다. 이러한 지식의 구성은 F. Kahn, ACI Committee 442 [참고문헌 B.26], Monograph on Planning and Design of Tall Building[참고문헌 B.27]과 같은 문헌에서 발췌한 지식을 담은 MENU DB와 CONFIG KB에 의하여 이루어진다. 조합된 건물의 여러 형상들은 평가함수에 의하여 부적절한 것은 버려지고, 나머지 중에서 가장 좋은 것은 선정되고 나머지는 저장되어 다음의 설계과정에서 이용할 수 있도록 한다.

### 세부설계 모듈

여기서는 예비설계의 단계에서 선택된 후보에 대하여 실제로 그 건물을 구성하게 될 부재의 초기성질을 가정하고, 입력파일을 작성하여 체크한 뒤, 구조해석과 부재의 최적설계를 행하고, 그 결과를 검토하여 새로운 부재에 대한 재해석 및 재설계가 이루어지는 단계이다.

Step 1 : 선택된 안으로 설계를 행하기 위하여 부재의 성질등의 설계변수들이 보편적인 스패ん/깊이 비율에 의해 보의 깊이, 슬라브의 두께등을 산정하듯이 경험에 의하거나 portal method와 같은 간단한 알고리즘등에 의하여 IDEALIZED-MODEL KU에서 결정된다.

Step 2 : 이러한 데이터를 바탕으로하여 INPUT-FILE KU는 해석을 위한 입력파일을 자동적으로 작성하고 체크한다. 이때에 처리되는 정보로서, JOB CONTROL 데이터는 사용자로부터 직접 입력되거나 입력된 상황을 감안하여 DEFAULT로 처리된다.

Step 3 : 입력파일이 완전함이 확인된 후, STRUCTURAL-ANALYSIS KU는 해석 프로그램인 BUILDS-A를 loading 하여 구조해석을 수행한 후에 가해진 하중에 대한

건물의 거동을 구한다.

Step 4 : 해석단계에서 구한 건물의 거동과 각 요소의 부재력을 바탕으로 STRUCTURAL-DESIGN KU에 의해 각 과정에 따라 BUILDS-S/C/F를 로딩하여 부재의 최적설계를 행하고, 기초부분의 최적설계를 수행한다.

Step 5 : 설계의 결과로부터 정보를 추출하여 건물시스템에 관한 제한조건(변위 및 전도모멘트등)과 부재에 관한 설계조건(응력및 변형등)을 관련법규 및 시방서등을 내장시킨 CODE KB에 의해 검색한다. 설계조건을 만족시키지 못할 경우에는 REFINE-MODEL KU에 의해 Step 3단계에서 저장되었던 민감도벡터(sensitivity vector)와 반복에 의한 수렴도(convergency) 및 경험에 의하여 새로운 부재를 선정한다.

#### 평가 과정 모듈

선택된 건물의 형상의 설계결과에 대한 견적과 적산을 수행하고, 이 안의 효율성 및 적합성을 평가하여 작업의 진행을 결정한다.

Step 1 : 최적설계로부터 구한 각 부재의 적산과 견적이 COST-ESTIMATION KU에 의해 BUILDS-B를 수행함으로써 구해진다.

Step 2 : SUGGEST-ALTERNATE-CANDIDATE KU는 위의 결과로 얻은 비용과 건물의 기능등의 적합성을 판정하여 예비설계의 단계로 가서 다른 안을 선택하던지, 이미 선택된 안을 수정하던지 또는 설계작업을 마무리할 것인지를 결정하게 된다.

Step 3 : FINAL-DESIGN KU에서는 최종결과를 토대로 하여 건물의 형상 및 구성 부재의 성질등의 데이터를 일반화시켜 CONFIG KB 및 REFINE-MODEL KU에 저장하여 다음 설계작업에서 풍부한 정보를 제공할 수 있도록 한다.

Step 4 : BUILDS의 각 부시스템에서 얻은 결과에 대해 BUILDS-D는 도면화작업을 수행한다.

### 문제의 재정의 모듈

건물의 설계에 대한 방향을 설정하게 될 경우에는 어떠한 설계에 대한 정보가 필요하게 되는데, 이러한 결정을 하는 부분으로서 NEW-CONFIGURATION KU가 기존의 CONFIG KB를 바탕으로하여 현재의 형상을 수정하던지 또는 이전에 저장되었던 다른 안을 제시할 것인지에 관한 일을 하게된다.

### 2.4 I-BUILDS의 구성

기존의 건물설계의 일판시스템인 BUILDS에 앞에서 언급한 건물설계의 모델에 따라 세부적인 BUILDS의 사용을 지원해 주는 I-BUILDS의 통합적 개념은 다음 그림 2.6과 같이 나타낼 수 있다. 즉 FORTRAN 으로 프로그래밍된 기존의 BUILDS시스템과 같은 수치계산을 주로 행하는 부분과 전문가 시스템에 의한 지식처리(심볼처리)를 주로 하는 부분으로 크게 나누어진다.

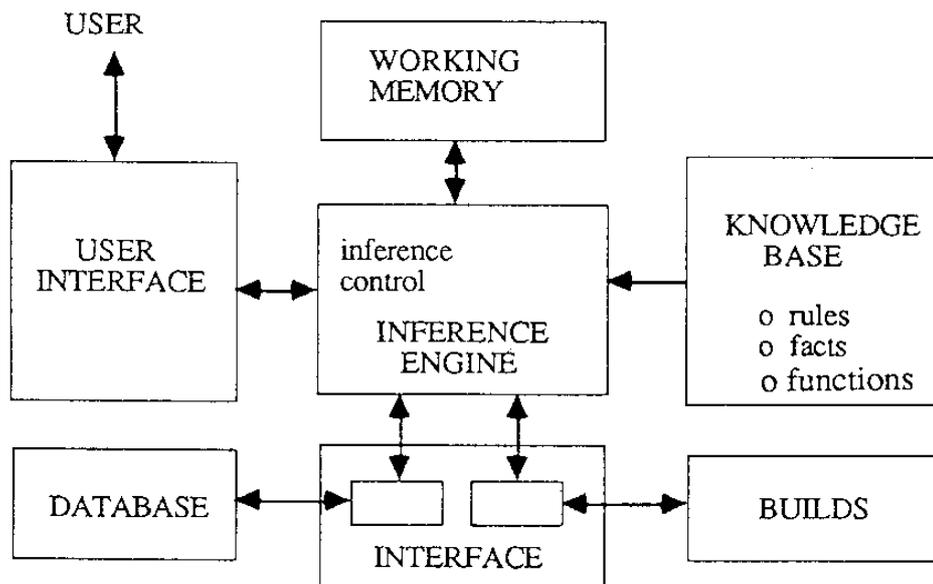


그림 2.6 I-BUILDS의 구조

그림 2.6을 보면 I-BUILDS는 기존의 BUILDS 부시스템으로 구성되어 있고, 전문가 시스템은 CLIPS라는 tool위에서 구성된 지식모듈과 데이터베이스 및 지식베이스로 이루어진다. 지식모듈은 주로 건물의 설계과정에 따라 제반사항을 관리하는 설계순서에 관한 지식이고, 이는 각 작업의 단계에서 각 데이터베이스 및 지식베이스, 그리고 지식단위(knowledge unit)라는 지식의 소뭉음을 다루게 된다. 물론 이러한 설계과정에서는 BUILDS의 각 부시스템들도 각각의 필요한 단계에서 불러져 원하는 결과를 제공하게 된다.

## 2.5 Computer Implementaion

CLIPS는 NASA/Johnson space center의 Artificial intelligence section 에서 개발한 전진추론(forward chaining)을 지닌 production system이다. 이 시스템의 개발 목적은 범용 컴퓨터에서의 LISP의 떨어지는 기능을 보완하고 현재 LISP으로 개발된 도구와 하드웨어의 높은 가격에 비해 보다 싼 가격으로 전문가 시스템용 도구를 개발하고, 또한 타 언어와의 연계성을 강화하기 위한 것이다.

K-CLIPS의 기본적인 구조는 주된 지식베이스와 사용자로 하여금 지식베이스를 조절하도록 해주는 일련의 보조모듈로 구성되며, 지식베이스에 있는 지식은 4개의 부류로 구분할 수 있는데 이들은 다음과 같다.

- o Facts
- o Rules
- o Frames
- o K-CLIPS 구문으로 이루어진 User-defined-functions

그러므로 지식기반 전문가 시스템을 개발하는 과정은 지식베이스내의 4가지 형태로 획득된 지식을 표현하는 것이라 할 수 있다. 사용자는 사실(facts) 및 프레임으로 static한 지식을 표현하고 이를 조절하기 위한 규칙 및 함수를 설정함으로써 특정분야의 문제 해결을 지적으로 해결할 수 있는 프로그램을 작성할 수 있다.

### 3. I-BUILDS의 구축개념

#### 3.1 건물의 설계과정과 탐색

설계(Design)라는 용어는 "구조물의 형태(형상,재료,치수등)를 주어진 조건에 합당하게 결정해나가는 과정"을 일컫는다. 여기서 주어진 조건으로서는 다음과 같은 사항을 들 수 있다.

- 안전성 : 건축물은 자체중량, 적재하중, 적설, 풍압, 토압 및 수압 그리고 지진, 기타 진동 및 충격에 대해 안전하여야 한다.
- 기능성 : 건물의 목적(용도)에 적합하여야 한다.
- 경제성 : 합리적 설계이어야 한다.
- 미적 : 아름다워야 한다.

위와같은 설계조건에 대해 설계를 진행해 나가는 작업의 과정은 초기의 추상적인 데이터로부터 관련된 조건들을 추가하면서, 이러한 조건을 만족하는 상태를 찾아나가는 가는 일련의 탐색과정으로 볼 수 있다. 건물설계에 관련하여 그 예를들면 그림 3.1과 같이 계획단계에서는 건축물에 대한 초기정보가 입력되고, 이를 바탕으로하여 예비설계 단계에서는 구조물의 형상에 대한 후보를 경험에 의존하여 만들고, 각 후보들을 평가 인자별로 평가하여 최적인 안을 선택한다. 여기서 선택된 안을 기존의 수치해석 프로그램으로 세부적인 사항을 검토하며 그 결과를 평가, 제시하게 된다. 이러한 과정에서 다루어지는 데이터를 중심으로 하여 세분해 보면 다음과 같다.

설계단계	내용	탐색
계획단계	초기정보	Initial
예비설계	<ul style="list-style-type: none"> <li>○ 구조형상후보재생</li> <li>○ 구조형상후보평가</li> </ul>	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <p>재료</p> <p>X방향 프레임</p> <p>Y방향 프레임</p> <p>슬래브</p> </div> </div>
세부설계	<ul style="list-style-type: none"> <li>○ 모델링</li> <li>해석</li> <li>부재설계</li> <li>적산</li> </ul>	선택된 안의 확산

그림 3.1 설계단계와 탐색

\* 건축물 구상

건물의 목적(용도), 기준설정이 이루어진다.

\* 구조계획

이 단계에서는 사용자의 요구에 의한 건물의 목적(용도)이 결정되고, 이와더불어 건물의 전체적인 형상, 규모, 재료, 적용될 설계기준등이 언급된다.

\* 예비설계

건축물의 예비형상을 결정하기 위하여 하중을 계산하고, 약산법이나 또는 경험에 의한 초기부재가 가정된다. 이를 위하여 계획단계에서 주어진 사항에 대하여 구조물의 여러 형상들이 고려되는데, 각 대안별로 구성부재의 가능성 및 효율성으로 평가하여 최적의 후보를 제시한다.

\* 세부설계

○ 모델링 : 예비설계에서 선정된 구조물에 대하여 구조해석을 위한 수치적 모델이 구성된다.

○ 해석 : 모델링과정에서 입력된 데이터를 바탕으로 사용될 구조해석 프로그램에 알맞는 입력파일을 구성하고, 구조해석으로부터 건물의 안정성(건물의 변위, 부재의 응력 및 변형등)에 대한 각종 데이터를 구한다. 이러한 결과를 바탕으로 구조적타당성을 검토하고 만약 그 결과가 부적절한 경우에는 부재요소를 개선하기 위하여 모델링을 다시 하던지 또는 예비설계과정에서 얻은 다른 후보를 고려하게 된다.

○ 부재설계 : 구조해석 과정에서 얻어진 각 부재에 대하여 설계기준에 따라 세부단면설계를 행한다.

\* 평 가

현재의 대안이 설계조건에 맞는지, 또는 최적의 상태인지를 판단하기 위하여 여러가지 기준으로 평가를 하게된다.

\* 계산서, 도면, 시방서 작성

모델링, 구조해석, 그리고 부재설계의 과정에서 얻은 데이터를 체계화된 상태로 제시한다.

### 3.2 전문가시스템의 지원영역

기존의 BUILDS시스템을 지원해주기 위하여 전문가 시스템이 갖는 구조, 즉, 기억공간, 규칙, 추론과정을 이용함으로써, 설계과정에서 창조적인 또는 판단에 의존하게 되는 부분에 컴퓨터의 활용을 극대화하게 한다. 전문가시스템에 의한 설계과정의 지원영역을 열거하면 다음과 같다.

1. 전문가 시스템의 기본적인 구조를 이루는 기억공간(working memory)을 데이터교환을 위한 중심체역할로 유도하기 위하여 사실들을 체계화시켜 관계형 데이터베이스의 모형으로 테이블화하고, 데이터의 관리는 Action Menu에 의한다.
2. 초기에 입력되는 건물의 개요에 관한 데이터를 바탕으로 하여 합리적인 구조시스템을 재생하고, 그 평가결과를 사용자에게 제시한다.
3. 건축물의 수치해석의 모델과정은 기억공간내의 사실들을 구축하는 일련의 과정으로 보고, 입력된 이러한 사실들을 기존의 수치해석을 위한 입력파일의 형태로 자동적으로 전환시킨다.

4. 구조해석등의 수치해석 결과는 기억공간에 자동적으로 추가된다.
5. 현재의 기억공간에 있는 사실들에 관하여 법규 및 기타사항의 조건이 있을 경우에는 자동적으로 검토한다.
6. 부재의 단면에 대한 개선이 요구될 때에는 경험 및 민감도 벡터에 의하여 자동적으로 개선된다.
7. 기존의 수치해석 프로그램이 갖는 일방적인 작업의 흐름을 개선하여 사용자의 참여를 최대화하는 인터페이스를 구성한다.

### 3.3 사실의 구성

#### 3.3.1 사실의 구성개념

대형의 전문가 시스템을 구축하기 위해서는 수많은 사실 및 규칙을 구성하게 되는데, 본 연구에서는 이들을 효율적으로 관리하기 위하여 사실을 다음과 같이 구조화한다.

사실은 기억공간에 존재하면서 규칙의 적용가능성에 영향을 주게 된다. 이러한 사실들은 설계의 진행에 따라 추가 및 삭제가 되기 때문에 본 연구에서는 기억공간에 있는 사실의 수를 감소하기 위하여 설계의 전과정을 통하여 항상 존재하게되는 Global한 사실과 특정한 설계과정에 관련한 Local한 사실로 분류하였다. 그리고 K-CLIPS에서는 이러한 사실의 표현을 단순한 symbol list로 나타내는데, 본 연구에서는 관계형 데이터베이스의 개념을 활용하여 하나의 사실을 하나의 레코드(record)로 보고 그 레코드를 구분할 수 있는 자료를 첫번째 열(column)과 두번째 열로 하였다. 즉, 첫번째 열은 관련한 그룹(class)으로, 두번째 열은 그 사실을 구분짓는 고유번호(NO/ID)로 하였고, 나머지는 다른 데이터 항목을 나타내는 부분(field)으로 이용하였다. 즉,

( <클래스> <고유이름/번호> <항목리스트> )

### 3.3.2 Global 사실

#### \* 설계과정 조절용

##### PHASE 사실

기능 : 사용자가 메뉴에서 선택하는 설계과정을 스택으로 보관한다.

예 : (PHASE Grid Planning CONTROL K-CLIPS)

##### MENU 사실

기능 : 사용자가 선택하는 설계과정에 관한 메뉴 트리를 구성한다.

예 : (MENU Planning CONTROL K-CLIPS)

#### \* 구조계획 관련사실

##### Planning 사실

기능 : 계획단계에서 처리되는 데이터 보관한다.

예 : (Planning code WSD)

#### \* 건물구조 묘사 사실

건물을 구성하는 기본적인 구성요소는 구성부재이며, 또한 기존의 유한요소 프로그램에서는 이러한 부재단위로 구조를 묘사하고 있다. 이러한 구성부재를 묘사하기 위한 데이터는 크게 4가지로 구분된다. 즉, 요소의 위치를 나타내는 좌표계, 재료의 성질, 단면의 성질, 그리고 요소하중들이다. 구조요소는 이러한 4가지의 데이터로 완벽하게 묘사할 수 있는데, 보통은 데이터의 중복을 피하기 위해 4 개의 그룹으로 분류하여 요

소의 묘사를 이리한 각 그룹에 대한 지칭(indexing)으로 한다. 본 연구에서는 부재의 묘사를 이와같이 하되 잘못된 지칭에 대해서는 즉각 통고하도록 하여 오류의 데이터가 기억공간에 존재하는 것을 방지하여 해석의 잘못된 결과를 미연에 방지하도록 하였다. 여기에 관련한 내용은 제 5장의 구조물의 이상화중에서 5.3 모델의 구축부분을 참고하길바란다.

### 3.3.3 Local 사실

#### \* 설계과정 조절용

##### o FILE 사실

기능 : File에 대한 정보

표현 : (FILE <File name> <input or output>)

예 : (FILE "test.dat" input)

##### o 기타(DATA/TEMP등)

#### \* 설계조건 사실

##### o CONSTRAINT 사실

기능 : 설계조건에 대한 사항

(1) 데이터 타입 - integer, real, word, string

(2) 데이터 조건 - EQ, NEQ, GT, LT, LE, RANGE, AMONG, SET

예 : (CONSTRAINT Planning code CONTENT word AMONG WSD USD)

### 3.3.4 메뉴 트리 구조

건물의 설계단계에 따라 구성된 메뉴는 다음 그림과 같다. 그림의 메뉴에서 대문자

로만 된 메뉴는 그 이하의 서브메뉴가 있음을 알리고, 대문자와 소문자의 복합으로 표현된 메뉴는 해당메뉴가 사실의 Class로서 이용됨을 알린다. 그리고 소문자로만 된 메뉴는 데이터를 구축하기 위한 메뉴이거나 작업을 실행하는 메뉴, 즉, 행위메뉴(Action Menu)임을 알린다. 여기서 행위메뉴는 사용자를 위한 메뉴자체가 어떠한 행위를 유발하는 것으로 소문자로 되어있다. 예를들면 list, insert, delete, modify, save, load 등이다. 다음 그림 3.2는 메뉴트리의 일부분은 보여준다.

그림 3.2를 보면 메뉴트리의 말단부분은 데이터의 관리를 위한 명령어로 되어 있음을 알 수 있다. 이를 본 연구에서는 Action메뉴라 칭하며 그 작업내용은 다음과 같다.

- o help : 해당설계단계에 관하여 설명한다.
- o list : 현재의 데이터부류에 관련된 정보들을 제시한다.
- o insert : 현재의 메모리에 사용자가 정의하는 사항을 입력한다.
- o delete : 사용자가 정의하는 사항을 메모리에서 제거한다.
- o modify : 데이터의 수정을 한다.
- o load : 정보를 지시하는 파일로부터 읽어들인다.
- o save : 지시하는 파일로 정보를 저장한다.
- o check : 데이터의 형식 및 값에 대하여 검토한다.
- o 기타

### 3.4 규칙의 구성

#### 3.4.1 규칙의 구성개념

3.3절에서 언급한 MENU사실의 트리구조와 PHASE사실의 스택구조로 사용자는

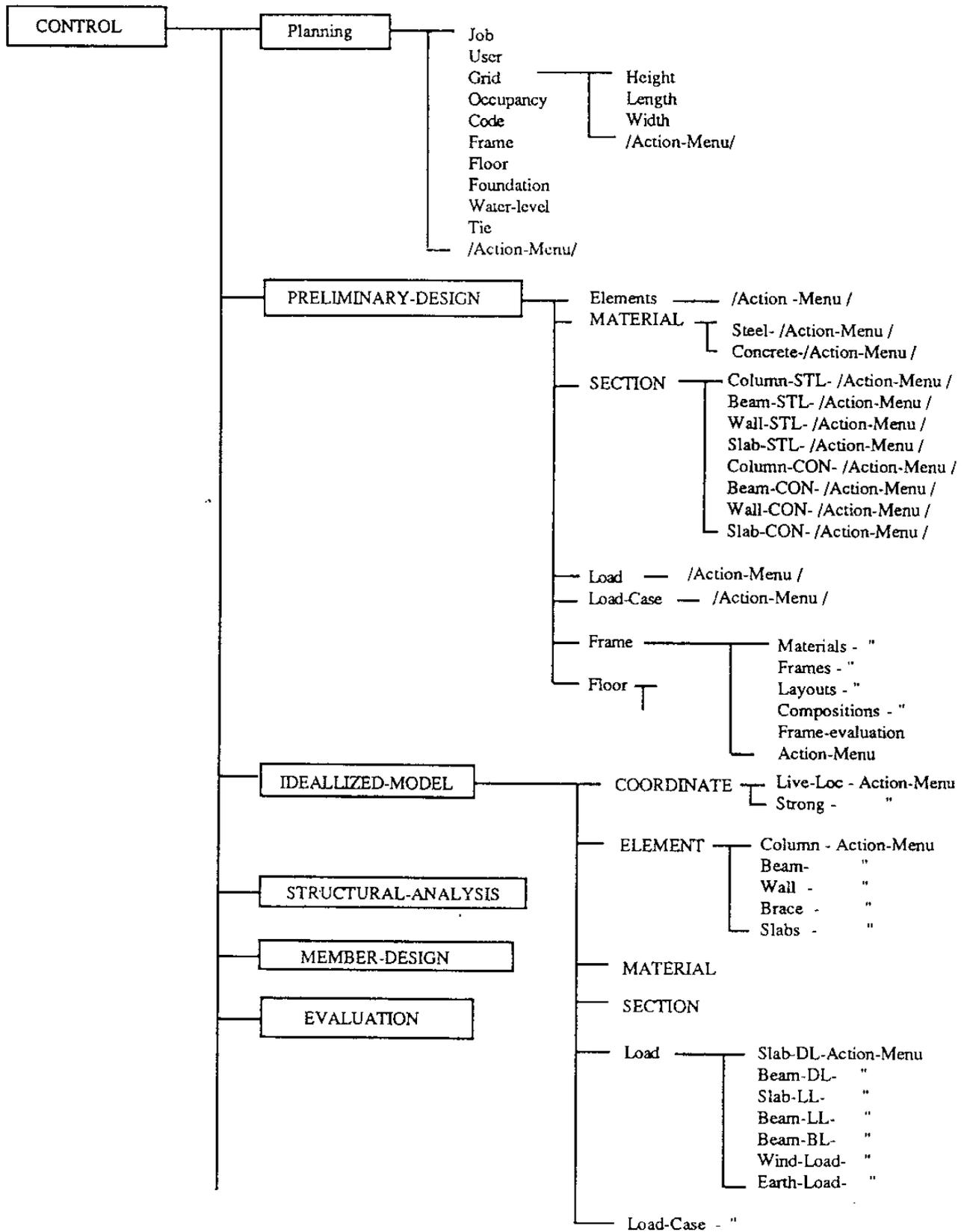


그림 3.2 I-BUILDS의 메뉴구조

어느 설계단계이든지 원하는 것을 선택할 수 있다. 이러한 설계단계의 변화에 대하여 적용할 수 있는 규칙은 크게 두 가지로 대별할 수 있다. 즉, 어느 순간이든지 적용할 수 있는 global한 것과 특정설계단계에서만 적용할 수 있는 local한 것이다. 또한 기억 공간에 존재하는 사실에 대하여 적용가능한 규칙에 여러개 있을 경우에는 규칙의 성격상 먼저 작용하도록 유도할 필요가 있는 경우가 있다. 예를들면 어떤 사실에 대하여 조건이 부가된다면 그 사실로 어떤 일을 하기 전에 조건을 먼저 검토해 보는 과정이 필요하다. 이러한 연유로

본 연구에서는 K-CLIPS가 제공하는 우선권의 개념을 이용하여 특정설계단계에서 적용될 규칙을 몇개의 부류로 분류하였다. 여기서 우선권은 각 규칙을 구성하는 시기에 각 규칙에 부여할 수 있는 수치(-10000 - 10000)인데, 높을수록 먼저 작용하게 된다. 본 연구에서 사용한 우선권의 개념은 다음과 같다.

표 3.1 규칙의 우선권 이용

우선권 (Priority)	기능
10000	안내부
5000	조건처리부
3000	입,출력부
0	작업명령처리부
-5000	결과검토부
-7000	결과제시부
-10000	설계단계제시부

### 3.4.2 Global 규칙

이 규칙은 설계단계의 진행에 따라 언제든지 작용할 수 있는 것인데 이들은 우선권의 개념으로 다음과 같은 작업 모델을 구성한다.

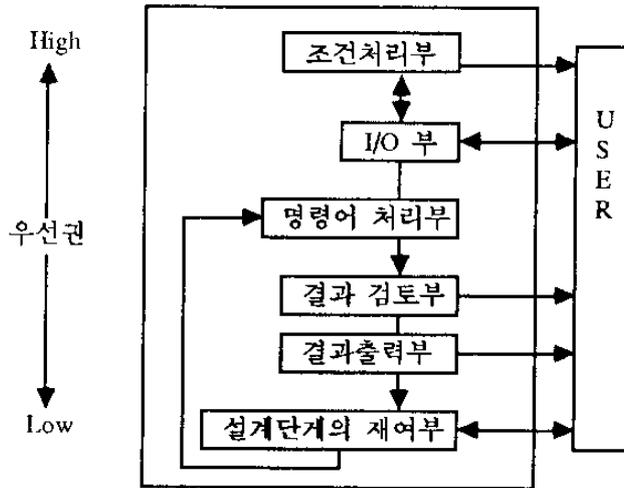


그림 3.3 특정 설계공간내의 작업모델

(a) 조건처리부

기능 :

- (1) CONSTRAINT 사실 처리
- (2) 현재의 설계단계에 불필요한 사실 (DATA, MENU, CONSTRAINT, HELP등)의 제거
- (3) File 이름 입력 및 File Open
- (4) 불필요한 사실의 제거

(b) 입,출력부

기능 :

- (1) Action Menu의 입,출력처리

(c) 명령처리부

기능 : Action Menu (help,list,insert,delete,modify,load,save, grid+ 등) 처리

(d) 결과검토부

기능 :

(1) Action Menu의 입력결과 검토

(2) File close

(e) 결과출력부

기능 :

(1) 결과출력

(2) 불필요한 사실의 제거

(f) 설계단계의 제시부

기능 : 다음 설계단계의 제시 및 선택

### 3.4.3 Local 규칙

각 설계단계에 국한된 규칙을 말하는데, 대부분의 경우는 Global한 규칙이 적용되어 사실을 처리하므로 Local규칙에서는 주로 해당설계단계의 MENU, CONSTRAINT, HELP등의 사실을 추가하게 된다.

## 3.5 부재의 설계와 패턴매칭

### 3.5.1 부재단면의 선택과 패턴매칭

기존의 수치해석프로그램에서는 부재의 설계에 대한 최적인 해를 구하기 위하여 수학적 프로그램에 의존한 복잡한 알고리즘을 이용하기 때문에 많은 시간과 노력을 요하였다. 이러한 노력에도 불구하고 아직까지 실용화를 위한 최적설계프로그램은 별로 없는 실정이다. 본 연구에서는 이러한 어려움을 전문가시스템의 추론과정에 의한 패턴매칭기능을 활용함으로써, 주어지는 몇가지 조건을 만족하는 부재를 추론과정에서 쉽게 선택할 수 있게하였다. 단지 이러한 설계를 위해서는 기존단면의 데이터가 현재의 기억공간에 존재하여야 하기 때문에 본 연구에서는 아래와 같은 부재의 단면데이터베이스를 이용하였다.

### 3.5.2 부재의 데이터베이스

#### \*건물재료가 철골인 경우

건축구조물의 재료가 철골인 경우에는 현재 우리나라에서 생산되어 사용되고 있는 제품을 데이터베이스로 하여 그 제품의 성질(면적, 깊이, 폭, 두께, 기타 단면성질등)을 정리하여 고유번호만으로 호출하여 사용할 수 있다. 본 연구에서는 기존의 형강제품을 제품단면의 특성상 기둥에 많이 사용되는 부류와 보에 많이 사용되는 부류로 대별하여 기둥인 경우에는 45개, 보는 49개의 데이터가 표 3.2의 구조로 저장되어 있다 [참고문헌 B.15].

표 3.2 철골부재의 변수

기둥	: No.	W	D	B	Tw	Tf	A	Ix	Iy	Jx	Jy	Zx	Zy
보	: No.	W	D	B	Tw	Tf	A	Ix	Iy	Jx	Jy	Zx	Zy

여기서 No.는 고유번호, D는 깊이, B는 폭, Tw와 Tf는 웹 및 플랜지의 두께, A는 면적, Ix, Iy, Jx, Jy, Zx, Zy는 x, y축에 대한 단면 2차모멘트, 반경 그리고 단면계수를 의미한다.

\* 건물재료가 철근콘크리트인 경우

건축구조물의 재료가 철근콘크리트인 경우는 크게 두가지 형태로 부재를 설계할 수 있다. 첫째는 기존의 건물에서 사용된 콘크리트부재의 단면을 분석하여 철골제품마냥 그 성질을 데이터 베이스로 저장된 것을 사용하는 것과, 둘째는 데이터베이스를 사용하지 않고 주어진 부재력에 대하여 일반적인 단면설계를 행하는 방법이다.

본 연구에서는 사용자가 원하는 철근의 번호를 입력하면 그 철근을 이용한 단면의 데이터베이스를 로딩하여 계산된 부재력이 데이터베이스의 최대부재력을 초과할 경우에는 허용응력법에 의하여 부재를 설계하도록 하였고, 데이터베이스내에 있을 경우에는 가능한 부재중에서 비용이 최소인 부재를 선택하도록 하였다. 데이터베이스의 구성은 기둥에 대해서는 D19, D22, D25인 각각의 경우에 대한 약350개의 단면성질과, 보에 대해서는 D19, D22의 각각 약 250개인 경우에 대하여 구성되어 있다. 각 요소의 사실(facts)은 표 3.3의 구조로 구성된다[참고문헌 B.16].

표 3.3 철근콘크리트 부재의 변수

기둥	: No.	B	H	BN	HN	CN	Agro	Ares	Zxgro	Zygro	Zxeff	Zyeff
보	: No.	B	H	TN	CN	Zeff						

여기서 No.는 고유번호, B는 단면의 폭, H는 단면의 깊이, BN은 단면의 폭 방향의 철근수, HN은 깊이 방향의 철근수, CN은 단면 모서리의 철근수 (4), Agro는 전체 환산단면적, Ares는 등가단면적( =  $0.3A_g + 0.4(F_y/F_c)A_{st}$ ), Zxgro 및 Zygro는 X, Y방향의 전체 단면계수, Zxeff 와 Zyeff는 X, Y방향의 유효단면계수, TN, CN은 인장 및 압축측의 철근수를 의미한다.

## 4. 예비설계를 위한 전문가시스템

### 4.1 기초사항

#### 4.1.1 예비설계의 목적

구조설계는 계획된 건축물에 골조를 배치하고, 부재요소의 크기, 세부단면을 결정하는 것인데, 이러한 일련의 과정에서는 건물의 초기계획을 하는 예비설계는 설계과정상 가장 중요한 단계이다. 예비설계는 여러사람에 의하여 설계가 진행되거나, 구조체의 골격이 다른 설계 팀에 영향을 주게 되는 경우에 최종설계가 정상적인 과정으로 이루어지도록 하는데 중요한 역할을 한다. 이러한 예비설계의 옳고 그름은 건물의 완성에 큰 영향을 주게 되므로 신중히 검토하여야 한다.

예비설계를 행하는 목적은 다음과 같다.

1. 개개 요소의 배치를 결정하게 되는 구조체의 골격을 결정한다. 구조적으로 주된 부재를 일정한 기준으로 위치시킨다.
2. 다른 설계팀의 일에 영향을 주게 되는 주요한 구조부재에 대해서는 크기에 대한 개요를 설정한다. 부차적인 부재들은 구조체의 골격에는 영향을 주지 않으므로 예비설계의 단계에서는 가끔 그 크기를 설정할 필요가 없다.
3. 구조체의 주요한 특성 즉, 다른 설계팀이 유의하여야 할 사항을 검토하여 제시한다. 이는 다른부서의 설계자들이 설계변경이 불가능한 최소한의 구조적인 특성을 제시함을 말한다.
4. 요구되는 재료의 양을 측정하여 건설비를 예측할 경우의 준비자료를 제공한다.

즉, 본 연구에서의 예비설계의 목적은 건축공간을 형성하는 실체로서 구체화하게 되는 설계의 초기단계로서, 기본적으로 구조재료의 역학적 성질, 구조시스템이 갖는 구조특성, 안정성 및 경제성등의 구조기술상의 재반적인 문제에 대해 기존의 전문지식을 컴퓨터에 내장하여 구조분야뿐만 아니라 타 영역의 설계진행에 대해서도 적절한 도움을 주기위한 것이다.

#### 4.1.2 예비설계의 진행계획

공학문제는 일반적으로 단계적인 절차를 통하여 해를 구하게 된다. 각 단계에서는 처리해야 할 일들이 수행되며, 이러한 단계가 순서대로 진행됨에 따라 필요없는 정보는 버려지고 결국에는 문제의 해를 구하게 된다. 이와 비슷하게 인공지능분야에서도 문제해결의 모델이 개발되었는데, 본 연구에서 고려하는 문제풀이전략에 대해서는 다음과 같은 항목을 들 수 있다.

##### (1) 후보제시와 검토(Generate and Test)

이 방식은 우선 가능성이 있는 해를 가정하고, 이것이 해로서 가능한 지에 대한 확신을 가지기 위하여 문제에 대한 조건을 적용하여 가정을 검증해 내는 설계방법이다. 공학문제에서 해석과 설계는 이러한 방식을 광범위하게 사용하고 있다. 예를들면 설계의 초기에 변수를 가정하고 해석의 결과가 규준에 합당한지를 검토하는 것을 들 수 있다. 만약 검토단계에서 만족되지 않으면 초기의 가정을 수정, 보완하고 다시 해를 구하는 과정이 반복된다. 이러한 방법에서 가장 중요한 부분은 해를 가정하는데 필요한 지식이다. 이 부분의 지능을 최대화함으로써 초기가정을 갱신하기 위한 과정을 최소화할 수 있다. 이러한 예로 본 연구에서는 건물의 예비설계 단계에서 추천된, 가능한 구조시스템에서 가장 적절한 것을 고르는 것을 들 수 있다.

(2) 계층적 접근(Hierarchical approach)

대부분의 전문가들이 그들의 문제해결과정에서 활용하는 방법은 계층적인 접근이다. 즉, 전체적인 개념(예를들면 건물의 평면계획)을 결정하고 점차 세부적인 사항(예를들면 구조요소의 설계)을 생각하게 된다. I-BUILDS는 이러한 전문가의 접근방식을 이용하게 되는데, 설계의 진전에 따라 표 4.1에서처럼 단계적으로 세부사항에 근접해간다. 표 4.1을 보면 I-BUILDS에서 다루는 데이터의 전체적인 흐름을 볼 수 있는데, 설계단계가 구체화됨에 따라 건물을 구성하는 각 요소의 부재설계로 정보가 흐르는 것을 볼 수 있다. 이러한 단계에 사용되는 지식의 부류는 예비설계단계의 이상은 주로 경험적 자료에 의존하고 세부설계분야는 정해진 알고리즘에 따라 결정됨을 알 수 있다.

표 4.1 데이터의 관리방법

자료의 분류	정보의 레벨	설계단계	자료처리
실행제어데이터 건물형상데이터	-	설계작업조절	입력
	TOP	건물계획단계	입력
BLDG SYSTEM			
3 - D			
구조요소	2 - D	예비설계단계	경험지식
	ELEMENTS	세부설계단계	경험지식과 알고리즘 (BUILDS등)
재료성질			입력 + 기준

### (3) 문제분할방식(Problem reduction method)

문제분할방식은 전체문제를 일련의 작은 문제로 줄이고 각각의 작은 문제에 대하여 목표를 선정하여 해결을 구하는 방식으로 전체문제를 풀어나가는 방식이다. 제어는 작은 한 문제로부터 다른 문제로 문제의 해결정도, 상대적인 우선 순위에 따라 넘겨지게 된다. 이러한 전략은 대형의 문제에서 필수적인 것인데 특히 문제풀이과정이 확실하여 뚜렷한 목표가 선정될 수 있을 때 유리하다. 본 연구에서는 건물을 크게 수직력 저항 시스템과 수평력 저항시스템으로 나누고, 여기서 수직력 저항시스템은 슬라브 시스템으로 가정하며, 수평력 저항시스템은 건물의 폭방향과 길이방향의 2차원 프레임으로 교차된 조합체로 모델링한다. 설계의 진행과정은 수직력 저항시스템을 결정하고, 수평력저항 시스템에 대해서는 폭방향을 먼저 결정하고 이와 어울릴 수 있는 길이방향의 시스템을 결정한다. 그리고 각각의 시스템을 결정하는 과정은 세부적인 내 단계, 즉 종합단계, 분석단계, 평가단계 그리고 선택단계로 진행된다.

### (4) 조건처리기법

조건처리에 의한 설계기법은 새로운 조건을 추가하는 조건의 형성단계와 이러한 조건에 맞는 값을 구하는 조건의 평가단계로 구분된다. 이러한 개념으로 (3)항에서 언급한 설계단계를 분석하면 다음과 같다. 종합단계에서 가능한 조건에 의해 후보가 제시되면 이러한 후보중에서 최적절한 것을 찾기위하여, 각 후보에 대하여 구조요소, 전도모멘트, 평가완료등의 조건이 분석단계에서 형성되는데, 이 단계에서는 전도모멘트등의 구조시스템에 대한 전반적인 조건과 구조요소의 부재력에 대한 조건이 검토되고 만족이 되면, 변수의 설정단계에서 부재력에 대한 재료 및 단면설정이 이루어진다. 평가완료에 대한 조건은 평가단계에서 사용되는데, 이 때는 평가를 위한 새로운 평가요인이 형성되어 후보의 순위를 부여하기위한 자료로 이용된다.

#### (5) 깊이우선탐색전략

(4)항의 조건처리 기법에 의한 과정에서, 종합과정을 통하여 제시된 가능한 구조시스템에 대한 가능성을 검토해보기 위하여 분석-부재설계-평가의 단계를 깊이 우선의 전략으로 특정 후보에 대하여 모든 사항을 검토해보고 이러한 과정중에서 불만족한 사항이 나오면 이 후보를 제거하며, 최종의 단계까지 이르면 평가치를 부여하는 작업을 각각의 후보별로 반복한다. 이러한 이유는 각각의 후보에 대하여 구조적 가능성을 검토해 보는 과정에서 발생하게 되는 사실(facts)들을 후보의 평가가 끝나는 시점에서 제거시킬 수 있기 때문에 사실의 확장으로 인한 작동기억공간(working memory)을 효율적으로 사용할 수 있기 때문이다.

#### 4.1.3 연구의 범위

현재 본 연구의 범위는 건물의 형태가 사각형이고, 용도는 상업용 또는 주거용일 경우로서 층수는 중층(10층) 이상의 고층에 한한다. 건물의 재료는 철골인 경우와 철골구조인 경우로 하였다. 이러한 내용으로 사용자가 입력하게 되는 자료는 건물설계의 초기단계에서 나타나는 건물의 그리드(grid)로서 건물의 폭 및 길이방향의 베이수와 그의 크기, 코어위치, 층수, 층고등이 있으며 건물의 개요에 대한 것으로 건물의 용도, 구조등이 있다. 출력으로는 사용자의 입력에 대한 가능한 구조시스템을 선택하고 BUILDS의 부시스템을 설계과정에 따라 제어하는 모든 정보이다. 이렇게 함으로써 I-BUILDS는 구조설계자가 하는 일 즉, 건축주 및 건축가가 요구하는 설계조건에 접합시키면서 가능성 높은 기능성, 안정성 그리고 시공성등을 구비하는 구조체를 창조해 내는 행위를 모방할 수 있는 것이다.

(1) 건물의 개요

- 건물의 용도 : 주거용 또는 상업용
- 건물의 구조 : 상부구조 - 철골구조 또는 철근콘크리트 하부구조 - 연구중
- 건물의 규모 : 층수, 건물고, 건물의 폭과 길이, 두방향의 베이수, 코어의 유무

(2) 설계기준

- 부재의 설계방법 : 허용응력법에 의함
- 설계 기준 :
  - 1) 건축물의 구조기준등에 관한 규칙
  - 2) 철근 콘크리트 구조계산기준
  - 3) 강구조 계산기준

(3) 구조재의 성질

- 철골구조 : 항복강도( $F_y$ )를 입력
- 철근콘크리트 구조 :
  - 1) 철근 : 항복강도( $F_y$ )를 입력
  - 2) 콘크리트 : 설계기준강도( $F_c$ )를 입력

(4) 지반의 성질

- 허용지내력
- 지하수위
- 내부마찰각

## 4.2 예비설계의 모델

본 연구에서는 예비설계단계를 크게 그림 4.1과 같이 4단계로 구분하여, 설계를 진행하는데, 종합단계에서는 주어진 데이터에 의해 후보를 재생하고 이러한 후보중에서 최적적인 것을 찾기위하여 분석단계에서는 각 후보에 대한 구조요소, 전도모멘트등의 조건이 형성되어 부재력을 구하고, 전도모멘트조건을 검토한다. 이 조건이 만족이 되면, 각 부재에 대하여 현재의 부재력에 대한 단면산정 및 설계가 이루어진다. 평가단계에서는 평가를 위한 새로운 평가요인들이 형성되어 후보의 순위를 부여한다.

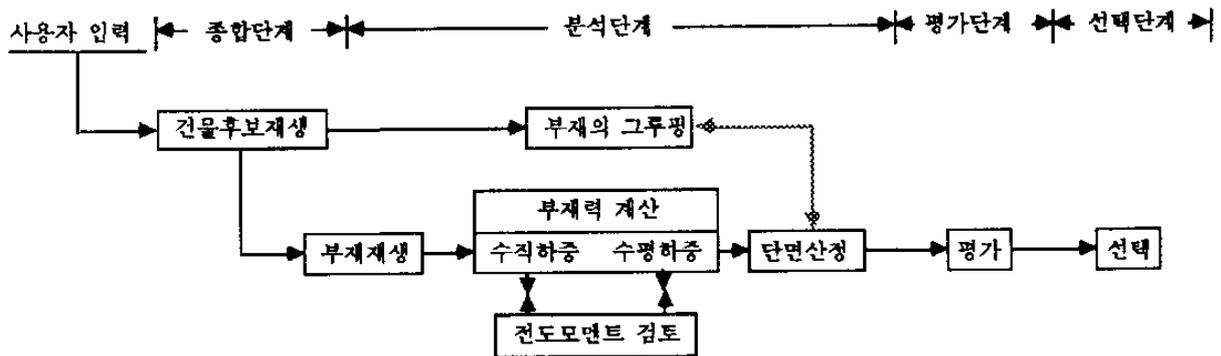


그림 4.1 예비설계의 모델

(1) 종합단계 : 구조시스템에 대한 후보를 구성하기 위하여 모든 가능성으로 후보를 검토해보고, 부적절한 것은 제거규칙에 의하여 미리 후보대상에서 배제시킴으로서 불필요한 검토노력을 방지한다.

(2) 분석단계 : 이 단계부터는 종합단계에서 재생된 후보의 가능성을 검토해보기 위하

여 분석조건을 형성하고, 그 만족여부를 평가하게 된다. 본 단계에서는 현재의 후보에 대하여, 구성하는 부재의 단면성질이 동일함을 나타내는 부재의 그룹과 그 후보를 실제로 구성하게 되는 모든 부재를 재생하고, 각 부재에 대하여 중력과 수평력에 대한 부재력을 구하고, 전도모멘트를 검토한다. 그리고 부재력이 구해진 부재에 대해서는 설계변수를 설정하여 부재설계의 가능성을 검토한다. 만약 제한된 조건(예, 단면크기) 하에서 지속적인 단면성질의 개선에도 불구하고 불만족하게 될 경우에는 구조시스템이 불가능한 것으로 본다.

(3) 평가단계 : 구조시스템의 평가는 여러가지 평가요인을 가지게 되는데, 이러한 요인으로서는 경제성, 공간적 효율성 및 구조적 적합성등을 들 수 있다. 선형의 평가함수가 본 연구에서는 이용된다.

(4) 선택단계 : 구조적으로 가능한 모든 시스템의 평가치를 정리하여 사용자에게 정보를 제공해주며, 사용자에게 선택을 요구하게 된다.

### 4.3 예비설계의 지식

#### 4.3.1 종합단계

종합단계에서는 구조적인 가능성이 있는 건물의 후보가 제시된다. 이를 위해서는 크게 두 단계로 설계가 진행되는데, 첫째는 계획 단계로서 건물을 나타내는 각 계층별 후보를 재생하는 것이고, 둘째는 건물후보를 재생하기 위하여 각 계층별 후보를 조합하는 것이다. 두번째 단계에서 생겨나는 건물의 후보중에서 경험적 판단으로 부적절하다고 여겨지는 후보에 대해서는 미리 후보 대상에서 제거한다.

#### (1) 각 계층별 후보 재생

건물을 두 방향의 프레임의 조합체로 모델링 할 때, 건물을 묘사하는 일반적인 방법은 건물의 재료, 슬라브의 종류, 두 방향의 프레임의 종류, 배치방법, 구성방법등을 들 수 있다. 본 연구에서는 이러한 방법으로 건물의 후보를 재생하기 위하여 건물을 체계적으로 계층분류하여 다음과 같이 각 계층별 후보를 재생하고, 이들을 가능성 있는 실제의 건물후보로 조합하였다. 각 계층의 순서는 일반적인 설계의 순서에 준하여 건물의 재료, 수직력저항시스템으로서의 층 프레임(Floor), 그리고 두 방향의 수평력저항시스템으로 진행된다. 여기서 수평력저항시스템은 각각의 종류, 배치방법, 구성방법에 의하여 세분된다. 각 계층별 후보는 다음과 같다.

#### 건물의 재료계층

철골(steel)과 철근콘크리트(RC)재료가 가능하다.

#### 슬라브 종류계층

현재로서는 RC-slab만이 가능하다.

#### 폭방향의 프레임 종류계층

프레임의 종류로는 보,기둥만으로 이루어진 모멘트저항 프레임이외에, 트러스로 전단보강된 브레이스보강 프레임과 전단벽으로 보강된 벽체보강 프레임이 있다. 이러한 프레임의 배치 및 전단보강되는 스패의 위치는 아래 두 계층에 따라 정해진다.

#### 폭방향의 프레임의 배치계층

사용자가 입력한 폭방향 그리드(X) 상에 전단보강된 프레임이 놓일 수 있는 그리드를 말한다. 여기서 전단보강되지 않는 위치의 프레임은 모멘트저항골조가 놓이는 것으로 가정한다. 이를 위하여 기존의 건물의 전단보강 패턴을 분석하여, 그 종류를 분류해본 결과, 다음과 같이 크게 5가지로 분류하였다. 이러한 배치형태에 대한 기본적인

가정은 프레임의 한쪽을 치우치게 보강하여 비틀림을 받는 것을 피하기 위하여 내 풍 및 내진효과를 효율적으로 하기 위한 것이다.

○ 배치 1(none) : 전단보강 하지 않음 폭방향 전 그리드상에 전단보강을 하지 않는다.

○ 배치 2(ext.) : 외부측으로의 전단보강 수평하중을 외부에 위치하는 프레임으로 저항하게 설계한 경우로서 전단보강을 건물의 외부양측으로 배치하며, 사용자가 입력하는 그리드수에 따라 여러 종류가 생성된다. 예를들면 다음 그림 4.2와 같다. 여기서 (a)인 경우는 해당방향의 프레임이 위치 가능한 그리드의 수가 홀수인 경우이고, (b)는 짝수인 경우이다.

○ 배치 3(all) : 그리드 전부에 전단보강 사용자가 제시하는 그리드에 해당방향으로 프레임을 전부배치시키는 경우로서 배치1의 역으로 생성된다.

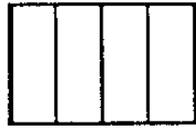
○ 배치 4(int.) : 내부측으로의 전단보강 내부측으로 전단보강을 하는 것으로 배치 2의 역으로 생성된다. 단, 여기서 하나의 그리드만 전단보강되는 경우는 드물기 때문에 제거하게 된다.

○ 배치 5(user) : 코어를 원하는 위치에 배치

사용자가 입력한 데이터로서 코어가 있는 경우에는 코어의 주변을 지나게 되는 그리드에 프레임을 배치하는 방법이다.

사용자가 입력하는 그리드수가 6X5이고, 원하는 코어의 위치가 폭방향으로 향하는 2,3의 X그리드, 길이방향으로 1,2,3의 Y그리드(그림 4.5참조)일때 배치에는 표 4.2 (a)와 같다.

방법 1 : ext-1

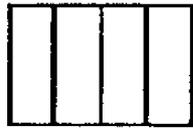


(a)



(b)

방법 2 : ext-2



(a)



(b)

방법 3 : ext-3 - int-3의 역



(a)

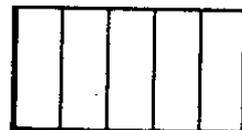


(b)

방법 4 : ext-4 - int-4의 역



(a)



(b)

그림 4.2 배치 2



(a)



(b)

그림 4.3 배치 3

방법 1 : int-1



(a)



(b)

방법 2 : int-2

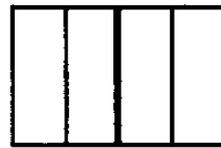


(a)



(b)

방법 3 : int-3 - ext-3의 역



(a)



(b)

방법 4 : int-4 - ext-4의 역



(a)



(b)

그림 4.4 배치 4

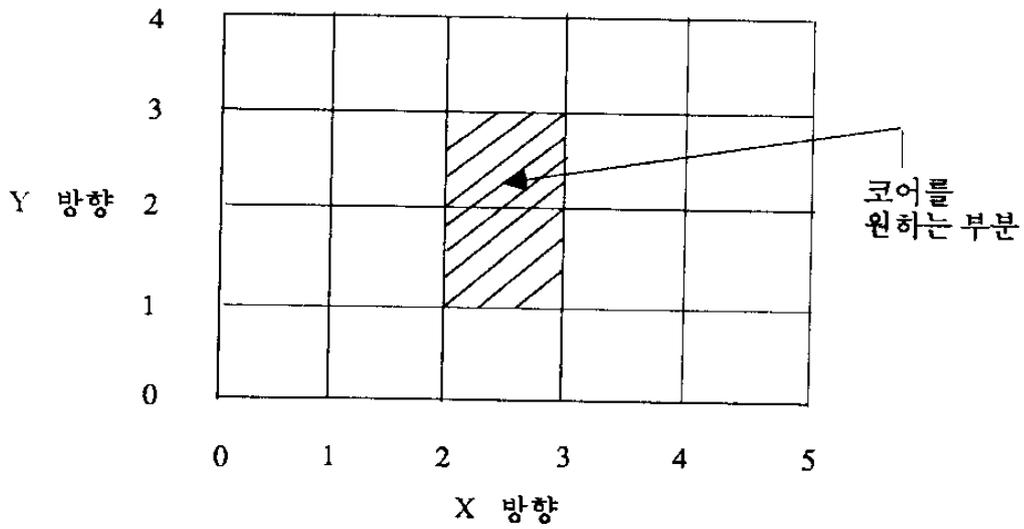


그림 4.5 그리드 번호

표 4.2 폭 방향 프레임의 배치 및 구성

○ : 배치된 그리드

방법 \ 그리드 (X)	0	1	2	3	4	5
배치 1	-	-	-	-	-	-
배치 2	0	-	-	-	-	0
배치 3	0	0	-	-	0	0
배치 4	0	0	0	0	0	0
배치 5	-	0	0	0	0	-
	-	-	0	0	-	-
	-	-	0	0	-	-

(a) 전단보강 프레임의 배치

○ : 보강스팬

방법 \ 그리드 (Y)	0-1	1-2	2-3	3-4
구성 1	-	-	-	-
구성 2	0	-	-	0
구성 3	0	0	0	0
구성 4	-	0	0	-
구성 5	-	0	0	-

(b) 전단보강 스패 구성

### 폭방향의 프레임의 스펠구성계층

프레임의 내부 스펠에 대하여 전단보강위치를 나타내는 것으로 그 종류는 배치종류처럼 크게 5가지로 나누어지는데, 외부측 및 내부측 보강에 따른 스펠보강 크기는 길이방향의 그리드수에 좌우된다. 구성에는 표 4.2(b)와 같다.

- 구성 1(none) : 전 스펠을 전단보강하지 않는다.
- 구성 2(ext.) : 전단보강을 프레임의 외부스펠으로 하는 것으로 사용자의 그리드 입력에 따라 여러 종류가 재생된다.
- 구성 3(all ) : 전 스펠을 전단보강하는 것으로 구성 1의 역으로 재생된다.
- 구성 4(int.) : 내부측으로 전단보강을 하는 것으로 배치 2의 역으로 재생된다.
- 구성 5(user) : 사용자가 원하는 그리드 사이에 전단보강을 한다.

### 길이방향의 프레임 종류계층

폭 방향의 종류와 같다.

### 길이방향의 프레임의 배치계층

방법은 폭 방향과 같으며 외부,내부측 보강의 종류는 길이 방향의 그리드 수에 따르며 그 예는 표 4.3(a)과 같다.

### 길이방향의 프레임의 구성계층

방법은 폭방향과 같으며 스펠보강크기는 폭방향의 그리드 수에 따르며 예는 표 4.3(b)과 같다.

표 4.3 길이방향 프레임의 배치 및 구성

○ : 배치된 그리드

방법 \ 그리드 (Y)	0	1	2	3	4
배치 1	-	-	-	-	-
배치 2	0	-	-	-	0
배치 3	0	0	-	0	0
배치 4	-	0	0	0	-
배치 5	-	-	0	-	- (제거)
	-	0	0	0	-

(a) 전단보강 프레임의 배치

○ : 보강스팬

방법 \ 그리드 (X)	0-1	1-2	2-3	3-4	4-5
구성 1	-	-	-	-	-
구성 2	0	-	-	-	0
구성 3	0	0	0	0	0
구성 4	-	0	0	0	-
구성 5	-	-	0	-	-
	-	-	0	-	-

(b) 전단보강 스펀구성

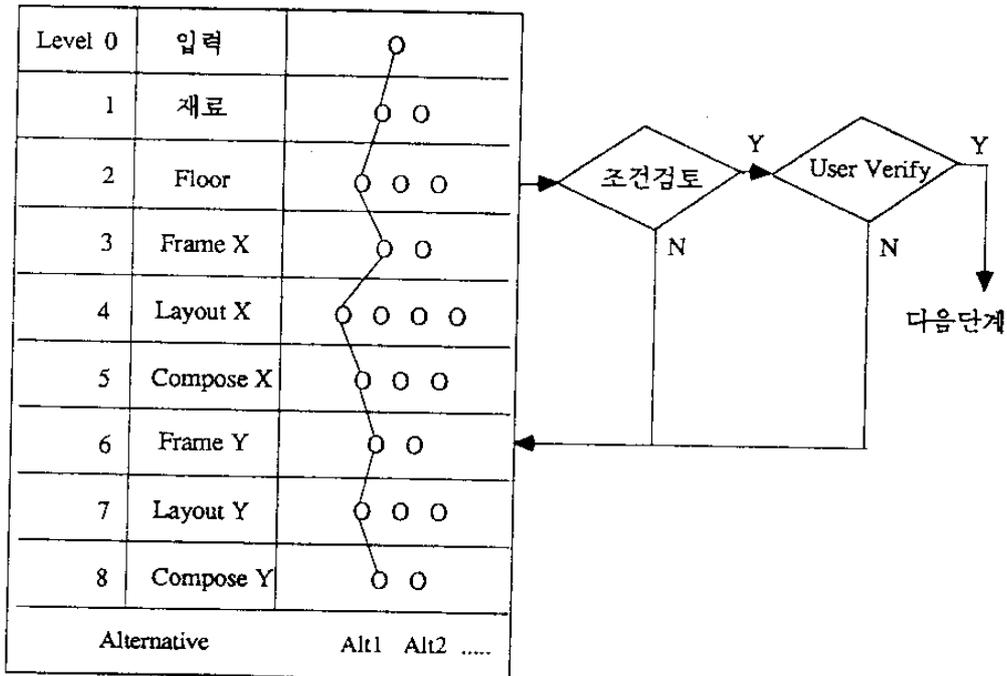


그림 4.6 건물후보의 재생방법

## (2) 건물후보재생

이러한 각 계층별 후보는 하나의 건물후보를 재생하기 위하여 각 계층별로 하나씩 선택되어 조합되는데, 이 단계에서는 깊이우선탐색에 의하여 상위레벨에서 하위레벨까지 선택하여 후보를 재생한다. 이 단계에서 다음과 같은 조합의 경우에는 후보대상에서 제거된다.

1. 설계자의 요구조건에 어긋나는 것 : 예를들면, 건물 재료의 단계에서 사용자가 원하는 재료가 제시된 경우에는 그 재료이외의 재료로 이루어진 후보는 제거된다.
2. 공간적 조건이 합당하지 않은 것 : 프레임의 종류가 모멘트저항일때 배치 1이 아니거나 구성 1이 아닌 경우, 또는 프레임의 종류가 전단보강이 된 경우에 배치 1이거나 구성 1인 경우에는 후보대상에서 제거된다. 그리고 프레임의 배치가 내부측으로 전단보강이 된 경우(배치 4)에는 프레임의 스패ن구성도 내부스팬보강(구성 4)이 대부분이므로 다른방법으로 스패ن보강이 된 경우는 제거된다. 또한 내부보강프레임의 배치크기가 해당방향의 프레임수의 반이 초과되는 경우에는 제거된다. 또한 사용자의 코어위치 선택으로 전단보강이 될 때는 프레임의 배치및 내부스팬구성은 배치 5 와 구성 5로만 조합되며 다른 경우로 조합된 것은 제거된다.
3. 이전의 선택과 불일치 하는 것 : 재료계층의 철골과 프레임계층의 벽체보강 프레임이 조합된 경우나, 또는 철근콘크리트와 브레이스보강 프레임이 조합된 경우는 후보대상에서 제거된다.
4. 구조적으로 합당하지 않은 것 : 전단보강되는 프레임인 경우 배치 3인 경우는 드물기 때문에 후보대상에서 제거한다.

이와같은 기준으로 예상되는 건물의 후보는 그림 4.7과 같다.

#### 4.3.2 분석단계

분석단계의 역할은 종합단계에서 제시된 후보에 대하여 구조적 가능성을 타진해 보는 단계로서, 이를 위하여 우선 분석하려는 조건을 설정하고 이러한 조건을 만족시키려는 과정으로 설계를 진행한다.

##### (1) 요소그룹의 설정

구조시스템을 구성하는 각 구조요소는 그룹별로 분류되어 부재의 크기나 성질이 그룹단위로 결정되고 처리되는데, 본 연구에서는 요소의 그룹을 일정한 층의 범위로 구분하며, 각 그룹내의 각 구조요소는 단면성질이 동일하다고 본다. 예를들면, 1층에서 4층까지가 한 그룹이면 이 범위의 모든 기둥은 그 단면의 성질이 같다고 본다. 이러한 그룹 층의 범위는 건물의 층수에 따라 조금씩 달라지는데, 본 연구에서는 다음 표 4.4와 같이 설정하였다.

##### (2) 설계하중

건물에 작용하는 외력은 그 성질에 따라 여러가지로 분류할 수 있는데, 본 연구에서는 자중을 비롯한 고정하중, 적재하중, 보상단하중, 풍하중, 지진하중등의 종류로 구분하였다. 이를 구조물에 가하기 위해서는 고정 및 적재하중은 슬라브 요소에, 그리고 벽체하중등에 의한 보상단하중은 보요소에 요소하중으로 설정하고, 수평하중은 건물의 층고에 따라 등분포된다고 가정하여 그 크기를 사용자가 입력하도록 하였다.

##### (3) 재료의 성질

재료성질에 대한 상수로는 단위하중, 탄성계수, 포아송비등을 들 수 있고, 각 부재

표 4.4 층수에 따른 그룹의 층범위

건물의 층수	그룹의 층 범위
90 층 이상	8 개층
70 층 이상	7 개층
50 층 이상	6 개층
30 층 이상	5 개층
15 층 이상	4 개층
15 층 이하	3 개층

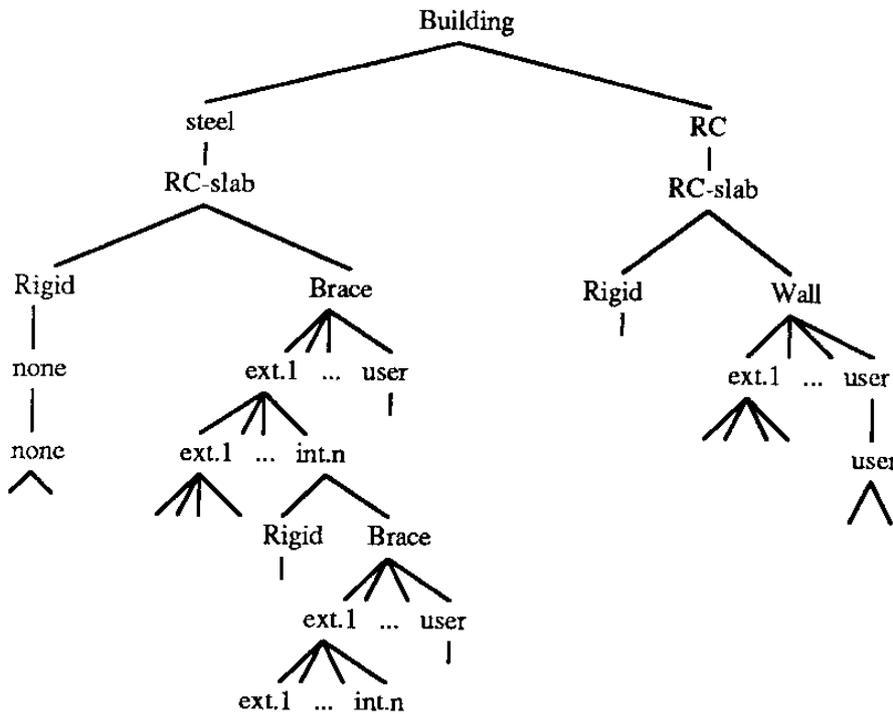


그림 4.7 건물 후보의 재생

를 설계하기 위한 허용응력은 철골일 경우에는 항복강도( $F_y$ ), 철근콘크리트인 경우에는 설계기준강도( $F_c$ )를 대상으로하여 참고문헌 B.17-21에 따라 계산하였다. 단, 수평 하중을 고려할 경우에, 위에서 계산된 허용응력을 33%증가시키는 사항은 하중경우의 계수로서 조절할 수 있도록 하였다.

#### (4) 하중경우의 수

위에서 언급한 설계하중을 건물에 실제 작용시키기 위해서는 하중경우의 수에서 해당하중의 부류에 따라 계수를 입력하여야 한다. 예를들면 중력하중만을 고려할 경우에는 고정하중, 적재하중, 보상단하중을 고려하는 사항에 계수 1.0을 입력하여야 한다. 그러나 풍하중을 동시에 고려할 경우에는 재료의 허용응력이 33%증가되므로 이를 감안하기 위하여 원하는 각 계수에 0.75를 곱하여야 한다.

#### (5) 보-기둥의 단면 데이터베이스

보요소 및 기둥요소의 설계시에는 기존의 단면데이터를 분석하여 저장해둔 데이터베이스를 이용한다.

#### (6) 조건의 형성

종합단계에서 선택되어진 건물의 형상에 대하여 구조적 가능성을 타진하기 위하여 해당후보를 구성하는 모든 부재가 재생된다. 이를 위하여 모든 후보에 대하여 슬라브, 보, 기둥요소가 재생되고, 브레이스 보강골조(Braced Frame/Frame-Sheartruss)에 대해서는 트러스 요소가, 벽체보강골조(Wall Frame/Frame-Shearwall)에 대해서는 전단벽요소가 추가로 재생된다. 해당후보의 구조적 적절성은 이러한 부재요소로 부재력을 구하고, 전도모멘트를 검토한 다음, 만족이 되면 부재의 단면산정의 가능성여부를 검

표 4.5 철골의 구조재료의 상수

설계 변수	내 용
항복 강도 (Fy)	: 입력하여야 함
허용 인장 응력 (ft)	: Fy/1.5
허용 휨 압축 응력 (fb)	: ft와 동일
허용 압축 응력 (fc)	: 주 1) 참조
허용 전단 응력 (fv)	: Fy/(1.5*√3)
단위 하중 (wt)	: 7.85 t/m <sup>3</sup>
탄성 계수 (Es)	: 2.1 × 10 <sup>6</sup> kg/cm <sup>2</sup>
포아송 비 (r)	: 0.3

주 1) 허용 압축 응력도는 압축재의 세장비에 따라 다음과 같다.

$$SR \leq Cc, \quad fc = \frac{\left(1 - 0.4 \left(\frac{SR}{Cc}\right)^2\right) Fy}{n} \quad (\text{kg/cm}^2)$$

$$SR > Cc, \quad fc = \frac{0.277 Fy}{\left(\frac{SR}{Cc}\right)^2} \quad (\text{kg/cm}^2)$$

여기서 Cc (한계세장비) :  $\sqrt{\frac{\pi^2 Es}{0.6 Fy}}$

Es : 탄성 계수

$$n \text{ (안전율)} : \frac{3}{2} + \frac{2}{3} \left(\frac{SR}{Cc}\right)^2$$

$$SR \text{ (세장비)} : \frac{lk}{i}$$

i (단면 2차 반경)

표 4.6 철근의 구조재료의 상수

설계변수	내 용
항복강도 (Fy)	: 2400, 3000, 3500, 4000 ( kg/cm <sup>2</sup> ) 등으로 입력
허용인장응력 (ft)	: Fy/1.5 와 2000 ( kg/cm <sup>2</sup> ) 중 작은 것
허용압축응력 (sfc)	: ft 와 동일
단위하중 (wt)	: 7.0 t/m <sup>3</sup>
탄성계수 (Es)	: 2.1 × 10 <sup>6</sup> ( kg/cm <sup>2</sup> )

표 4.7 콘크리트의 구조재료의 상수

설계 변수	내 용
설계 기준 강도 (Fc)	: 150, 180, 210, 240, 270, 300 (kg/cm <sup>2</sup> ) 등으로 입력
허용 압축 응력 (fb)	: 0.3 Fc
허용 휨 압축 응력 (fc)	: 0.4 Fc
허용 전단 응력 (v)	: 1.33 √Fc
단위 하중 (wt)	: 일반; 2.4 t/m <sup>3</sup> , 경량; 1.6-2.0 t/m <sup>3</sup>
탄성 계수 (Es)	: 2.1 × 10 <sup>5</sup> * (r/2.3) <sup>1.5</sup> * (Fc/200) <sup>0.5</sup>
포아송 비 (r)	: 0.1667

토하는 순서로 진행된다.

#### (7) 부재력 계산 및 전도모멘트의 검토

각 후보를 구성하는 부재들의 효율성을 검토하기 위해서는 구성부재들의 부재력에 의한 단면설정을 해보지 않을 수 없다. 본 연구에서는 이를 위하여 기존의 간단한 수 계산방법을 이용한다. 물론 복잡하고 정확한 해석이 최종안에 대해서는 수행되어야 하지만 예비설계단계에서는 수 많은 대안들에 대하여 정밀한 계산을 한다는 것은 시간 소비적이고, 비용이 많이 든다. 건물에 대한 간단한 근사해석으로 구성부재들의 부재력을 추정하고, 이를 바탕으로 부재의 예비단면을 가정함으로써, 세부설계단계에서 초기 단면가정을 합리적으로 하게되며 최적설계를 위한 수렴도를 향상시켜 계산시간을 단축시킬 수 있기 때문에 많은 대안중에서 불필요한 대안을 빨리 걸러내는데에 더욱 효과적이다.

각 후보를 구성하는 부재에 대한 부재력의 계산은 수직하중에 대해서는 부재의 분담면적에 의거하여 계산한다. 수평하중에 대해서는 건물의 폭 방향과 길이방향의 주된 구조시스템이 전담한다고 가정하여 각각 계산되는데, 프레임의 종류, 배치방법 그리고 보강스팬크기에 따라 다음과 같이 계산한다.

#### 모멘트 저항골조인 경우

강접합된 프레임 구성요소에 대한 부재력의 계산은 수직하중과 수평하중에 대하여 다음과 같이 계산된다.

○ 수직하중에 대한 해석 : 수직하중에 대한 각 요소의 부재력은 다음의 순서에 의하여 구하여 진다. 우선 각 슬라브 요소에 대하여 두 방향 스패의 크기 비율을 감안하여 고

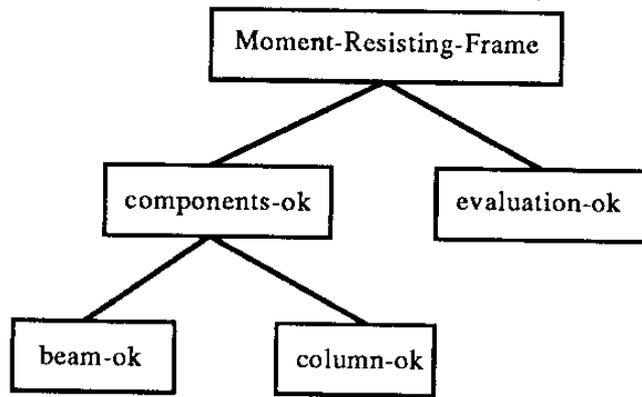


그림 4.8 모멘트 저항골조의 조건

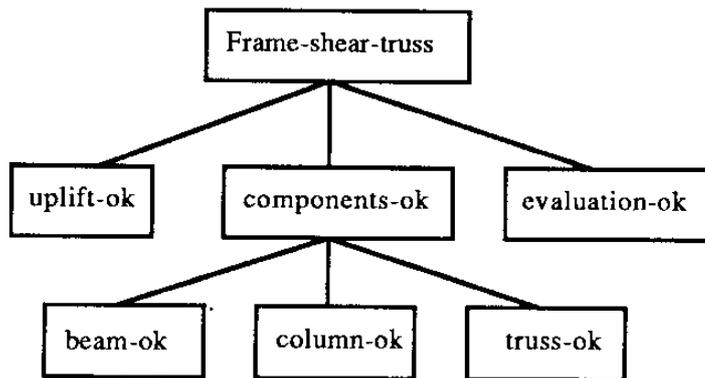


그림 4.9 프레임-트러스의 조건

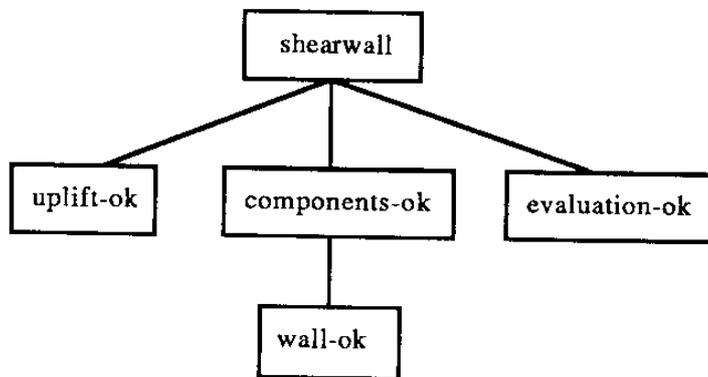


그림 4.10 프레임-벽체의 조건

정 및 적재하중에 대한 부재력을 구하게 되고, 모든 슬라브요소에 대하여 그 슬라브를 지탱하는 보에 슬라브의 스패크기에 따라 그림 4.11과 같이 분담하여 부재력을 계산하여 현재의 부재력에 가산한다. 그리고 모든 보에 대한 부재력이 구해지면 보의 고정단 모멘트 및 전단력이 그 보를 지탱하는 양측기둥에 보의 방향을 고려하여 가산한다. 윗층 기둥의 축력을 아래기둥으로 전달하기 위하여 최상층으로부터 아래층으로 기둥의 축력이 가산된다.

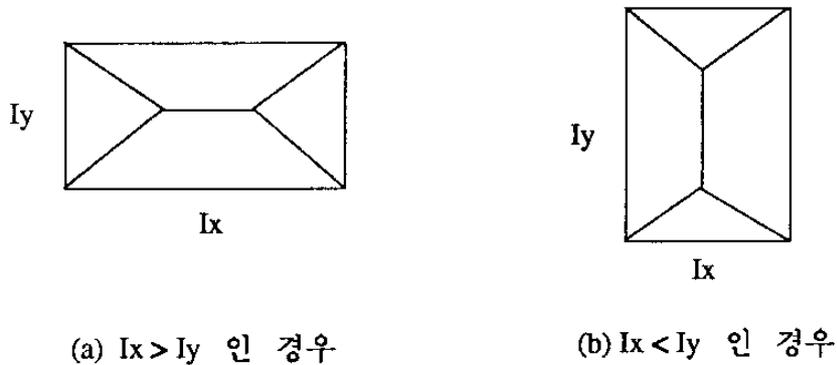


그림 4.11 슬라브하중의 보요소분담

○ 수평하중에 대한 해석 : 모멘트 저항골조의 수평하중에 대한 해석은 모든 프레임이 수평력을 동일하게 분담한다고 가정하여 일반적으로 널리 이용되는 포탈방법(Portal Method)을 이용하였다. 단, 포탈방법에 의하여 저층 및 고층부분에 대한 변곡점의 위치조정은 다음의 세부설계단계의 구조해석단계에서 보완이 되므로 무시하기로 한다.

이 방법은 다음과 같은 기본적인 가정을 필요로 한다.

1. 스패인이 여러개일 경우에는 각각의 쪽은 두개의 이웃한 기둥과 이들을 연결시키는 보로 구성된 여러개의 portal프레임으로 작용된다.

2. 모든 기둥의 변곡점은 기둥의 중심에서 일어난다.
3. 보의 변곡점도 보의 중심이다.
4. 스펠이 여러개일 경우에는 내부기둥의 전단력은 모두 같으며, 외부기둥은 내부기둥의 전단력의 반을 받는다. 이는 각각의 스펠의 크기가 다른 경우에는 옳지않게 되지만 예비설계의 특성상 각 스펠의 크기가 동일하다고 본다.

각 그룹의 기둥부재력의 계산은 임의의 그룹의 최하층에서 고려되는데, 부재력을 구하기 위하여 그룹의 최하 부분을 도시하면 그림 4.12과 같다.

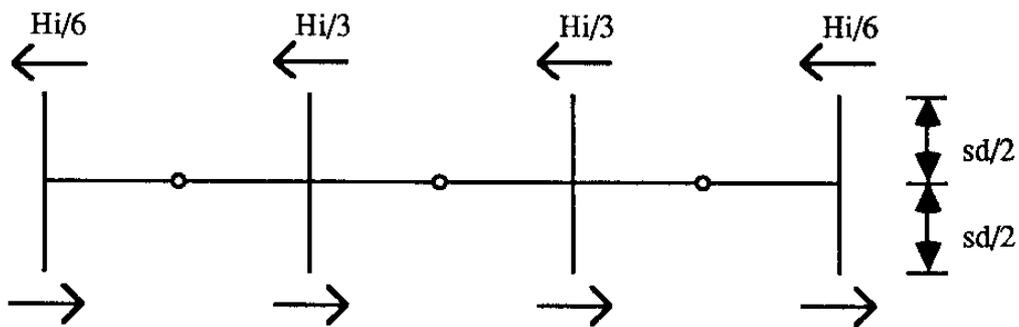


그림 4.12 기둥부재력의 계산

그림 4.12에서  $H_i$ 를 그 층윗부분에 있는 건물의 층고라 할 때, 그 층이 받는 모든 전단력과 전도모멘트는 다음과 같다.

$$H_i = W_w H_i \text{ mag} / n_f$$

$$M_i = W_w H_i H_i \text{ mag} / 2n_f$$

여기서  $W_w$ ; 풍하중,  $\text{mag}$ ; 풍하중을 받는 건물의 폭,  $n_f$ ; 풍하중을 받는 프레임의 수이

다. 기둥의 부재력은 내부, 외부에 따라 다르게 되는데 기둥의 전단력은 다음과 같다.

$$V_c = H_i / 2nbo ; \text{외부기둥인 경우}$$

$$V_c = H_i / nbo ; \text{내부기둥인 경우}$$

여기서  $nbo$ 는 프레임의 스패의 수이다. 전단력에 의한 기둥의 모멘트는 다음의 식에 의한다,

$$M_c = V_c sd / 2$$

여기서  $sd$ 는 기둥의 높이로서 층고에 해당된다. 한편, 전도모멘트에 의하여 기둥이 받는 축력의 계산은 예비설계의 특성상 외부기둥에 의해서만 지지된다고 본다. 그러므로

$$F_i = M_i / m_{ago}$$

여기서  $m_{ago}$ 는 프레임의 폭이다.

수평하중에 의한 보의 모멘트는 내부기둥이 받는 기둥의 모멘트의 크기와 같다고 본다.

#### 트리스보강 프레임인 경우

이 시스템의 분석방법은 참고문헌 B.24에 의하여 전단트리스시스템은 수직트리스 구조로 처리되는데, 기둥은 트리스 구조에서 상,하현재 역할을 하며, 트리스는 경사재 역할을 한다.

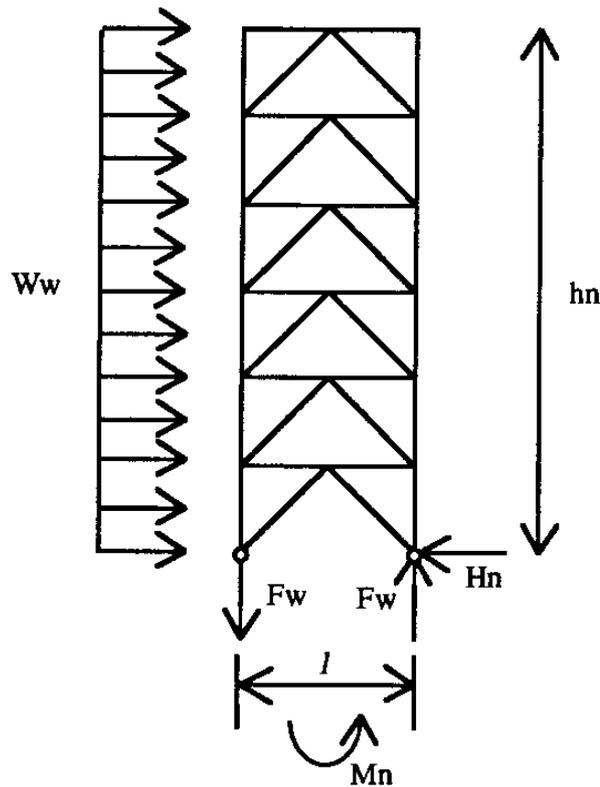


그림 4.13 프레임 전단트러스의 모델

○ 수직하중에 대한 해석 : 모멘트 저항골조인 경우와 같다.

○ 수평하중에 대한 해석 : 작용되는 수평하중에 대하여 건물의 평면에 배치된 모든 전단보강스팬이 공동분담한다고 가정한다. 기둥부재력은 전단보강되는 부분의 양 끝 부분에 위치하는 기둥의 축 방향력이 전단보강부분이 받게되는 전도모멘트를 지지한다고 가정하여 전도모멘트를 스패크기(양측기둥사이의 거리)로 나눈 값으로 계산된다. 트러스의 부재력을 구하기 위해서는 전단트러스를 K 브레이싱으로 가정하고, 부재는 최하층을 대상으로 한다. 트러스의 부재력은 전단보강되는 한 스패를 기준으로 하여 다음 식과 같이 구한다.

$$F_w = H_f / (2 \sin(a) \text{ span수})$$

여기서  $H_f$ 는 하나의 스펠보강부분이 받는 전단력이고,  $a$ 는 보와 브레이싱이 이루는 각도이고  $\text{span수}$ 는 하나의 전단보강스팬을 이루는 베이수이다.

○ 전도모멘트의 검토 : 전 기둥에 대하여 고정하중에 의한 축력이 수평하중에 의해 받게되는 축력의 1.5배보다 작을 경우에는 건물의 안정성을 위반한다고 보고 후보대상에서 제거한다.

$F_d$ 를 고정하중의 분담면적이 가장 작은 부분에서의 1층 기둥이 받는 축력이라하고,  $F_w$ 를 프레임 전단트러스에 작용되는 전도모멘트를 프레임의 폭으로 나누어 계산된 기둥의 축력이라 할 때, 프레임 전단트러스의 가능성에 대한 전도모멘트의 검토는

$$F_d \geq 1.5 F_w$$

에 의하여 검토된다.

#### 벽체보강 프레임인 경우

이 시스템의 분석은 벽체요소가 중력에 대해서는 내력벽의 역할을, 수평하중에 대해서는 전단벽의 역할을 하는 것으로 가정한다. 하나의 벽체요소의 길이는 스펠보강된 전 베이크기이다.

○ 수직하중에 대한 해석 : 모멘트 골조에서처럼 보, 기둥의 부재력을 구하고, 벽체가 있는 위치에 보, 기둥이 있을 경우에는 해당 보, 기둥의 요소는 제거되고 기둥의 축력이 벽체가 받는 하중으로 전달된다. 벽체이외의 부분에 위치하는 부재는 그대로 유지되어 중력에 대하여 지탱한다고 본다(그림 4.14참조). 프레임내에 전단보강되는 위치에 따

른 수직하중의 분담면적은 그림 4.15와 같다.

○ 수평하중에 대한 해석 : 각 전단벽이 수평하중을 전부지지하며, 그 크기는 공동으로 분담한다고 가정한다. 보강스팬의 총수를  $nf$ 라 하면 벽체의 전단력( $H_f$ ) 및 전도모멘트( $M_f$ )는 다음 식과 같이 구한다.

$$H_f = W_w H_n \text{ Width} / nf, M_f = W_w H_n^2 \text{ Width} / 2nf$$

○ 전도모멘트의 검토 : 각 벽체의 요소에 대하여 자중에 의한 안정모멘트가 전도모멘트의 1.5배보다 작을 때는 후보대상에서 제외된다.

#### (8) 하중경우의 수 조합

수직 및 수평하중에 대한 해석의 결과를 하중의 경우의 수로 조합한다.

#### (9) 부재설계

단면산정은 각 부재의 설계변수들을 정의하고 그 값을 구하는 것이다. 설계변수의 설정과 설계방법은 구조요소의 재료 및 요소 종류에 따라 다르게 되는데 요소의 설계에 대한 기본적인 개념은 철골구조에 대해서는 내장된 철골부재의 데이터베이스를 이용하고, 철근콘크리트인 경우에는 한정된 범위내에서 기존의 콘크리트 단면을 분석하여 저장한 데이터베이스에 의한 경우와 실제 설계를 행하는 두 가지 경우가 있다. 설계방법은 참고문헌 B.17-21에 의한 허용응력법에 따른다. 각 부재에 대한 개략적인 설계순서는 다음과 같다.

#### 슬라브의 설계

슬라브설계의 주된 설계변수는 단위길이(100cm)의 폭, 두께와 철근량으로 가정하고, 두께는 슬라브의 두 방향 스패의 1/40과 12cm중 작은 값, 철근량은 최소철근비로 초

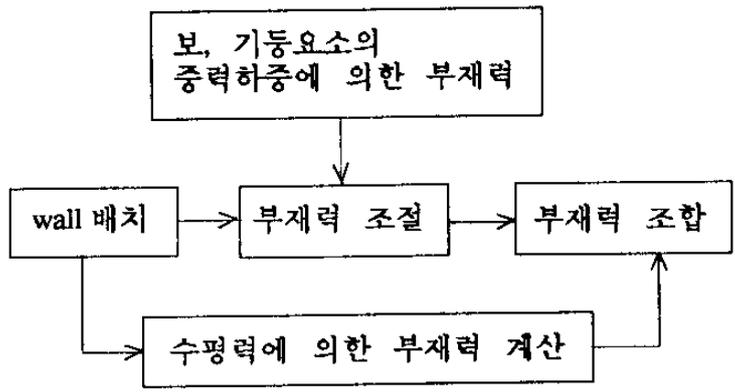


그림 4.14 벽체요소의 중력하중에 의한 내력계산

배치 종류	스팬수가 짝수인 경우	스팬수가 홀수인 경우
내부 위치		
외부 위치		
전 범위		

그림 4.15 전단보강위치 및 중력분담면적

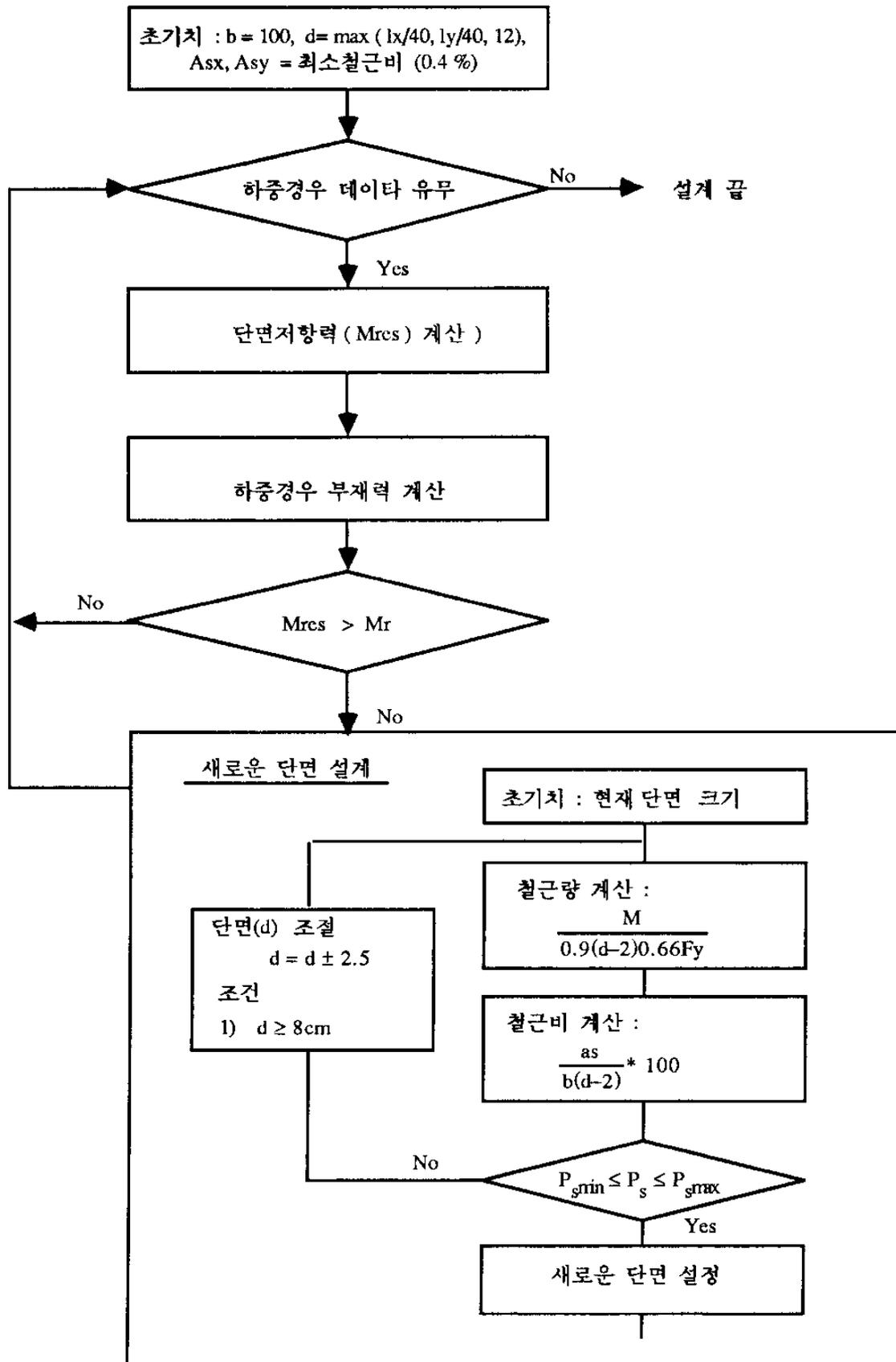


그림 4.16 슬라브의 설계

기가정하였다. 설계방법은 현재의 단면저항력과 각 하중경우의 부재력을 비교하여 단면저항력이 부재력보다 작으면 이 부재력에 대하여 단면설계를 수행하게 되는데, 현재의 두께를 단면초기치로 가정하고 다음 식과 같이 새로운 철근량을 계산하여 그 양이 최소, 최대철근비 조건을 만족시키도록 단면설계를 한다. 단 두께의 최소치는 8cm이다. 새로이 설계된 단면이 결과로서 선정되고 다른 하중의 경우에 대하여 설계가 계속된다.

$$A_s = M / (0.9 (t-2) 0.66 F_y)$$

% 초기가정 데이터

$d = lx/(16+24a)$ 에서  $a=1$ (두방향)이면  $lx/40$ ,  $a=2$ (한방향)이면  $lx/32$ 가 되는데 본 연구에서는  $lx/40$ 와 12cm중 작은 것을 선택하며,  $jd=0.9d$ 로 한다.

## 보의 설계

### (a) 철골보인 경우

I-BUILDS에 있어서의 철골보의 설계는 철골부재의 데이터베이스에서 요소를 선택하는 것을 말한다. 건물에 가해진 하중에 의하여 보요소의 부재력은 분석과정에서 계산이 되었으므로, 이 휨모멘트에 의한 부재의 응력이 앞에서 언급한 휨압축응력인  $0.66 F_y$ 를 초과하지 않게 부재를 선정할 수 있다. 이를 위하여 보부재에 발생하는 최대휨모멘트(+ 또는 -)를 허용응력으로 나눔으로서 요구되는 단면계수( $Z_{req}$ )를 산정한 후, 철골부재의 데이터베이스에서 이보다 큰 단면계수를 가진 형강을 택하면된다. 부재를 선정하는 과정에서 부재가 여러종류가 있을 경우에는 단위길이당 무게를 최소로 하는 경우와 춤(depth)을 최소로 하는 경우등의 여러 기준이 있을 수 있는데, 본 연구에서는 고층건물의 특성상 건물의 층고에 직접 연관이 되어, 전체건물에 더 큰 영향을 주

게되는 춤을 기준으로 하여 최소춤을 가지는 부재를 선택하도록한다.

철골보의 설계에 대한 알고리즘은 그림 4.17와 같다.

즉, 철골보의 설계인 경우에는 설계의 주된 변수를 단면계수로 가정하며, 초기단면의 설정은 최대모멘트를 허용휨압축응력( $0.66F_y$ )으로 나눈 값, 즉 요구단면계수를 구하고, 이 단면계수보다 큰 단면의 부재중에서 춤이 최소인 부재를 택하였다. 설계방법은 현재 단면저항력과 각 하중경우의 부재력을 비교하여 저항력이 현재의 부재력보다 작으면 이 부재력에 대한 요구단면계수를 새로이 계산하여 부재선택을 반복하게 된다.

#### (b) 철근콘크리트보인 경우

데이터베이스를 이용할 경우에는 단면저항력이 현재의 부재력보다 큰 것 중에서 최소비용을 가진 부재를 택하고, 실제설계를 할 경우에는 설계의 주된 변수를 폭, 깊이, 그리고 철근량으로 가정하고, 초기치로 깊이는 보 스패의  $1/15$ 과  $50\text{cm}$ 중 큰 값, 폭은 깊이의  $1/2$  과  $30\text{cm}$ 중 큰 값, 철근량은 최소철근비로 가정한다. 설계방법은 각 하중의 경우의 부재력이 현재 단면의 저항력보다 커면 현재의 폭과 춤을 초기치로 하여 최소,최대철근비를 만족하는 새로운 폭,춤,철근량을 구하게 된다. 본 연구에서는 폭과 춤의 크기를 그 비율이  $0.4-0.6$ 인 범위에서 설계하였다. 깊이의 허용 최대치는 초기치의 2배이다.

철근콘크리트 보에 작용하는 휨모멘트는 콘크리트의 압축력과 철근의 인장력에 의한 짝힘에 의하여 저항된다. 이는 콘크리트의 크랙을 가정하여 인장력을 모두 철근에 의하여 지지된다고 보는 것이다. 철근콘크리트 보의 유효깊이는 압축력을 받는 상단과 철근의 중심과의 거리로 나타내며 힘간의 거리(lever arm)는 대개 유효깊이의  $0.875-0.90$ 의 비율에 이르는데, 본 연구에서는  $0.875$ 로 가정하였다. 소요되는 철근의 인장력

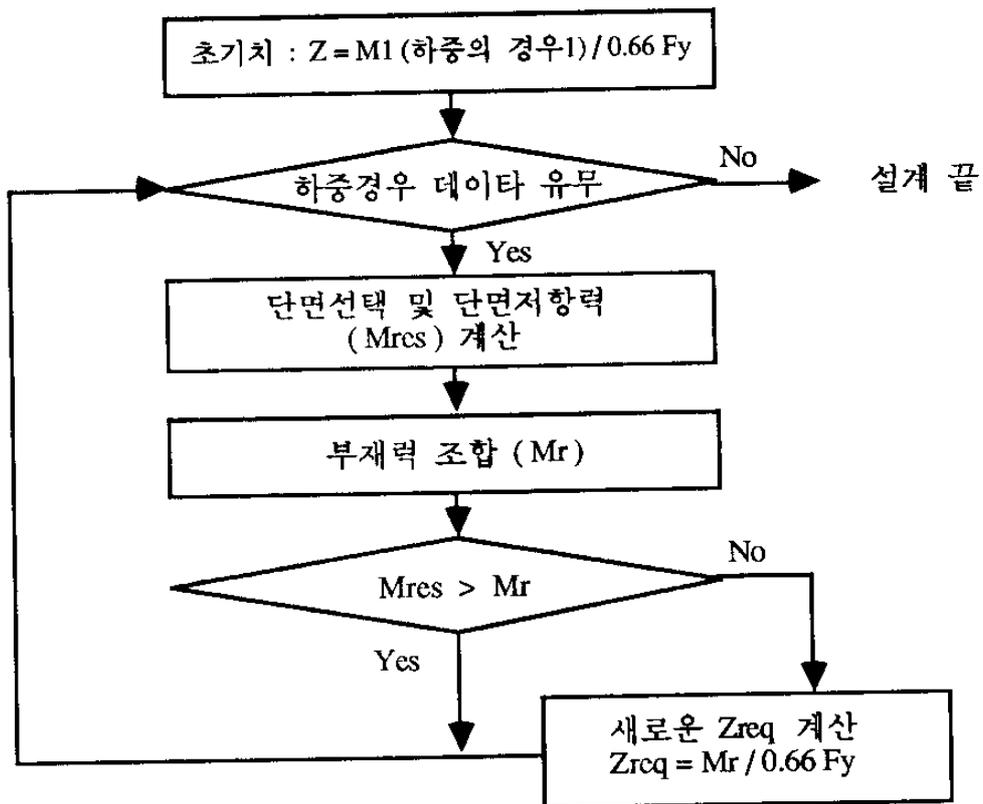


그림 4.17 첩골보의 설계

은 위에서 언급한 힘간의 거리와 부재력에 의하여 계산되어지며, 이는 철근비를 계산하는 데, 직접적으로 사용된다. 요구되는 철근비는 다음과 같이 계산한다.

$$A_s = M_{req} / (0.875 (d-3.0) ft)$$

$$AR = A_s / (b (d-3.0))$$

철근콘크리트 보의 철근비는 대한건축학회의 규준에 의하여 다음과 같이 정의되어 있으므로 그 만족여부를 검토하게 된다.

$$AR_{min} = 0.004$$

$$AR_{max} = AR_b \text{ 단 } AR_b = 0.5 (1/(f_t/f_b(1+f_t/nf_b)))$$

여기서 만약 철근비가 만족이 되지 않으면 제한된 범위내에서 설계를 반복하게 된다. 즉, 철근비가 너무 낮으면 보의 춤을 감소시키고 너무 높으면 춤을 증가시키게 된다. 이러한 과정은 철근비가 만족이 될 때까지 또는 보의 춤이 초기에 가정된 춤의 두 배가 되어 본 연구에서는 설계가 불가능하다고 판단이 될 때까지 반복한다.

철근콘크리트의 보설계의 알고리즘은 그림 4.18과 같다.

#### % 초기가정 데이터

휨모멘트에 의한 단면의 설계시는 폭, 깊이, 그리고 철근비의 미지수를 갖게 되므로 두 변수를 가정하여야 하는데, 본 연구에서는 작은보가 걸리지 않는 보는 춤이 스패의 1/15, 작은 보가 걸리는 보는 1/12인 경험적 지식을 활용하여, 춤을 1/15 및 50cm중 큰 값, 폭은 보통 춤의 0.25-0.6인 비율을 참고로 하여 0.5 및 30cm중 큰 것으로 초기설계를 수행한다.

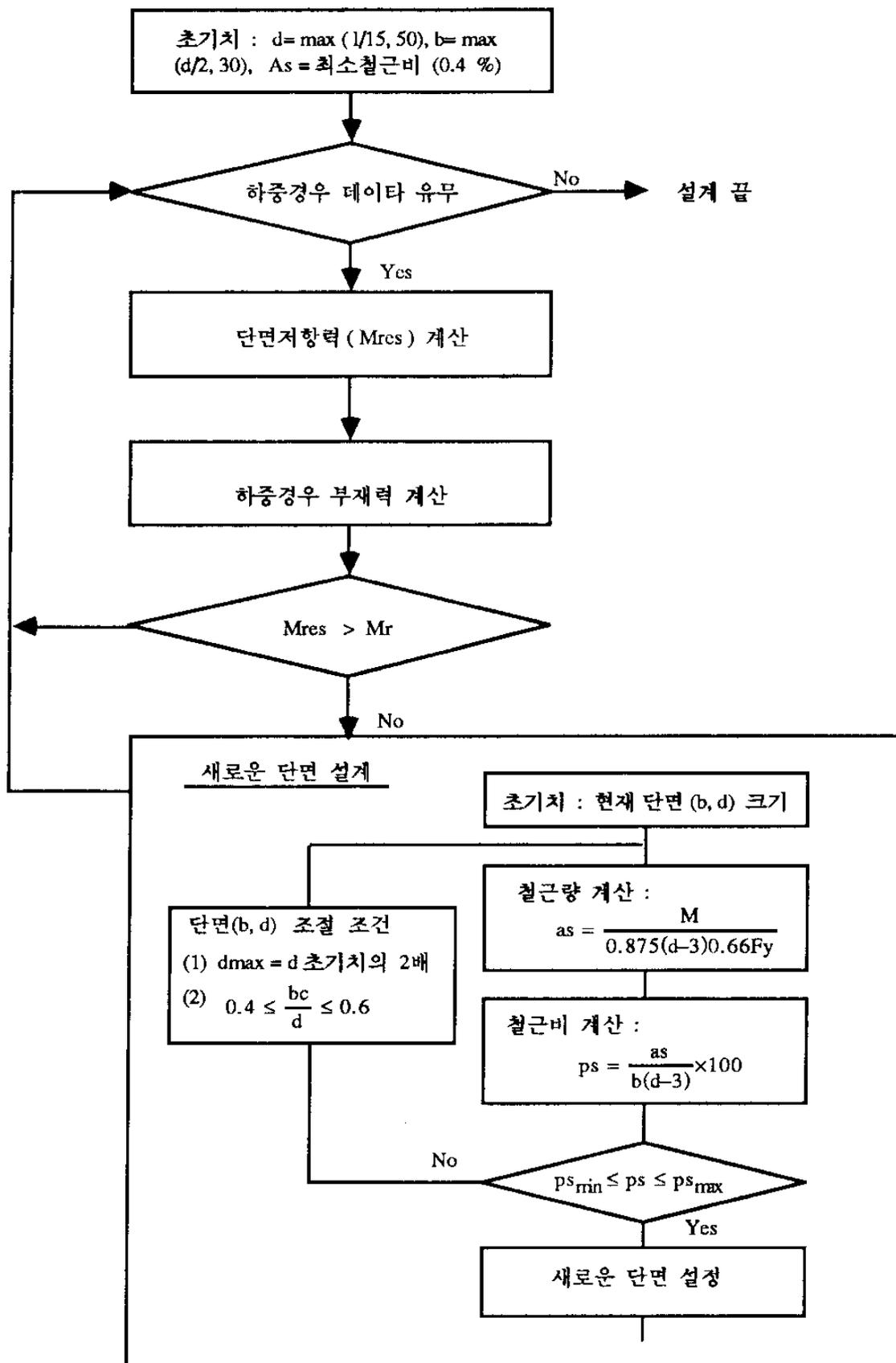


그림 4.18 철근콘크리트 보의 설계

## 기둥의 설계

### (a) 철골기둥인 경우

철골기둥의 설계는 국부좌굴, 기둥파괴 또는 보-기둥파괴중의 어느 형태인가에 따라 좌우된다. I-BUILDS에서는 철골기둥의 데이터베이스로 compact 단면(즉, 부재의 인장 및 압축파괴가 일어나기 전에 얇은 플랜지나 웨브가 좌굴되지 않는 단면)으로 구성되어 있으므로 국부파괴는 고려하지 않는다. 기둥파괴와 보-기둥파괴의 구분은 수평력이나 편심을 가진 수직하중에 의하여 휨응력이 존재하느냐의 유무에 따라 이루어진다. 어떠한 경우라도 허용압축응력을 결정하기 위해서는 기둥부재의 세장비가 한계세장비( $C_c$ )보다 작을 경우의 비탄성좌굴을 고려해야 한다. 기둥의 최대 세장비는 200으로 하며, 좌굴계수는 모멘트 저항골조는 1.2, 프레임 전단트러스인 경우에는 1.0으로 가정한다.

### ○ 축력만 작용하는 경우

기둥파괴의 경우에는 축방향력만 있는 경우이므로 이는 트러스와 동일한 방법으로 설계를 하면 된다.

### ○ 편심을 가진 축력 또는 축력+모멘트가 작용하는 경우

보-기둥의 파괴는 모멘트저항골조처럼 휨모멘트와 압축력이 작용할 때 발생한다. 축방향력에 휨모멘트가 작용할 경우에는 불안정하게 예상되므로 단면의 축방향의 저항능력이 감소된다. 본 연구에서는 철골단면의 선택을 다음의 기준에 의한다.

$$f_c'/f_c + f_b'/f_b \leq 1$$

여기서  $f_c$ ,  $f_b$ 는 허용압축 및 휨압축응력이고,  $f_c'$ ,  $f_b'$ 는 축력과 휨모멘트에 의한 실

재 축응력 및 휨응력이다. 윗식에서  $fc' = Freq/A$ ,  $fb' = B Mreq/fb$ , (단 B는 휨계수로  
서  $B = A/Z$ )로 두면

$$Areq \geq Freq/fc + B Mreq/fb$$

설계방법은 경험치에 의하여 B와 fb의 값을 상수로두고, Areq에 대한 이 식으로부터  
요구되는 단면적을 구할 수 있다. Areq보다 큰 단면의 부재의 새장비로부터 허용  
압축응력을 계산하여 윗식에 다시 대입하는 과정을 반복하여 만족되는 부재를 선정하  
다. 만족되는 부재를 선정할 수 없을 경우에는 에러를 제시하며 이 과정은 끝나게 된  
다.

철골기둥설계의 알고리즘은 그림 4.19과 같다.

즉, 설계의 중요한 변수를 단면적으로 보고, 초기부재를 선택하기 위하여 최대등가  
축하중(축력과 두 방향 모멘트중 큰 값으로 계산됨)을 가정된 허용압축응력( $0.375F_y$ )으  
로 나눈 값, 즉 요구단면적을 계산하고, 이 단면적보다 큰 부재중에서 최소단면적을  
가진 부재를 선택하였다. 설계는 각 하중의 경우에 대하여 현재 선택된 단면의 저항력  
(축력 및 양방향 모멘트)을 구하고 다음 식과 같이 그 가능성을 검토하여 위반시는 새  
로운 등가축하중에 의한 요구단면적을 계산하여 부재선택을 반복한다.

$$F / Fr + Mx / Mxr + My / Myr \leq 1$$

% 초기가정 데이터

compact단면의 휨계수는  $0.071 < B_x < 0.142$ ,  $0.193 < B_y < 0.591$ 의 범위의  
값을 가지게 되는데, 본 연구에서는  $B_x = 0.098$ ,  $B_y = 0.276$ 으로 가정하였다. 허용  
휨응력(fb)은  $0.60F_y$ 로 두고, 초기축방향응력(fc)은  $0.425F_y$ 로 하였다.

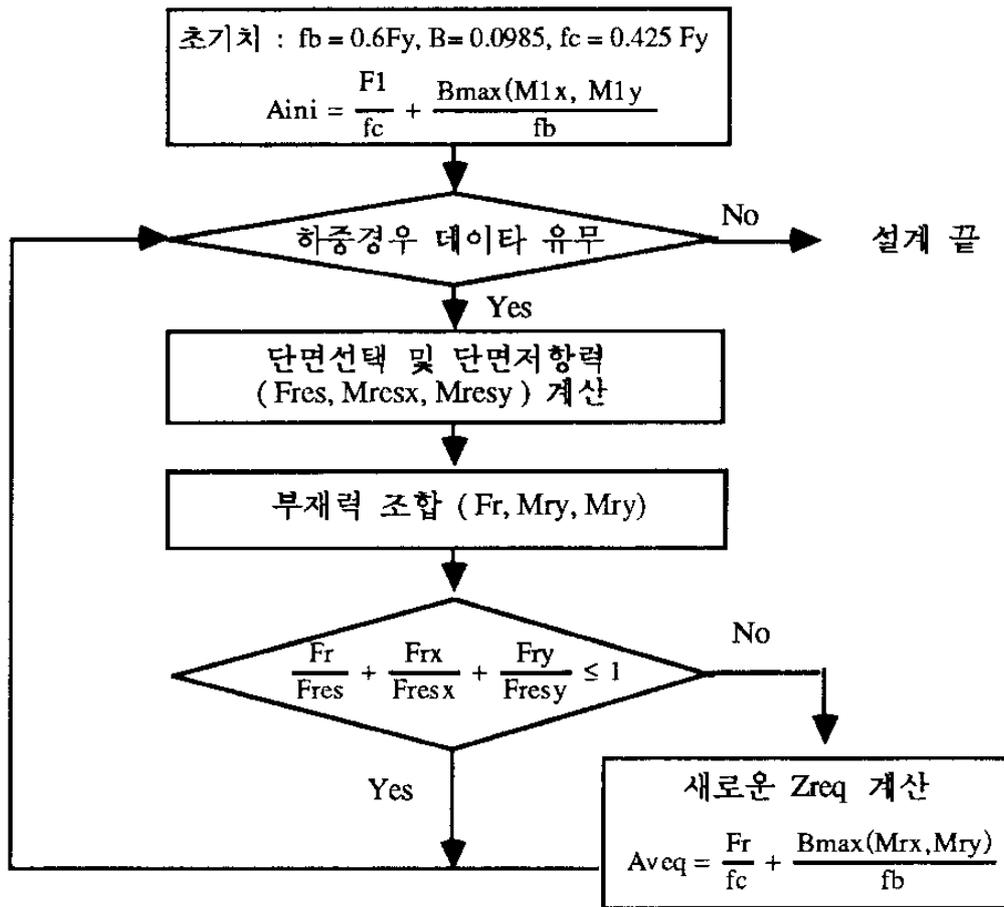


그림 4.19 철골기둥부재의 설계

(b) 철근콘크리트기둥인 경우

철근콘크리트 기둥의 설계는 단주일 경우의 기둥의 파괴와 장주일 때의 불안정한 경우의 두 가지 경우로 구분된다. 단주와 장주의 구별은 대한건축학회의 철근콘크리트 계산규준에 의하여 세장비( $SR=sd/d$ ,  $sd$ 는 기둥의 높이,  $d$ 는 기둥의 폭중 작은 값)가 15이하를 단주라 하고, 15-25를 장주라 한다. 25이상일 경우에는 본 연구에서는 고려하지 않는다. 여기서 기둥이라 함은 띠기둥을 이른다.

기둥을 설계하기 위하여 데이터베이스를 이용할 경우에는 철근콘크리트 보의 설계 방법과 같고, 실제설계를 할 경우에는 주된 설계변수를 폭(정사각형 기둥으로 가정)과 철근량으로 보고, 초기치로 폭을 높이의 1/10, 인접한 모든 스패중 최대 스패의 1/20에 10cm더한 것, 30cm중 최대인 것을 고르고, 여기에 지지하는 5개층마다 5cm씩 더한 값으로, 그리고 철근량은 철근비 1.0%로 가정하였다. 설계는 각 하중에 대하여 현재의 단면저항축력이 새로운 부재력의 증가축하중보다 작으면 새로운 단면설계를 수행하게 되는데, 단면의 초기치는 현재의 단면크기로하고, 세장비에 의한 단주, 장주효과를 감안한 증가축하중으로 최소, 최대 철근비조건하에서 설계를 한다. 기둥부재의 폭의 최소치는 층고의 1/20, 최대치는 인접한 최대스패의 1/6이다.

철근콘크리트 기둥설계의 알고리즘은 그림 4.20과 같다.

o 축압만 받는 경우

단주일 경우의 기둥의 허용축하중은 기둥의 단면적을  $A_g=d*d$ 라 하면,

$$F_a = 0.3 F_c A_g + 0.4 F_y A_{st}$$

장주인 경우에는 단주인 경우의 허용축하중에 강도계수(c)를 적용하여 허용축하중을

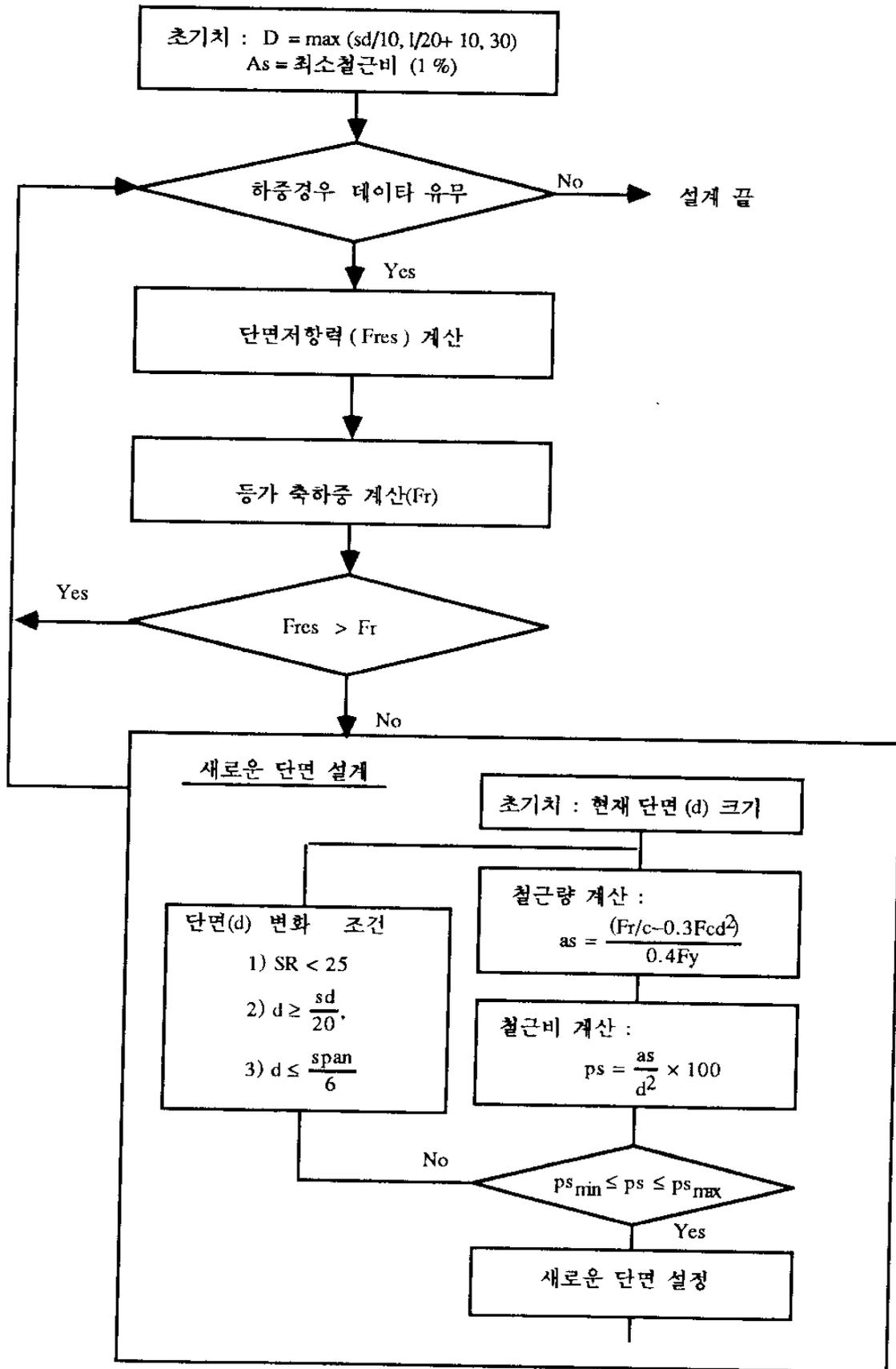


그림 4.20 철근콘크리트 기둥의 설계

감소시킨다.

$$c = 1.3 - 0.02 \text{ SR}, 15 \leq \text{SR} \leq 20 \text{인 경우}$$

$$c = 1.7 - 0.04 \text{ SR}, 20 \leq \text{SR} \leq 25 \text{인 경우}$$

윗 식을  $A_s$ 에 대해 전개하면

$$A_s = ( \text{Freq}/c - 0.3 F_c A_g ) / 0.4 F_y$$

여기서, 철근비를 구하면 다음과 같다.

$$AR = A_s / (d^2 100)$$

규준에서는 콘크리트의 균열 및 불가피하게 유발될 수 있는 휨모멘트에 대비하기 위하여 최소철근비와, 두 재료간의 상호작용이 충분히 이루어질 수 있도록 최대철근비를 규정하고 있다.

$$AR_{\min} =$$

0.4%, SR이 5이하일 때,

0.08 SR %, 5이상 10이하일 때,

0.8%, 10이상 25이하일 때,

$$AR_{\max} = 6.0\% \text{ (단, } F_c < 180 \text{이면 } 4.0\%)$$

그러므로 위에서 구한 철근비를 최대, 최소철근비와 비교하여, 너무 낮으면 기둥의 폭을 줄이고 너무 높으면 폭을 증가시켜 철근면적을 구하는 과정을 만족할 때까지 반복한다. 이 과정에서 적절한 범위내의 단면을 선정할 수 없을 경우에는 에러를 제시한다. 본 연구에서 정하는 적절한 범위는 다음과 같다.

$$sd/20 \leq d \leq \max(1/6, \text{초기가정의 두배})$$

#### % 초기가정 데이터

기둥의 폭은 보통 최상층일 경우  $sd/d$ 의 값이 8-12의 범위의 크기인데, 본 연구에서는 초기의 가정을 10으로 하고, 스펠에 대해서는  $1/20+10$ , 30층 큰 것으로 가정하고 그룹의 번호에 따라 5cm씩 더한다.

#### o 축하중과 휨모멘트를 받는 경우

철근콘크리트 기둥이 압축력과 휨모멘트를 동시에 받는 경우에는 기둥의 모멘트 효과를 축하중으로 증가시킨 등가축하중에 의한 설계를 한다. 참고문헌 B.24에 의하면 편심( $e=M_{req}/F_{req}$ )이 기둥의 폭의 1/4이 되는 시점에서는 기둥의 철근비가 2-3%인 보통의 경우에는  $2F_{req}$ 만에 의한 설계를 하면 모멘트의 효과를 감안한다고 알려져 있다. 본 연구에서는 이러한 사실을 바탕으로 등가축하중의 계산을 다음의 식처럼 한다.

$$F_{req}' = (1 + 4e/d) F_{req}$$

단 등가축하중에 의한 설계는  $e/d$ 값이 0.5이하인 경우에만 적용이 가능하며, 0.5이상일 경우에는 보처럼 설계를 해야한다.

#### % 초기가정데이터

초기 가정되는 기둥의 폭( $d$ )는  $4M_{req}/F_{req}$ 와 축 하중만 받는 경우와 비교하여 큰 것을 선정한다.

#### 트러스 부재의 설계

본 항은 건물의 구조가 철골인 경우인 프레임 전단트러스 시스템의 기둥과 트러스

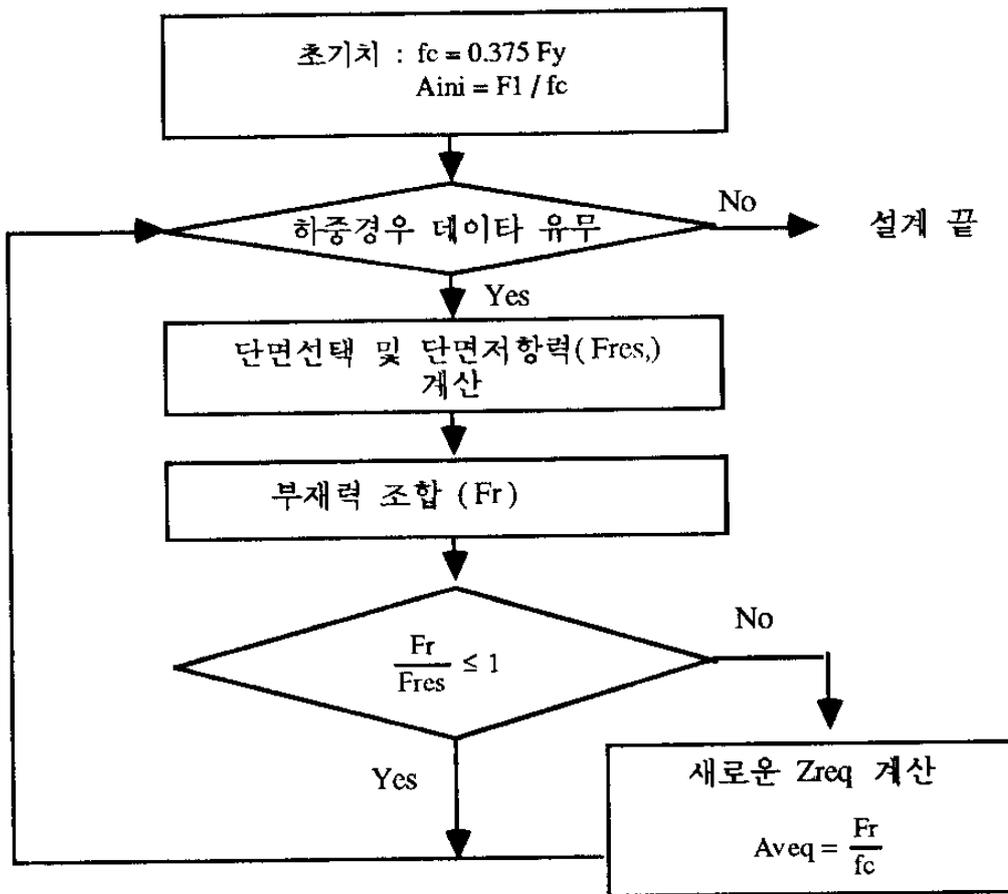


그림 4.21 트러스부재의 설계

부재에 대하여 적용이 되며, 이때의 부재는 휨모멘트가 작용하지 않고 또한 편심을 무시할 수 있다고 가정하여 기동과괴의 형태를 가진다.

부재의 설계방법은 초기에 허용압축응력( $f_c$ )를 가정하여 요구되는 기둥의 단면적을 계산하고, 이 면적보다 큰 부재를 선택한다. 선택된 부재의 단면 2차 반경으로부터 세장비를 구하고, 이를 바탕으로 허용압축응력을 구하여 허용축하중을 계산한다. 이를 요구되는 축력과 비교하여 허용축하중이 크면 만족하게 되고, 그렇지 않으면 새로운 허용압축응력으로 반복하게 된다. 만족되는 부재가 데이터베이스내에서 선정되지 않으면 부적절한 결과를 제시하며 이 단계가 끝나게 된다.

% 초기가정 데이터

허용압축응력을  $0.375F_y$ 로 가정한다.

### 벽체의 설계

벽체의 설계는 설계변수를 두께로 보고 초기두께는 15cm에 지지높이 3m마다 1cm씩 더한 크기와 30cm중 큰 값으로 한다. 설계는 최하층의 부재를 대상으로 현재의 부재력에 의한 응력이 허용응력범위인지를 검토하는 것으로 한다. 즉 최대축응력과 최대전단응력이 허용압축응력 및 허용전단응력범위일때는 설계가 만족된 것으로 보고, 위반시는 벽체의 두께를 5cm증가시켜 부재력을 구하는 과정부터 다시 반복된다. 두께의 최대치는 최초에 가정된 두께의 2배이다. 압축응력의 검토는 수직하중에 의한 압축응력과 수평하중에 의한 휨압축응력의 합으로 수행한다. 압축응력이 허용응력( $0.4 F_c$ )보다 작을 때 만족된다고 본다. 전단응력은 전단력을 단면적으로 나눈 값, 즉 평균전단응력의 1.5배가 허용전단응력( $= 1.33 \sqrt{F_c}$ )보다 작을 때 만족된다.

전단벽의 배치는 다음의 규칙을 따른다고 가정한다.

(1) 수평력에 저항할 수 있는 최소 스패 수 이상의 벽체를 배치한다.

(2) 코어가 있는 경우는 코어의 주변에도 배치한다.

(1)항인 경우의 전단응력과 수직응력의 계산은 다음과 같다.

$$V_{req} = 1.5 H_n / A$$

$$f_{req} = f_d + f_l + f_w$$

여기서 A : 벽체의 단면적

$f_d, f_l, f_w$  : 고정, 적재, 풍하중에 의한 수직응력

$$f_w : M_n / I_w \cdot l / 2$$

l : 전단벽의 길이

$I_w$  ; 전단벽의 단면 2차 모멘트

(2)항인 경우의  $I_w$ 의 계산은 다음의 식에 의한다.

$$I_w = b_e l_e^3 / 12.0 - b_i l_i^3 / 12.0$$

여기서,  $b_e, l_e$ 는 코어 바깥의 폭과 길이이며,  $b_i, l_i$ 는 코어 내부의 폭과 길이이다.

% 초기가정 데이터

전단벽의 초기 두께는 30cm로 가정한다.

(10) 부재의 그룹핑

각 요소의 부재설계시 같은 그룹내에 속하는 다른 부재의 단면저항능력이 현재부재의 단면저항능력보다 크면 다른 부재의 성질(재료 및 단면성질)이 그대로 선택된다.

### 4.3.3 평가단계

이 단계의 목적은 가능한 후보를 평가하기 위하여 일정한 기준하에 수치적인 순위를 부여하는 것이다. 이 과정에서 이용되는 평가방법은 기존의 수치알고리즘에 의한 black-box형태보다는 평가 및 순위를 부여하는 과정을 설계자가 명확하게 이해할 수 있도록 효율적이어야 한다. 이를 위하여 본 연구에서는 만족도 측정(Percentage Optimization)에 의한 방법을 선택한다. 이 방법은 각 평가요인별 단위에 상관없이 처리할 수 있다는 것이 장점이다.

각 평가요인에 대한 평가는 최적값에 이르는 정도에 따라 다음과 같이 분류된다.

#### (1) 최소화에 의한 평가

평가요인이 갖는 값의 범위에서 최소값이 요구되는 것으로서 다음과 같이 계산된다.

$$(\text{max-value})/(\text{max-min}) * 100$$

예를들면 건물의 높이가 90m일때 최대허용수평처짐은 27cm( $9,000 * 0.003$ )인데, 현재의 대안이 18cm였다면 수평변위에 대한 평가치는 33%의 만족도로 평가된다.

#### (2) 최대화에 의한 평가

평가요인이 갖는 값의 범위에서 최대값을 요구하는 경우에는 평가치는 다음과 같이 계산된다.

$$(\text{value-min})/(\text{max-min}) * 100$$

#### (3) 범위를 만족하는 경우

평가요인이 갖는 값의 범위에 드는 지를 검토하는 경우로서 주어진 범위에 들면 100%

만족, 들지못하면 0%로 평가된다.

이와같이 각 평가요인에 대한 평가치가 구해지면 현재의 대안에 대한 전체평가치는 선형함수를 이용하여 다음과 같이 구해진다.

$$\text{Eval} = \sum W_i X_i$$

여기서  $W_i, X_i$ 는 각각 가중치, 평가인자의 평가치이다. 가중치는 평가기준사이의 중요도를 설정하기 위한 것으로 값은 사용자의 입력에 의한다.

본 연구에서 사용되는 평가인자로는 수평변위, 건설속도, 프레임의 구성에 관한 난이성, 기둥이 받는 응력의 상태, 보 춤의 크기, 코어의 만족도, 내부공간의 효율성, 기둥의 단면크기, 건설비등을 들 수 있는데, 이러한 평가요인에 대한 종류는 표 4.8과 같으며 평가치를 구하는 방법은 다음과 같다.

표 4.8 평가인자 특성

평가요인	최소값	최대값	목적함수	W.F.
수평변위	0	건물고*0.003	min	1.0
건설속도	0	1.0	max	1.0
프레임 구성	0	프레임수	min	1.0
기둥의 응력	0	기둥의 총수	min	1.0
보의 춤크기	0	층고*0.2	min	1.0
코어링 정도	0	코어주변 벽수	max	1.0
내부공간	0	층베이수	min	1.0
기둥의 면적	0	건물면적*0.0025	min	1.0
건설비	0	*	min	1.0

\* 수평변위

이는 프레임의 수평처짐을 평가하기 위한 것으로 평가치는 수평처짐 크기이다. 수평 처짐의 계산방법은 참고문헌 B.24에 의하여 다음과 같다.

○ 모멘트저항 프레임인 경우 : 이 시스템의 처짐은 기둥과 보의 휨변형에 기초하며, 캔틸레버작용에 의한 기둥의 효과는 무시한다. 전체수평의 처짐은 전단력이 가장 큰 1층의 수평변위만을 구하고 이 값의 반을 각 층의 평균적인 수평변위로 보고, 이에 건물층수를 곱함으로써 전체처짐을 구한다. 1층의 수평처짐은 다음 식과 같이 계산한다.

$$\text{Drift1} = V_c \text{sd}^2 (\text{sd} / I_c + 1 / I_b) / 12E$$

여기서  $V_c$ 는 외부기둥의 전단력,  $I_c$ ,  $I_b$ 는 기둥, 보의 단면2차 모멘트,  $\text{sd}$ 는 층고,  $l$ 은 스패ن 크기, 그리고  $E$ 는 탄성계수이다. 그러므로 전체층의 수평변위는 다음의 식에 의하여 구할 수 있다.

$$\text{Drift} = (\text{Drift1} / 2) \text{nst}$$

○ 브레이스 보강 프레임인 경우 : 이 시스템의 수평변위는 축력에 의한 변위에 기인하며 트러스효과는 무시한다.

$$\text{Drift} = 2 F_w H_n^2 / (A l E)$$

여기서,  $F$ 는 풍하중에 의한 1층 기둥의 축력이고,  $A$ 는 기둥의 단면적이다.

○ 벽체 보강 프레임인 경우 : 수평처짐은 수평하중에 의한 캔틸레버형 처짐에 의한다.

$$\text{Drift} = W_w H_n^4 / (8 E I_w)$$

\* 건설속도

이 요인은 건물구조의 재료에 따른 건설속도를 평가하는 것으로서 평가치는 철골구조인 경우는 1.0, 철근콘크리트인 경우에는 0.0을 가진다.

\* 프레임의 구성난이성

철골 모멘트저항프레임인 경우에는 강접을 위한 불리한 조건을 수반하게 되므로 이를 평가하기 위한 것으로 평가치는 두방향의 철골 모멘트저항프레임 갯수를 나타낸다.

\* 기둥의 응력상태

철골 모멘트저항프레임인 경우에 기둥이 양방향의 힘을 받게되면 불리하게 되므로 이를 감안하기 위한 것으로 평가치는 두방향 철골 모멘트저항골조수를 곱한 값이다.

\* 보의 춤크기

1층에서의 보의 춤크기에 따른 층고의 영향을 분석하기 위한 것으로 춤이 클수록 층고를 높이므로 불리하게된다. 평가치는 보춤의 크기이다.

\* 코어의 유무

프레임 전단트러스 및 전단벽인 경우에는 구조체의 전단보강이 코어주변에 사용된 경우에는 유리한 점이 있으므로 이를 평가하기 위한 것으로 평가치는 현 후보에 의하여 코어 주변에 배치된 벽체수를 나타낸다.

\* 내부공간의 효율성

프레임 전단트러스 및 전단벽인 경우에 코어주변이 아닌 스펠에 전단보강이 되면 내부공간이 불리하게 되는데, 평가치는 코어주변을 제외한 곳에 배치된 벽체의 수이다.

\* 기둥의 단면적의 크기

건물의 저층에서 기둥의 단면적이 클 경우에는 내부의 사용되는 공간면적을 감소시키게 되므로 불리하게 된다. 평가치는 1층에서의 기둥의 총단면적을 나타낸다.

#### \* 건설비

각 프레임을 구성하는 건설비를 통계적으로 감안하기 위한 것으로 아직은 수행 불가능하다.

#### 4.3.4 선택단계

이 단계는 종합, 분석, 변수설정, 평가단계를 거친 구조시스템의 모든 후보에 대하여 선택하는 단계로서 사용자 자신이 원하는 후보 또는 평가치가 최대인 것을 자동적으로 선택할 수 있다.

#### 4.4 소 결 론

본 연구에서는 건축설계의 초기단계에서 건물의 완성에 커다란 영향을 주게되는 예비설계의 분야에 대하여 추론에 의하여 결과를 도출해 내는 전문가 시스템을 적용하였다. 이러한 분야의 연구는 지금까지 최적의 구조설계라는 세부설계과정에서의 경제성, 즉 물량추정에 의한 경제성에 국한되었던 컴퓨터에 의한 설계를 구조계획이라는 전문가영역까지 확대가능하게 하였다. 이렇게 함으로서 모든 데이터가 주어진 구조물에 대하여 많은 계산과 시간의 소모로 얻게 되었던, 구조적인 경제성이외에도 구조시스템의 안정성, 시공성, 그리고 건축공간성등의 타영역 및 경험적 영역까지를 고려한 많은 후보를 고려해 봄으로서 더욱 효율적인 설계를 가능하게 하고, 또한 설계의 오류를 초기에 방지하게 한다.

## 5. 구조물의 이상화

### 5.1 서론

#### 5.1.1 연구의 필요성

매트릭스에 의한 구조해석방법은 컴퓨터의 보급과 더불어 이제는 각 분야에서 널리 이용되고 있다. 이전에는 불가능했던 문제나 많은 시간이 소요되었던 문제도 이제는 단숨에 이들의 해를 구할 수 있다. 그러나 이러한 방법에 의하여 해를 구하기 위해서는 필연적으로 겪어야 할 과정이 있다. 그것은 많은 시간이 걸리고 지루한 작업인 입력데이터의 준비와 결과를 검토하는 과정이다. 이런 두 과정에는 사용자에게 의하여 의례적으로 에러가 유발된다. 이 과정에서 에러를 줄이고 사용자에게 작업의 효율성을 제공해주기 위하여 개발된 프로그램으로서 전처리기(Pre-Processor)와 후처리기(Post-Processor)가 최근의 프로그램에서는 제공되고 있다.

컴퓨터에 사용하여 수치해를 구하는 과정에서 발생하는 비용을 참고문헌 B.36에 의하여 표 5.1과 같다.

표 5.1 구조해석과정의 비용

작업내용	배치양식	대화식
형상모델링	65%	20%
컴퓨터비용	20%	30%
결과분석	15%	10%

표 5.1을 보면 데이터준비를 위한 형상모델링 과정에서 발생하는 비용이 65%이고, 결과의 분석과정에서 발생하는 비용이 15%로서 전체과정에서 컴퓨터사용이외의 비용

이 80%나 되며, 실제로 컴퓨터의 사용비용은 20%에 불과하다. 이러한 컴퓨터 사용비용은 근래에는 자체보유한 경우가 많으므로 절감시킬 여지가 별로없는데 비해 데이터 검토부분과 결과의 평가부분은 그 효율성을 높임으로서 비용절감을 할 수 있다. 그런데 결과의 분석부분은 전체의 15%정도밖에 안되므로 절감효과가 적고, 또한 그래픽지원에 의하여 많은 지원이 이루어지고 있다. 하지만 아직도 대부분이 수작업으로 진행되고 있는 입력모델의 준비에 관해서는 가장 경제성이 있다. 실제의 자료를 검토해보면 표 5.1과 같이 전처리기와 후처리기를 대화식으로 함으로서 모델비용은 20%, 컴퓨터결과 분석은 10%정도로 감소가 가능하고, 컴퓨터 사용시간은 30%로 증가되었다.

이러한 모델준비과정에서 대화식문장 또는 그래픽 에디터를 사용하면 데이터의 수정이나 해석의 재실행이 간단히 이루어진다. 여기서 가장 중요한 사실은 사용자가 설계하려는 개념을 실제의 설계안으로 빠른시간에 검토해볼 수 있다는 것이다. 이는 곧 같은 시간동안에 많은 대안을 고려하여 최종안을 선정함으로써 설계의 질(Quality)을 높일 수 있음을 의미한다.

오늘날에는 컴퓨터를 이용한 구조해석 프로그램은 수 없이 많다. 이러한 프로그램들은 일반적인 구조물을 대상으로 구조해석이 가능한 범용 프로그램과 건물과 같이 특정한 부류의 구조해석에 적합하게 그 용도가 국한된 전용프로그램등이 있는데, 이를 효율적으로 이용하기 위해서는 해당프로그램의 능력 및 그 장,단점이 명확하게 파악을 하여야 한다. 본 연구에서는 이를 해결하기 위하여 전문가시스템의 기본구조를 이용하여 건축물의 구조해석을 수행하기 위한 수치모델을 구축하는데 필요한 사용자를 위한 편의를 제공한다. 이렇게 함으로서 구조해석을 수행하기 위하여 사용자가 처리하여야 할 데이터를 최소화할 수 있고, 또한 필요한 정보들을 효율적으로 관리하며, 결국에는 이를 입력파일의 형태로 전환할 수 있게한다. 이 시스템은 예비설계단계에서 얻어진

데이터에 의하거나 또는 사용자가 직접 건물의 모델링을 하여 BUILDS-A에 의한 구조해석을 수행할 경우에 필요하다.

### 5.1.2 연구의 범위

본 연구에서는 건물의 구조해석을 위한 전용 프로그램이 BUILDS-A 프로그램의 해석모델개념에 의하여 건물의 구조해석 및 설계에 필요한 데이터준비를 지원해주는 시스템을 개발하고자 한다. 이 단계에서 사용자의 입력에 의하여 재생된 데이터는 구조해석을 위한 BUILDS-A 및 부재의 설계(BUILDS-S/C/F)등의 입력과일형태로 인터페이스 모듈에 의하여 전환된다.

## 5.2 건물의 모델링 개념

건물을 구성하는 기본적인 구성요소는 구성부재이며, 또한 기존의 유한요소 프로그램에서는 이러한 부재단위로 구조물을 묘사하고 있다. 이러한 구성부재를 묘사하기 위한 데이터는 크게 4가지로 구분된다. 즉, 요소의 위치를 나타내는 좌표계, 재료의 성질, 단면의 성질, 그리고 요소하중들이다. 구조요소는 이러한 4가지의 데이터로 대체로 묘사할 수 있는데, 보통은 데이터의 중복을 피하기 위해 4 개의 그룹으로 분류하여 요소의 묘사를 이러한 각 그룹에 대한 지칭(indexing)으로 하게된다. 여기서 각 요소를 정의하는 데이터로서 좌표계, 재료성질, 단면설질이 최소한 정의되어야 그 요소가 실제로 물리적인 의미를 가진다고 볼 수 있다.

본 연구에서는 부재의 묘사를 그림 5.1과같이 하며, 이때 잘못된 지칭에 대해서는 즉각 통고하도록 하여 오류의 데이터가 기억공간에 존재하는 것을 방지하여 해석의 잘못된 결과를 미연에 방지도록 하였다. 이와같은 데이터이외에도 건물전체의 하중과 기준에 의한 하중의 조합을 쉽게할 수 있도록하기 위하여 하중의 조합을 위한 데이터

가 있다.

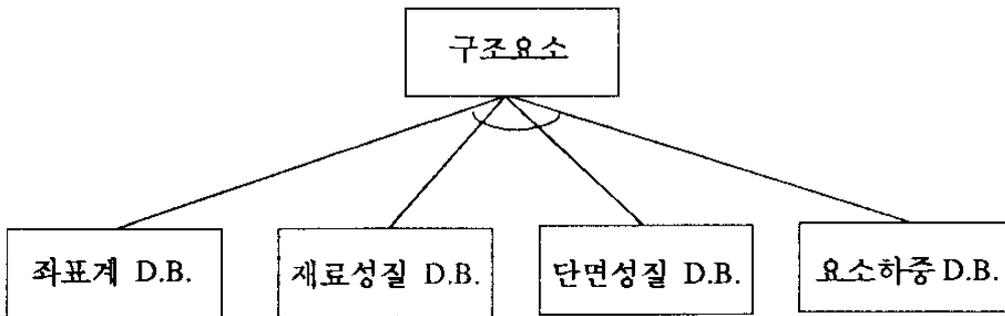


그림 5.1 구조요소의 데이터구조

이러한 데이터의 종류는 메뉴트리에 의하여 선택하게되고, 그 관리는 Action메뉴 (4.3.2 Global사실 참조)를 선택함으로써 이루어진다. 본 연구에서 건물을 묘사하는 데이터의 관리를 위하여 분류하면 다음과 같다.

- 좌표계 관련 : 건물의 x,y,z 그리드좌표
- 재료성질 : 건물의 부재재료성질
- 단면종류 : 부재종류별 단면의 성질
- 하중 : 요소하중및 건물하중의 입력
- 구조요소 : 슬라브, 보, 기둥, 벽체, 트러스요소의 처리
- 하중의 경우 : 하중의 조합형태조절

### 5.3 모델의 구축

#### 5.3.1 건물좌표계 데이터베이스

대부분의 건물은 정방형으로서 두 방향의 프레임이 교차됨으로서 수평하중에 저항하고, 각 층에 슬라브를 배치함으로서 수직하중에 저항하게 된다. 본 연구에서는 이러한 건물의 특성을 이용하여 건물의 좌표계를 각 방향의 그리드좌표값으로 입력하도록 한다. 건물의 좌표계는 X, Y 그리고 Z방향의 3종류로 나누어진다. 이러한 그리드의 각 좌표값은 부재요소와 건물의 하중에 관한 데이터에 의하여 지칭되게된다.

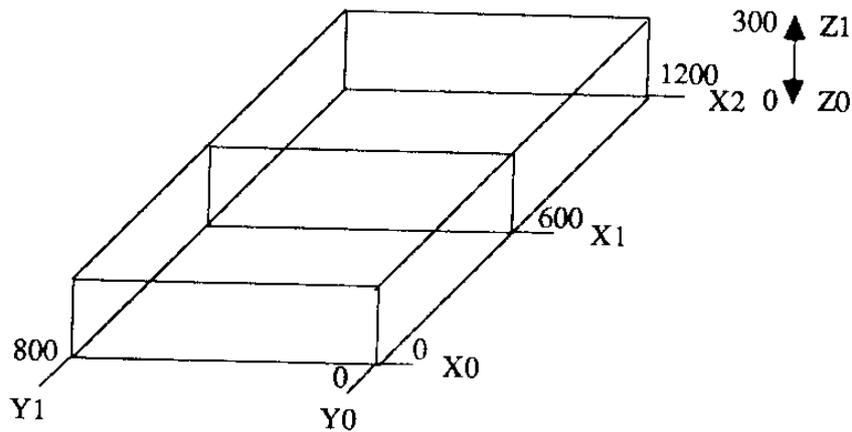


그림 5.2 모델링의 예제건물

#### o Xgrid 사실

기능 : 건축물의 X방향의 그리드상에서 위치를 나타낸다.

표현 : (Xgrid <NO> <X-COORDINATE>)

예 : 그림 5.2 예제건물의 데이터는 다음과 같이 표현된다.

(Xgrid 0 0.0)

(Xgrid 1 600.0)

(Xgrid 2 1200.0)

o Ygrid 사실

기능 : 건축물의 Y 그리드를 나타낸다.

표현 : (Ygrid <NO> <Y-COORDINATE>)

예 : 그림 5.2 예제건물의 데이터는 다음과 같이 표현된다.

(Ygrid 0 0.0)

(Ygrid 1 800.0)

o Zgrid 사실

기능 : 건축물의 Z 그리드를 나타낸다.

표현 : (Zgrid <NO> <Z-COORDINATE>)

예 : 그림 5.2 예제건물의 데이터는 다음과 같이 표현된다.

(Zgrid 0 0.0)

(Zgrid 1 300.0)

### 5.3.2 재료성질 데이터베이스

건물의 재료는 철골 또는 철근콘크리트 구조로 되어있다. 이러한 건물의 재료성질은 결국 각 구성부재의 재료성질로 정의되게 되는데, 철골일 경우에는 Steel만, 철근콘크리트일 경우에는 Steel사실이외에 Concrete사실까지 필요하다.

o Steel 사실

기능 : 철골재료상수

표현 : (Steel <NO> <ELASTIC> <RHO> <POISSON> <YIELD>)

예 : (Steel 1 2.1E06 7.5 0.3 4000 )

\* 본 연구에서는 자동적으로 위 예의 데이터가 입력된다.

#### o Concrete 사실

기능 : 콘크리트재료상수

표현 : (Concrete <NO> <ELASTIC> <RHO> <POISSON> <YIELD>)

예 : (Concrete 1 2.1E05 2.4 0.17 210 )

\* 본 연구에서는 자동적으로 위 예의 데이터가 입력된다.

### 5.3.3 단면성질 데이터베이스

부재의 설계단면은 부재의 종류 및 재료성질에 따라 다르게 표현된다. 본 연구에서는 각 부재를 묘사하는 단계로서 이러한 단면의 성질에 대하여 그 메뉴를 입력해두고, 이러한 단면의 종류를 지칭하도록 하였다. 단면의 종류는 다음과 같다.

#### o Slab-STL 사실

기능 : 콘크리트기둥 단면묘사

표현 : (Slab-STL <NO> <T>)

#### o Slab-CON 사실

기능 : 콘크리트슬라브 단면 묘사

표현 : (Slab-CON <NO> <T> <SBD>)

#### o Beam-STL 사실

기능 : 철골보 단면묘사

표현 : (Beam-STL <NO> <W> <D> <B> <Tw> <Tf> <A> <Ix>

<Iy> <ix> <iy> <Zx> <Zy>)

\* 철골보의 데이터는 본 연구에서 제공하는 데이터베이스를 이용할 수 있다.

o Beam-CON 사실

기능 : 콘크리트보 단면묘사

표현 : (Beam-CON <NO> <B> <H> <SBD> <Tn> <Cn> <Ze>)

\* 철근콘크리트 보의 데이터는 본 연구에서 제공하는 데이터베이스를 이용할 수 있다.

o Column-STL 사실

기능 : 철골기둥 단면묘사

표현 : (Column-STL <NO> <W> <D> <B> <Tw> <Tf> <A> <Ix>  
<Iy> <ix> <iy> <Zx> <Zy>)

\* 철골기둥의 데이터는 본 연구에서 제공하는 데이터베이스를 이용할 수 있다.

o Column-CON 사실

기능 : 콘크리트 단면묘사

표현 : (Column-CON <NO> <B> <H> <SBD> <Tn> <Cn> <Ag>  
<Ar> <Ax> <Zxg> <Zyg> <Zxe> <Zye>)

\* 철근콘크리트 기둥의 데이터는 본 연구에서 제공하는 데이터베이스를 이용할 수 있다.

o Wall-STL 사실

기능 : 철골판넬 단면묘사

표현 : (Wall-STL <NO> <T>)

o Wall-CON 사실

기능 : 콘크리트판넬 단면묘사

표현 : (Wall-CON <NO> <T> <SBD>)

5.3.4 건물 하중 데이터베이스

건물에 작용되는 하중은 크게 수직력과 수평력으로 구분된다. 여기서 수직하중은 중력에 의한 것으로 슬라브 및 보요소에 의하여 1차적으로 지탱하게되므로 슬라브 또는 보의 요소하중으로 처리되고, 수평하중은 바람 및 지진에 의한 하중으로서 건물전체하중으로 처리된다. 본 연구에서 사용가능한 하중에 대한 입력내용은 다음과 같다.

o Slab-DL

기능 : 슬라브 사하중 묘사

표현 : (Slab-DL <NO> <W>)

o Slab-LL

기능 : 슬라브 활하중 묘사

표현 : (Slab-DL <NO> <W>)

o Beam-DL

기능 : 보 사하중 묘사

표현 : (Beam-DL <NO> <ML> <VL> <MR> <VR> <MC>)

o Slab-LL

기능 : 슬라브 활하중 묘사

표현 : (Slab-LL <NO> <W>)

o Beam-LL

기능 : 보 활하중 묘사

표현 : (Beam-LL <NO> <ML> <VL> <MR> <VR> <MC>)

o Beam-BL

기능 : 보하중 묘사

표현 : (Beam-BL <NO> <ML> <VL> <MR> <VR> <MC>)

o Wind-Load

기능 : 풍하중 묘사

표현 : (Wind-Load <NO> <Story> <FX> <FY> <X-COORDINATE>  
<Y-COORDINATE>)

o Earth-Load

기능 : 지진하중 묘사

표현 : (Earth-Load <NO> <Story> <FX> <FY> <X-COORDINATE>  
<Y-COORDIANTE>)

### 5.3.5 건물형상 데이터베이스

건물의 묘사는 크게 두 단계로 분류할 수 있다. 첫째는 프레임레벨이고 두번째는 각 요소의 단계이다. 본 연구에서는 프레임 단계로 사용자가 입력을 할 경우에는 자동적으로 그 프레임을 구성하게 되는 각 요소가 자동적으로 발생되므로 어떤 방법으로든 정보를 입력할 수 있다.

o Frame-ID 사실

기능 : 프레임의 종류 묘사

표현 : (Frame-ID <ID> <DIR> <Xs> <Xe> <Ys> <Ye> <Zs>  
<Ze> <Cost> <Drift> <type> <<span-reinforce>>)

o Frame-No 사실

기능 : 프레임의 배치묘사

표현 : (Frame-No <ID> <DIR> <Xs> <Xe> <Ys> <Ye> <Zs>  
<Ze> <Frame-ID>)

o Floor-No 사실

기능 : 층 프레임 묘사

표현 : (Floor <ID> <DIR> <Xs> <Xe> <Ys> <Ye> <Zs> <Ze>  
<type>)

건물을 구성하는 구조적인 요소는 슬라브, 보, 기둥, 벽체, 그리고 트러스등이다. 건물을 묘사하기 위해서는 이러한 부재들의 데이터가 그 부재의 연결상태, 재료성질, 단면성질 그리고 요소하중등으로 구성되어야 한다. 여기서 부재의 연결에 대해서는 건축 구조물의 특성을 이용함으로써 쉽게 구성할 수 있다. 이러한 건물의 특성으로는

- 1) 평면상에서 기둥은 그리드의 교차점에 배치된다.
- 2) 보는 기둥사이에 배치된다.
- 3) 높이방향에 대해서는 부재배치가 반복된다.

등이다. 본 연구에서는 슬라브, 보 및 기둥에 대해서는 구조요소를 설정하려는 초기단

제에 자동적으로 그리드상에서 생성된다.

o Slab 사실

기능 : 슬라브요소의 묘사

표현 : (Slab <ID> <Xs> <Xe> <Ys> <Ye> <Zs> <Ze> <Steel>  
<Concrete> <Slab-STL> <Slab-CON> <Slab-DL> <Slab-LL>)

o Beam 사실

기능 : 보요소의 묘사

표현 : (Beam <ID> <Xs> <Xe> <Ys> <Ye> <Zs> <Ze> <Steel>  
<Concrete> <Beam-STL> <Beam-CON> <Beam-DL> <Beam-LL> <Beam-  
BL>)

o Column 사실

기능 : 기둥요소의 묘사

표현 : (Column <ID> <Xs> <Xe> <Ys> <Ye> <Zs> <Ze> <Steel>  
<Concrete> <Column-STL> <Column-CON>)

o Wall 사실

기능 : 벽요소의 묘사

표현 : (Wall <ID> <Xs> <Xe> <Ys> <Ye> <Zs> <Ze> <Steel>  
<Concrete> <Wall-STL> <Wall-CON>)

o Brace 사실

기능 : 트러스요소의 묘사

표현 : (Brace <ID> <Xs> <Xe> <Ys> <Ye> <Zs> <Ze> <Steel>  
<Concrete> <Column-STL> <Column-CON>)

\* 트러스요소는 기둥의 단면성질을 참조하게된다.

% 여기서 입력하게 되는 그리드의 시작번호와 끝 번호가 여러개의 그리드를 포함하게 될 경우에는 내부 그리드상의 요소는 자동적으로 분절되어 여러개의 요소로 생성된다. 이 때 사용자가 입력하게 되는 요소의 이름은 저절로 제거된다.

### 5.3.6 하중의 경우

건물의 구조해석을 하는 과정에서는 수직하중 및 수평하중의 해석결과에 대해서는 그 조합형태를 고려하여야 한다. 본 연구에서는 이를 위하여 하중의 조합에 관련한 정보를 보관할 수 있다.

#### o Load-Case

기능 : 하중의 경우 묘사

표현 : (Load-Case <NO> <DL> <LL> <BL> <WL> <EL>)

\* 사용자가 하중의 조합에 관련한 정보를 쉽게 처리할 수 있도록 하기 위하여 계획 단계에서 선택한 설계기준에따라 다음의 데이터가 자동적으로 처리된다.

- 허용응력법 설계인 경우 ;

(Load-Case 1 1.0 1.0 1.0 0.0 0.0)

(Load-Case 2 .75 .75 .75 .75 0.0)

- 극한강도법 설계인 경우 ;

(Load-Case 1 1.4 1.7 1.4 0.0 0.0)

(Load-Case 2 1.05 1.275 1.05 1.275 0.0)

#### 5.4 소결론

본 장에서는 건물의 구조해석을 위하여 사용자가 준비하여야 할 수치모델을 대화식으로 구축하기 위한 시스템을 구축하였다. 사용자에 의하여 준비된 데이터들은 BUILDS 프로그램에서 사용하기 위하여 원하는 설계의 진행단계에서 입력파일의 형태로 전환된다. 이러한 데이터는 설계과정의 진행에 따라 처리되는데 기존의 알고리즘에 의한 방법과는 달리 사실데이터로서 기억공간에 항상존재함으로서 유용하게 이용할 수 있다. 즉 BUILDS의 부프로그램에 의하여 공유될 수 있음은 물론 사용자가 데이터에 접근하여 수정 및 보완 할 수 있다.

## 6. 세부설계단계

### 6.1 입력파일의 구성

#### 6.1.1 연구의 필요성

사용자가 건물외 모델링을 도면상에서 하던지 또는 개념적으로 한 경우, 구조해석을 수행하기 위해서는 컴퓨터에 입력을 하여야한다. 컴퓨터에 의한 구조해석을 수행하는 과정에서 가장 많은 시간을 소비하며 사용자의 노력을 기울이게 되는 부분이 입력파일을 사용하려는 프로그램에 맞게 작성하는 과정일 것이다. 또한 작성된 입력파일이 처음의 시도로서 에러가 없다고는 보장을 못하기때문에 구조해석을 수행하기전의 단계로서 입력파일의 검토과정을 보통 거치게 된다. 검토과정에서 에러를 발견되지 않을 경우에는 구조해석이 실제로 수행된다. 그러므로 대부분의 구조해석 프로그램에서는 입력의 검토를 위한 기본적인 옵션이 있어 미리 작성된 파일을 검토해 봄으로서 컴퓨터비용을 절약할 수 있게 한다. 이러한 기능에도 불구하고 입력데이터가 에러가 없다고는 보장할 수 없다. 왜냐하면 검토기능이 보통은 요소강성의 조합이나 방정식을 푸는 과정을 생략하기 때문이다. 근래에는 이러한 입력과정의 에러를 줄이기 위하여 양식이 자유로운 (Free Format) 방식을 취하던지 또는 좌표계 및 요소의 연결상태가 규칙적인 경우에 대하여 자동발생기능을 채택하고 있다.

구조물의 모델링을 위해서는 대화식 컴퓨터 그래픽을 이용하는 방법이 가장 합리적이다. 요소의 입력과정에서 가장 많은 양을 차지하게 되는 것이 좌표값과 각 요소를 묘사하는 부분이다. 이러한 데이터를 검토해보기 위하여 이용되는 것이 전처리기와 그래픽이다. 요즘은 구조해석 모델의 자동화를 위하여 프로그램에 의한 메쉬(Mesh)발생기능과 Adaptive해석기법이 연구되고 있다.

연구에서는 전문가시스템을 이용하여, 건물의 모델링으로 입력화일을 구성하고, BUILDS-A에 의한 구조해석, 해석결과와 검토, 그리고 부재의 개선등의 작업을 지원하도록 한다. 사용자가 모델링과정을 거치지 않고 직접 입력화일을 작성한 경우에는 그 화일의 올바름을 검토해 볼 수 있으며 곧바로 실행이 가능하다. 이러한 방법은 곧바로 부재의 최적설계를 행하는 BUILDS-S/C/F, 및 구조물의 적산, 도면화 작업을 수행하는 BUILDS-B/D에게도 적용된다. 본 연구에서는 BUILDS-A에 의한 구조해석 프로그램과의 연결(interface)에만 적용한다.

### 6.1.2 입력파일의 작성개념

대화식으로 입력을 하게되는 프로그램을 제외한 대개의 프로그램은 입력카드의 작성요령을 사용자를 위한 안내서에 기술하고 있다. 그러므로 카드의 순서에 준하여 카드를 입력해 나가면서 데이터의 값에 따라 카드의 선택이 이루어지게 되는데, 필요한 카드의 선택이 작업의 진행과 더불어 자동적으로 되고, 각 입력변수에 대한 필드(Field)가 준비되어 알려진 변수에 대해서는 그 값이 전달되고, 값이 미지수인 변수에 대해서는 default나 사용자로부터 입력을 요구하는 등의 적절한 처리가 이루어질 수 있다면 입력파일의 자동 발생은 가능해질 수 있을 것이다. 대체로 입력화일을 구성하는 데이터는 논리적으로 대개가 제어카드가 앞서고, 이 카드가 갖는 값에 따라 그룹별 다음 카드가 작성된다. 그러므로 이러한 카드의 구조를 조직화하면 트리 구조로 표현할 수 있고, 각 데이터는 깊이우선의 탐색에 의하여 순차적인 화일로 전환시킬 수 있다.

본 연구에서는 다음과 같은 순서로 입력파일의 작성한다.

첫째, 이전의 카드로부터 현재 필요한 카드를 만든다.

들제, 카드의 각 변수에 대하여 알려진 데이터는 값을 전달한다.

새제, 현재의 카드가 같은 레벨 또는 더 낮은 레벨의 카드의

선택에 영향이 있을 경우에는 이를 처리한다.

위의 과정을 마지막카드까지 반복한다.

내제, 카드를 탐색에 의하여 순서를 정하면서 이를 입력파일에

담는다.

### 6.1.3 입력파일의 구조

입력파일을 구성하는 카드의 구조는 Master Card라 불리는 제어카드, 좌표계 카드, 각 요소의 데이터 카드, 하중카드, 그리고 하중조합카드등으로 몇 개의 블록으로 이루어진다. 이러한 카드는 물리적으로 순차적인 파일에 저장되지만 데이터파일을 이루는 카드는 논리적으로 계층적인 구조를 지니고 있다. 즉 각 데이터 블록은 해당블록의 제어카드와 제어카드에 따른 보조카드로 구성되는 계층구조로 나누어진다. 예를들면 마스터카드는 절점수, 요소의 종류수, 하중의 수, 하중의 조합의 갯수등의 제어정보, 즉 다음의 입력을 조절하는 데이터를 담고 있으며, 각 데이터군은 또한 그 그룹의 제어카드와 뒤따르는 카드그룹으로 구성되고 있다. 입력파일을 구성하는 여러카드중에서 제어정보를 담게되는 카드는 매수가 일정하게 정해지는데 반해, 뒤따르는 카드들은 제어카드에 있는 값에 따라서 그 매수가 정해지므로 이러한 데이터의 관리를 위해서는 링크구조가 적당하다.

위와같은 연유로 입력파일을 이루는 카드를 처리하기 위해서는 주제어카드를 루트 (Root) 노드로 하는 트리구조로 조직화할 수 있다. 이러한 구조는 그림 6.1과 같이 입력파일의 작업이 진행되어 감에 따라 상위레벨의 카드에 따른 정보에 의하여 하위 레벨의 카드변수및 카드매수가 정해지게된다. 입력파일로의 데이터전달을 위한 카드의

순서는 트리구조의 깊이우선탐색에 의하면 가능하다.

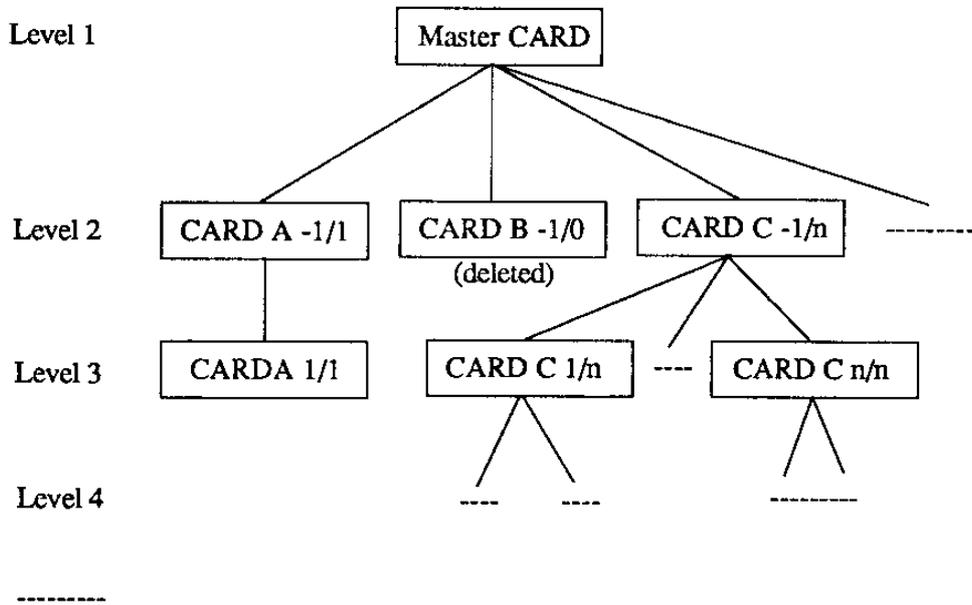


그림 6.1 입력카드의 구조

그림 6.1과 같은 카드의 구조화를 위하여 사용된 데이터 항목은 다음과 같다.

( <카드이름> <다음카드이름> <하위카드이름> ... [ <FORMAT> <변수리스트> ] )

여기서 첫 3개의 포인터(Pointer)는 카드의 구조화를 위하여 이용되는데, <카드이름>은 현재카드를 구분짓는 이름이며, <다음카드이름>은 같은 레벨에 존재하는 카드의 이름이며, <하위카드이름>은 현재의 카드하위에 위치되는 카드의 이름이다. 그리고 <FORMAT>은 변수리스트에서 나타나는 각 변수들의 출력양식인데, 이는 트리구조의 최끝단 노드에서만 나타나게 된다. 그러므로 하위레벨을 가지는 카드는 단지 카드의 연결상태에 관한 정보만을 갖게되므로 <FORMAT>이후의 정보는 필요하지 않게

된다. 표 6.1은 BUILDS-A에서 이용되는 입력카드의 구조를 보여주며 표 6.2는 입력카드의 변수들에 관한 출력양식을 보여주고 있다.

그림 6.1에서 상위레벨의 카드가 가지는 값에 따라 그 하위레벨의 카드메수가 정해지므로 카드의 데이터구조에는 같은 양식의 카드를 여러개 재생하는 항목이 필요하다. 즉 상위레벨의 어떤변수 값이 하위레벨의 다발생항목에 전달되게 되면 그 수만큼의 하위레벨카드가 링크구조로 자동적으로 재생된다. 만약 이 값이 0이면 하위카드는 안내서에서 제시하는 방법에 따라서 제거되거나 빈(Blank)카드로 처리된다. 이를 처리하는 규칙은 다른 내용에 우선하게 되므로 규칙의 우선권이 높게 되어있다.

이러한 방법에 의하여 구성된 입력카드는 3개의 포인터를 이용하여 깊이우선탐색 (Depth First Search)에 의하여 순서가 정해지므로 순차적인 파일에 정해진 출력양식과 변수값들로 효율적으로 전달된다. 이러한 기능을 하는 규칙을 예들들면 다음과 같다.

```
; 현재의 메모리에 있는 카드의 데이터를 입력파일의 형태로
; 저장하기 위하여 순서를 정하는 규칙
(defrule card-order2 "ordering"
(declare (salience -1000))
?rem <- (CARD STACK ?stack $?Stack QUEUE $?Queue ?cur)
(CARD ?cur ?next ?sub $?FORM)
=>
(retract ?rem)
(if (and (neq ?next no) (neq ?sub no))
```

```

then (assert (CARD STACK ?next ?stack $?Stack QUEUE $?Queue ?cur ?sub)))
(if (and (neq ?next no) (eq ?sub no))
then (assert (CARD STACK ?stack $?Stack QUEUE $?Queue ?cur ?next)))
(if (and (eq ?next no) (neq ?sub no))
then (assert (CARD STACK ?stack $?Stack QUEUE $?Queue ?cur ?sub)))
(if (and (eq ?next no) (eq ?sub no))
then (assert (CARD STACK $?Stack QUEUE $?Queue ?cur ?stack))))

```

#### 6.1.4 데이터의 전달

일단 필요한 카드가 준비가 되면 현재의 데이터메모리에 있는 알려진 변수는 규칙에 의하여 값이 전달된다. 여기서 규칙의 우선권은 규칙에 할당되는 수치로서 높은 수치를 가진 규칙이 낮은 수치를 가진 규칙보다 conflict resolution과정에서 발생가능한 규칙의 리스트 스택(stack)의 상위에 위치하게 된다. 그러므로 메모리에 있는 값을 카드의 변수에 전달해주는 규칙은 일반적인 규칙보다는 우선권이 높고, 카드의 발생과 제거에 관련한 규칙보다는 우선권이 낮게된다. 이러한 방법에 의하여 규칙에 의하여 만족이 되는 데이터는 쉽게 데이터카드에 전달된다.

때로는 카드의 변수의 값이 상위레벨 또는 이전의 카드의 값에 따라 영향을 받을 때도 있다. 예를들면 주제어카드에서 정적해석 (Static Analysis)옵션이 선택이 되었을 경우에는 뒤따르는 카드의 변수중 건물의 질량, 모드갯수등은 불필요하게 되는데, 이러한 카드간의 관계는 사용자 안내서에 설명이 따르므로, 이에따라 관련된 지식을 쉽게 추출하여 규칙으로 내장할 수 있다.

제5장의 건물의 모델링과정을 통하여 데이터를 준비하더라도 해석을 위한 모든 정

보가 제공되는 것은 아니다. 이러한 예로는 건물전체에 관한 것으로서 총질량, 지진데이터등이 있다. 상당한 양의 데이터가 지식베이스에 보관되어 있지만 입력데이터를 준비하는 과정에서는 이러한 데이터는 사용자로부터 직접 입력되는 것이 편리하다. 만약 제5장에서 언급한 모델링과정이 없이 사용자가 직접 이 모듈을 선택한다면, 이는 메모리에 지식이 하나도 없는, 즉 데이터카드가 재생되더라도 전달될 지식이 없기 때문에 모든 데이터를 사용자로부터 입력을 요구하게된다. 이러한 방법으로 입력을 하게되는 형식은 곧, 대화식 입력방식이 되므로 본 연구의 지원은 매뉴얼역할을 하게된다. 제5장을 통하여 데이터의 준비가 이루어진 경우에는 일부는 자동적으로 일부는 사용자의 입력에 의하여 데이터를 입력하게 되는데, 이때는 입력의 대부분을 차지하게되는 좌표 및 요소의 데이터는 자동적으로 준비가 되므로 상당한 도움을 받을 수 있다.

#### 6.1.5 소결론

본 연구는 데이터베이스에 있는 정보를 구조해석을 위한 입력파일로 전환하기 위하여 사용자 안내서에서 제공하는 지식을 바탕으로 데이터 카드를 구조화하고, 데이터베이스의 사실을 규칙을 통하여 데이터카드로 전달함으로써 입력파일의 자동발생에 대한 사항이다. 이렇게 함으로서 구조해석 프로그램의 사용을 위한 매뉴얼로서 사용자를 위하여 그 기능을 충분히 발휘할 수 있는 지원시스템의 개발이 가능하게 되었다.

이 시스템은 건물의 해석모델에 의한 입력파일작성에 소요되는 시간을 절약시켜주고, 데이터의 일관성을 보장하며, 오류를 미연에 방지하는 등의 편의를 사용자에게 제공하여 줌으로서 설계자로하여금 구조해석의 결과분석에 보다 많은 노력을 기울일 수 있게 한다.

#### 6.2 입력파일의 검토

### 6.2.1 개념

구조해석을 수행하는 과정에서 사용자는 해당 프로그램의 사용법에 의거하여 입력파일을 작성하게 된다. 이러한 방법으로 작성된 입력파일은 카드순서의 에러 및 그에 따른 포맷에러와 수치에러를 포함하게 된다. 사용자는 구조해석을 성공적으로 수행하기 위하여 해석프로그램에서 제공하는 데이터 검토 옵션에 의하여 입력파일의 올바른 작성을 체크하던지 또는 주의깊은 자신의 관찰에 의하여 이를 발견할 수밖에 없다. 이러한 검토과정에서도 그 에러를 찾아내기가 힘든 것이 수치크기의 옳고 그름이다. 즉, 카드의 잘못된 입력에 의하여 포맷에러를 피하더라도 입력되어진 수치의 크기는 다른 카드의 값을 읽어 들일 소지가 있으므로 수치의 정당성을 보장하기가 어렵다.

기존의 배치(Batch)양식으로 입력파일을 작성하였을 경우에 그 데이터의 타당성을 검토해보기 위한 방법으로서 앞에서 언급한 입력카드의 구조를 이용하여 작성된 입력파일을 순차적으로 읽어가면서 트리구조를 형성해보고, 데이터의 타입(정수형, 실수형 및 문자형등)과 데이터의 수치(값)을 내장된 지식으로 검토해볼 수 있다.

### 6.2.2 데이터 타입의 검토

각 카드에 따른 데이터를 사용자매뉴얼에서 제공하는 지식을 바탕으로 정해진 데이터 타입에 따른 내용을 표 6.2과 같이 정리하였다. 각 데이터 카드를 사용자가 제시하는 입력파일로부터 읽어들이는 데이터와 표 6.2에 의한 데이터 타입과를 비교함으로써 데이터타입의 적합함을 검토하고 에러가 발견되었을 경우에는 이를 사용자에게 제시한다.

### 6.2.3 데이터의 타당성 검토

표 6.1 BUILDS-A 입력카드의 구조

```

=====
; INPUT DATA FILE STRUCTURE OF BUILDS-A PROGRAM
=====
;   top      current  link      FORM
;CARD <cur> <equi> <sub> <name> <cur.no><tot.no> <name><cur.no><tot.no> FORM <var1> <var2> <var3>...)
;1.0 control data card
(CARD C1.0 C2.0 C1.1 master 1 1 control -1 1 FORM no)
(CARD C1.1 C1.2 C1.1.1 control 1 1 heading -1 1 FORM no)
(CARD C1.1.1 C1.1.2 no heading 1 1 job-name -1 1 FORM C1.1.1 ...)
(CARD C1.1.2 no no heading 1 1 user-name -1 1 FORM C1.1.1 ...)
(CARD C1.2 C1.3 no control 1 1 job-option -1 1 FORM C1.2 ...)
(CARD C1.3 C1.4 no control 1 1 input-ctrl -1 1 FORM C1.3 ...)
(CARD C1.4 no no control 1 1 input-ctrl -1 1 FORM C1.4 ...)

2.0 geometry topology card
(CARD C2.0 C3.0 C2.1 geometry 1 1 plan -1 1 FORM no)
(CARD C2.1 C2.2 no plan 1 1 plan-center -1 1 FORM C2.1 ...)
(CARD C2.2 C2.3 C2.2.1 plan 1 1 bay-width -1 1 FORM no)
(CARD C2.2.1 C2.2.2 no bay-width 1 1 x-dir -1 <NFX> FORM C2.2.1 ...)
(CARD C2.2.2 no no bay-width 1 1 y-dir -1 <NFY> FORM C2.2.1 ...)
(CARD C2.3 C2.4 C2.3.1 plan 1 1 frame-loc -1 1 FORM no)
(CARD C2.3.1 C2.3.2 no frame-loc 1 1 x-dir -1 <NFX> FORM C2.3.1 ...)
(CARD C2.3.2 no no frame-loc 1 1 y-dir -1 <NFY> FORM C2.3.1 ...)
(CARD C2.4 C2.5 C2.4.1 plan 1 1 column-line -1 <NFY> FORM no)
(CARD C2.4.1 no no column-line u <NFY> cl-loc -1 <NFX> FORM C2.4.1 ...)
(CARD C2.5 C2.6 C2.5.1 plan 1 1 beam-line -1 1 FORM no)
(CARD C2.5.1 C2.5.2 C2.5.1.1 beam-line 1 1 bl-locx -1 <NFY> FORM no)
(CARD C2.5.1.1 no no bl-locx u <NFY> bl-locy -1 <NFX> FORM C2.5.1.1 ...)
(CARD C2.5.2 no C2.5.2.1 beam-line 1 1 gl-locx -1 <NFY> FORM no)
(CARD C2.5.2.1 no no gl-locx u <NFY> gl-locy -1 <NFX> FORM C2.5.2.1 ...)
(CARD C2.6 C2.7 C2.6.1 plan 1 1 slab-line -1 <NFY> FORM no)
(CARD C2.6.1 no no slab-line u <NFY> sl-loc -1 <NFX> FORM C2.6.1 ...)
(CARD C2.7 C2.8 no plan 1 1 plotting -1 1 FORM C2.7 ...)
(CARD C2.8 C2.9 no plan 1 1 story-data -1 <NST> FORM C2.8 ...)
(CARD C2.9 no no plan 1 1 ext-stiff -1 <user> FORM C2.9 ...)

3.0 lateral forces
(CARD C3.0 C4.0 C3.1 load 1 1 lateral -1 1 FORM no)
(CARD C3.1 C3.2 no lateral 1 1 wind -1 1 FORM C3.1 ...)
(CARD C3.2 no no lateral 1 1 others -1 <user> FORM C3.2 ...)

4.0 member properties
(CARD C4.0 C5.0 C4.1 property 1 1 prop -1 1 FORM no)
(CARD C4.1 C4.2 no prop 1 1 material -1 <NMP> FORM C4.1 ...)
(CARD C4.2 C4.3 no prop 1 1 column-p -1 <NCP> FORM C4.2 ...)
(CARD C4.3 C4.4 no prop 1 1 beam-p -1 <NBP> FORM C4.3 ...)
(CARD C4.4 C4.5 no prop 1 1 slab-p -1 <NSP> FORM C4.4 ...)
(CARD C4.5 no no prop 1 1 truss-p -1 <NBRP> FORM C4.5 ...)

5.0 vertical composition of element line
(CARD C5.0 C6.0 C5.1 line-com 1 1 l-com -1 1 FORM no)
(CARD C5.1 C5.2 no l-com 1 1 c-line -1 <NCL> FORM C5.1 ...)
(CARD C5.2 C5.3 no l-com 1 1 b-line -1 <NBL> FORM C5.2 ...)

```

```

(CARD C5.3 no no i-com 1 1 s-line -1 <NSL> FORM C5.3 ...)

;6.0 initial strength of concrete
(CARD C6.0 C7.0 C6.1 concrete 1 1 initial -1 1 FORM no)
(CARD C6.1 C6.2 no initial 1 1 first -1 1 FORM C6.1 ...)
(CARD C6.2 no no initial 1 1 second -1 1 FORM C6.2 ...)

;7.0 substructure
(CARD C7.0 C8.0 no substructure 1 1 sub-str -1 <NSUB> FORM C7.0 ...)

;8.0 frame-location card
(CARD C8.0 C9.0 no geometry 1 1 frame-loc -1 <NTF> FORM C8.0 ...)

;9.0 frame card
(CARD C9.0 C10.0 C9.1 geometry 1 1 frame -1 <NDF> FORM no)
(CARD C9.1 C9.2 no frame u <NDF> control -1 1 FORM C9.1 ...)
(CARD C9.2 C9.3 no frame u <NDF> bay -1 <NC> FORM C9.2 ...)
(CARD C9.3 C9.4 no frame u <NDF> beam-load -1 <NFF> FORM C9.3 ...)
(CARD C9.4 C9.5 no frame u <NDF> panel -1 <NPAN> FORM C9.4 ...)
(CARD C9.5 C9.6 C9.5.1 frame u <NDF> blank -1 <NBLNK> FORM no)
(CARD C9.5.1 C9.5.2 no blank u <NBLNK> first -1 1 FORM C9.5.1 ...)
(CARD C9.5.2 no no blank u <NBLNK> first -1 <NCUT> FORM C9.5.2 ...)
(CARD C9.6 C9.7 no frame u <NDF> x-brace -1 <NBRAX> FORM C9.6 ...)
(CARD C9.7 no no frame u <NDF> k-brace -1 <NBRAK> FORM C9.7 ...)

;10.0 spectra or time history card
(CARD C10.0 C11.0 C710.1 loads 1 1 earth -1 1 FORM no)
; spectra data(if nat=3)
(CARD Ca10.1 Ca10.2 no earth 1 1 control -1 1 FORM Ca10.1 ...)
(CARD Ca10.2 no no earth 1 1 spectra -1 <NPC> FORM Ca10.2 ...)
; history data(if nat=4)
(CARD Cb10.1 Cb10.2 no earth 1 1 control -1 1 FORM Cb10.1 ...)
(CARD Cb10.2 Cb10.3/4/6 no earth 1 1 damping -1 <NFQ> FORM Cb10.2 ...)
(CARD Cb10.3 no no earth 1 1 history -1 <NPC> FORM Cb10.3 ...)
(CARD Cb10.4 Cb10.5 no earth 1 1 interval -1 1 FORM Cb10.4 ...)
(CARD Cb10.5 no no earth 1 1 history -1 <NPC/8> FORM Cb10.5 ...)
(CARD Cb10.6 no no earth 1 1 acceler -1 <NPC> FORM Cb10.6 ...)
; spectra data(if nat=5)
(CARD Cc10.1 Cc10.2 no earth 1 1 control -1 1 FORM Cc10.1 ...)
(CARD Cc10.2 Cc10.3/4/6 no earth 1 1 damping -1 <NFQ> FORM Cc10.2 ...)
(CARD Cc10.3 no no earth 1 1 history -1 <NPC> FORM Cc10.3 ...)
(CARD Cc10.4 Cc10.5 no earth 1 1 interval -1 1 FORM Cc10.4 ...)
(CARD Cc10.5 no no earth 1 1 history -1 <NPC/8> FORM Cc10.5 ...)
(CARD Cc10.6 no no earth 1 1 acceler -1 <NPC> FORM Cc10.6 ...)
; spectra data(if nat=6)
(CARD Cd10.1 Cd10.2 no earth 1 1 control -1 1 FORM Cd10.1 ...)
(CARD Cd10.2 no no earth 1 1 constants -1 <NFQ> FORM Cd10.2 ...)
; spectra data(if nat=7)
(CARD Ce10.1 Ce10.2 no earth 1 1 control -1 1 FORM Ce10.1 ...)
(CARD Ce10.2 Ce10.3 no earth 1 1 constants -1 <NFQ> FORM Ce10.2 ...)
(CARD Ce10.3 no no earth 1 1 mode -1 <NFQ> FORM Ce10.3 ...)

;11.0 load case card
(CARD C11.0 C12.0 no loads 1 1 load-case -1 <NLD> FORM C11.1 ...)

```

;12.0 plotting control card

```
(CARD  C11.0 C12.0 C12.1 plot  1      1 plotting -1    1  FORM no)
(CARD  C12.1 C12.2   no plotting 1      1 bldg-geo -1    1  FORM C12.1 ...)
(CARD  C12.2 no C12.2.1 no plotting 1      1 frm-geo  -1 <user>  FORM no)
(CARD  C12.2.1 C12.2.2 no frm-geo u <user> first  -1    1  FORM C12.2.1 ...)
(CARD  C12.2.2 C12.2.3 no frm-geo u <user> second -1    1  FORM C12.2.2 ...)
(CARD  C12.2.3   no no frm-geo u <user> second  -1    1  FORM C12.2.3 ...)
```

표 6.2 BUILDS-A 입력카드의 출력양식

```

.....
;*      BUILDS-A INPUT/OUTPUT DATA FORMAT FACT
.....
[no]
type
field
[END]
[C1.1.1]
s
72
[END]
[C1.2]
iiiiiiiiii
5 5 5 5 5 5 5 5 5 5
[END]
[C1.3]
iiiiiiiiiiii
5 5 5 5 5 5 5 5 5 5 5
[END]
[C1.4]
iiiiiiiiiii
5 5 5 5 5 5 5 5 5 5
[END]
[C2.1]
e e e e
10.3 10.3 10.3 10.3
[END]
[C2.2.1]
e e e e
10.3 10.3 10.3 10.3
[END]
[C2.3.1]
iiiiiiiiiiiiiiii
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[C2.4.1]
iiiiiiiiiiiiiiii
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[C2.5.1.1]
iiiiiiiiiiiiiiii
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[C2.5.2.1]
iiiiiiiiiiiiiiii
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[C2.6.1]
iiiiiiiiiiiiiiii
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[C2.7]
iii

```

```

5 5 5
[END]
[C2.8]
s x e e e e e
5 5 10.3 10.3 10.3 10.3
[END]
[C2.9]
i x e e .
5 5 10.3 10.3
[END]
[C3.1]
i i
5 5
[END]
[C3.2]
i i e e e e
5 5 10.3 10.3 10.3 10.3
[END]
[C4.1]
s e e e e
1 19.5 10.3 10.3 10.3
[END]
[C4.2]
i i s i e e e e e e e
1 4 1 4 10.3 10.3 10.3 10.3 10.3 10.3
[END]
[C4.3]
i i s i e e e e e
1 4 1 4 10.3 10.3 10.3 10.3
[END]
[C4.4]
i x e e e i i i i i
5 5 10.3 10.3 10.3 5 5 5 5 5 5
[END]
[C4.5]
i s i e
5 1 4 10.3
[END]
[CS.1]
i i i i i i i i i i i i i i
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[CS.2]
i i i i i i i i i i i i i i
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[CS.3]
i i i i i i i i i i i i i i
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[C6.1]
i i e
5 5 10.3
[END]
[C6.2]

```

```

i i e
5 5 10.3
[END]
[C7.0]
i i s
5 5 70
[END]
[C8.0]
i e e e e s
5 5 10.3 10.3 10.3 10.3 20
[END]
[C9.1]
i i i i i i i i i i s
5 5 5 5 5 5 5 5 5 5 5 20
[END]
[C9.2]
e e e e e e e e
10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3
[END]
[C9.3]
i i i i e e e e e
5 5 5 5 5 10.3 10.3 10.3 10.3 10.3
[END]
[C9.4]
i i i e e e e
5 5 5 5 10.3 10.3 10.3
[END]
[C9.5.1]
i i i i e e e e e
5 5 5 5 5 10.3 10.3 10.3 10.3 10.3
[END]
[C9.5.2]
e e e e e e e e
10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3
[END]
[C9.6]
i i i i i i
5 5 5 5 5 5
[END]
[C9.7]
i i i e e e e
5 5 5 5 10.3 10.3 10.3 10.3
[END]
[Ca10.1]
i e i i i s
5 10.3 5 5 5 40
[END]
[Ca10.2]
e e e e e e e e
10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3
[END]
[Ca10.1]
i i e e i i s
5 5 5 10.3 10.3 5 5 24
[END]

```

[Cb10.2]  
 e e e e e e e e  
 10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3  
 [END]  
 [Cb10.3]  
 i i i i e e  
 5 5 5 5 10.3 10.3  
 [END]  
 [Cb10.4]  
 e  
 10.3  
 [END]  
 [Cb10.5]  
 e e e e e e e e  
 10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3  
 [END]  
 [Cb10.6]  
 e e  
 10.3 10.3  
 [END]  
 [Ce10.1]  
 i i i e e i i s  
 5 5 5 10.3 10.3 5 5 24  
 [END]  
 [Ce10.2]  
 i x f f e e  
 5 5 5.2 5.2 10.3 10.3  
 [END]  
 [Ce10.3]  
 i i i i e e  
 5 5 5 5 10.3 10.3  
 [END]  
 [Ce10.4]  
 e  
 10.3  
 [END]  
 [Ce10.5]  
 e e e e e e e e  
 10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3  
 [END]  
 [Ce10.6]  
 e e  
 10.3 10.3  
 [END]  
 [Cd10.1]  
 i i i i i e e  
 5 5 5 5 5 10.3 10.3  
 [END]  
 [Cd10.2]  
 e e e e e  
 10.3 10.3 10.3 10.3 10.3  
 [END]  
 [Ce10.1]  
 i i i i e  
 5 5 5 5 5 10.3

```

[END]
[Ce10.2]
e e e e i i
10.3 10.3 10.3 10.3 10.3 5 5
[END]
[Ce10.3]
e e e e e e e
10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3
[END]
[C11.1]
f f f f f f f s
5.2 5.2 5.2 5.2 5.2 5.2 5.2 40
[END]
[C12.1]
i i i e e e e e
5 5 5 5 10.3 10.3 10.3 10.3 10.3
[END]
[C12.2.1]
i i i i i i i i e e e
5 5 5 5 5 5 5 5 5 10.3 10.3 10.3
[END]
[C12.2.2]
i i i i i i i i i i i i i i i i
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[END]
[C12.2.3]
i i i i e
5 5 5 5 5 10.3
[END]

```

데이터카드로부터 읽어들이는 데이터의 타입이 검토되고나면, 기존의 경험 및 일반적인 변수의 값에 의거하여 입력의 타당성을 검토하고, 에러가 있으면 그 결과를 제시한다.

#### 6.2.4 소결론

본 연구에서는 사용자가 작성한 데이터파일을 입력카드의 체계화된 트리구조를 이용하여 모형화함으로써 그 타당성을 검토하는 지원시스템을 개발하였다. 본 연구의 결과에 의해서는 입력데이터의 데이터타입과 수치의 적절함을 검토해볼 수 있다.

### 6.3 구조해석의 수행

6.1과 6.2에서 형성된 입력파일 또는 사용자가 직접 작성한 입력파일로 BUILDS-A에 의한 구조해석을 수행한다. 이 수치해석 과정에서 주요한 사실은 직접 기억공간에 추가되고, 부재단면의 개선을 위한 데이터로서 민감도 벡터가 계산된다.

### 6.4 해석결과의 검토

해석의 결과가 현재의 기억공간에 위치하게 되면 조건처리규칙에 의하여 건물의 안전성이 검토된다.

### 6.5 부재의 개선

부재의 개선이 요구될 경우에는 민감도 벡터를 기준으로 부재의 단면성질이 개선된다.

## 7. 예제의 실행

### 7.1 일반적인 수행예

본 절에서는 앞에서 언급된 건물의 예비설계에 관한 지식으로 컴퓨터의 실행결과를 보인다. 사용된 컴퓨터는 SUN 3/50으로 UNIX 운영체제이며, 구축용 도구는 K-CLIPS이다.

작업의 수행내용에 관한 설명부분은 ';'기호 다음에 나타내며, 사용자가 입력한 데이터를 표시하기 위하여 ".."로 표시한다. '<...>'부분은 프로그램에 의하여 자동적으로 제시되는 데이터에 대한 값이다.

; unix상의 k-clips의 실행

```
strsys2% "k-clips"
```

```
*****
*
*   Welcome to KAIST C Language Intergated Production System   *
*
*                               version 4.1                       *
*
*   KAIST   Dept. of Civil Eng.                                *
*
*   Structural System Lab.                                     *
*
*****
```

; 제어규칙의 로딩

```
K-CLIPS> "(load "control.rul")"
```

```
***** ; '*'는 규칙하나를 의미한다.
```

```
K-CLIPS> "(load "control.fct")"
```

```
$$$$$$$$$$
```

; 예비설계규칙의 로딩

```
K-CLIPS> "(load "prelimin.rul")"
```

```
*****
*****
*****
```

```

K-CLIPS> "(load "prelimin.fct")"
$
; 모델링 규칙의 로딩
K-CLIPS> "(load "idealize.rul")"
*****
; 해석부분에 관한 규칙의 로딩
K-CLIPS> "(load "builds-a.rul")"
*****
; working memory 초기화
K-CLIPS> "(reset)"
; 실행
K-CLIPS> "(run)"
; 타이틀 화면
** The start of help TITLE **

```

Welcome to I-BUILDS !!!

I-BUILDS ... An Intelligent BUILDing Design System

Developed by

C.K. Choi and E.D. Kim  
Department of Civil Engineering  
KAIST, Seoul, KOREA

January 1990

Notice!!! Units;KMS(Kg,cm,sec)

\*\* The end of help TITLE \*\*

Please, press <Return> Key to continue.

; 작업 메뉴의 제시

The menus of K-CLIPS are shown below. Which one do you want to select?

- 0] Exit to K-CLIPS
- 1] Planning
- 2] PRELIMINARY-DESIGN
- 3] DETAILED-DESIGN
- 4] EVALUATION

5] PRESENTATION

Otherwise, phase 'help' will be supplied.

Enter your selection : "1"

; 계획단계의 선택

The menus of Planning are shown below. Which one do you want to select?

- 0] Exit to Planning
- 1] list
- 2] modify
- 3] load
- 4] save
- 5] constraint

Otherwise, phase 'help' will be supplied.

; 계획에 관련한 데이터의 입력

Enter your selection<0> : "3"

\*\* The start of load Planning \*\*

Enter filename to load<planning.dat>"<CR>"

planning.dat is opened.

planning.dat is being loaded.

\*\* The end of load Planning \*\*

Please, press <Return> Key to continue.

; 데이터 리스트

The menus of Planning are shown below. Which one do you want to select?

- 0] Exit to Planning
- 1] list
- 2] modify
- 3] load
- 4] save
- 5] constraint

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

\*\* The start of list Planning \*\*

/ID/CONTENT/  
/job/nil/  
/user/nil/  
/occupancy/commercial/  
/material/Steel/  
/code/tmp/

/Zgrid/nil nil/  
/Xgrid/nil nil/  
/Ygrid/nil nil/  
/xcore/nil/  
/ycore/nil/  
/Vload/nil nil/  
/Wload/nil/  
/Eload/nil/  
/foundation/nil/  
/water-level/nil/  
/Fe/nil/

**\*\* The end of list Planning \*\***

Please, press <Return> Key to continue.

; 데이터의 입력

The menus of Planning are shown below. Which one do you want to select?

- 0] Exit to Planning
- 1] list
- 2] modify
- 3] load
- 4] save
- 5] constraint

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "2"

**\*\* The start of modify Planning \*\***

Enter Planning ID/NO Value/list to modify : "job"

ENTER Planning[job] CONTENT<nil>: "This is a test execution"

**\*\* The end of modify Planning \*\***

Please, press <Return> Key to continue.

; 데이터의 검토

The menus of Planning are shown below. Which one do you want to select?

- 0] Exit to Planning
- 1] list
- 2] modify
- 3] load
- 4] save
- 5] constraint

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "5"

; 데이터 검토의 결과 제시

\*\* The start of constraint Planning \*\*

???)> CONSTRAINT AMONG ERROR :

Planning[code] CONTENT<ddd> must be .AMONG./WSD/USD/]

\*\* The end of constraint Planning \*\*

Please, press <Return> Key to continue.

; 상위레벨로의 이동

The menus of Planning are shown below. Which one do you want to select?

0] Exit to Planning

1] list

2] modify

3] load

4] save

5] constraint

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

planning.dat is closed.

; 예비설계단계의 선택

The menus of K-CLIPS are shown below. Which one do you want to select?

0] Exit to K-CLIPS

1] Planning

2] PRELIMINARY-DESIGN

3] DETAILED-DESIGN

4] EVALUATION

5] PRESENTATION

Otherwise, phase 'help' will be supplied.

Enter your selection : "2"

; 사용자가 선택한 설계기준에 따른 하중경우의 자동입력

Default Load-Case is asserted.

; 예비설계에 관련한 서브메뉴

The menus of PRELIMINARY-DESIGN are shown below.

Which one do you want to select?

0] Exit to PRELIMINARY-DESIGN

1] COORDINATE

2] MATERIAL

- 3] SECTION
- 4] Load-Case
- 5] BUILDING

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "<CR>"

; 세부설계의 선택

The menus of K-CLIPS are shown below. Which one do you want to select?

- 0] Exit to K-CLIPS
- 1] Planning
- 2] PRELIMINARY-DESIGN
- 3] DETAILED-DESIGN
- 4] EVALUATION
- 5] PRESENTATION

Otherwise, phase 'help' will be supplied.  
Enter your selection : "3"

; 세부설계의 서브메뉴

The menus of DETAILED-DESIGN are shown below.  
Which one do you want to select?

- 0] Exit to DETAILED-DESIGN
- 1] IDEALIZED-MODEL
- 2] STRUCTURAL-ANALYSIS
- 3] STEEL-MEMBER-DESIGN
- 4] CONCRETE-MEMBER-DESIGN
- 5] FOUNDATION-DESIGN

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "<CR>"

; 평가단계의 선택

The menus of K-CLIPS are shown below. Which one do you want to select?

- 0] Exit to K-CLIPS
- 1] Planning
- 2] PRELIMINARY-DESIGN
- 3] DETAILED-DESIGN
- 4] EVALUATION
- 5] PRESENTATION

Otherwise, phase 'help' will be supplied.  
Enter your selection : "4"

; 평가단계의 서브메뉴

The menus of EVALUATION are shown below.  
Which one do you want to select?

- 0] Exit to EVALUATION

1] COST-ESTIMATION  
2] SUGGEST-ALTERNATE-DESIGN  
3] SELECT-BEST-DESIGN  
Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "<CR>"

; 결과정리 부분의 선택

The menus of K-CLIPS are shown below. Which one do you want to select?  
0] Exit to K-CLIPS  
1] Planning  
2] PRELIMINARY-DESIGN  
3] DETAILED-DESIGN  
4] EVALUATION  
5] PRESENTATION  
Otherwise, phase 'help' will be supplied.  
Enter your selection : "5"

; 서브메뉴

The menus of PRESENTATION are shown below.  
Which one do you want to select?  
0] Exit to PRESENTATION  
1] DRAFTING  
2] REPORT  
Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "<CR>"

; 작업의 끝

The menus of K-CLIPS are shown below. Which one do you want to select?  
0] Exit to K-CLIPS  
1] Planning  
2] PRELIMINARY-DESIGN  
3] DETAILED-DESIGN  
4] EVALUATION  
5] PRESENTATION  
Otherwise, phase 'help' will be supplied.  
Enter your selection : "0"

; 적용된 규칙의 수  
92 rules fired

; unix 상태로 회복  
K-CLIPS> "(exit)"

## 7.2 예비설계의 예

### 7.2.1 실행 예

예제의 대상 건물의 개요는 다음과 같다.

- 건물의 용도 : 상업용
- 건물의 구조 : 철골구조
- 설계방법 : 허용응력법
- 건물의 규모
  - 층수 : 30 층
  - 건물고 : 90 m
  - 폭 방향의 스패 수 및 크기 : 5개, 36.0m
  - 길이방향 : 4개, 19.8m
  - 코어 : 폭방향 3-4번째, 길이방향 그리드의 2-3번째
- 하중(단위, kg/m<sup>2</sup>)
  - 고정하중 : 500
  - 적재하중 : 300
  - 풍하중 : 200
- 구조재료의 성질(단위, kg/cm<sup>2</sup>)
  - 항복강도 : 4000

위와같은 건물의 데이터의 입력, 재생된 건물의 형상(프레임의 종류, 배치 및 구성), 그리고 각 후보에 대한 분석과정은 다음과 같이 진행된다.

; 규칙의 로딩과정

```
K-CLIPS> (load "control.rul")
```

```
*****
```

```
K-CLIPS> (load "control.fct")
```

```
$$$$$$$$
```

```
K-CLIPS> (load "prelimin.rul")
```

```
*****
```

```
*****
```

```
K-CLIPS> (load "prelimin.fct")
```

```
$
```

```
K-CLIPS> (load "idealize.rul")
```

```
*****
```

```
K-CLIPS> (load "planning.fct")
```

```
$
```

```
K-CLIPS> (reset)
```

```
K-CLIPS> (run)
```

```
; title 제시(생략)
```

The menus of K-CLIPS are shown below. Which one do you want to select?

- 0] Exit to K-CLIPS
- 1] Planning
- 2] PRELIMINARY-DESIGN
- 3] DETAILED-DESIGN

4] EVALUATION

5] PRESENTATION

Otherwise, phase 'help' will be supplied.

Enter your selection : 1

; 건물의 개요에 대한 데이터를 입력하기 위하여 계획단계 선정

; 입력과정은 생략

The menus of Planning are shown below. Which one do you want to select?

0] Exit to Planning

1] list

2] modify

3] load

4] save

5] constraint

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 1

; 입력결과제시

\*\* The start of list Planning \*\*

/ID/CONTENT/

/job/nil/

/user/nil/

/Eload/nil/

/foundation/nil/

/water-level/nil/

/Fe/nil/

/occupancy/commercial/

/code/WSD/

/material/Steel/

/Zgrid/30 9000/

/Ygrid/3 1980/

/Xgrid/5 3600/

/ycore/2 3/

/xcore/3 4/

/Wload/0.02/

/Vload/0.05 0.03/

\*\* The end of list Planning \*\*

Please, press <Return> Key to continue.

; 상위레벨로의 이동, 예비설계단계의 선택

The menus of K-CLIPS are shown below. Which one do you want to select?

0] Exit to K-CLIPS

1] Planning

2] PRELIMINARY-DESIGN

3] DETAILED-DESIGN

4] EVALUATION

5] PRESENTATION

Otherwise, phase 'help' will be supplied.

Enter your selection : 2

; 그리드 좌표계의 입력

The menus of PRELIMINARY-DESIGN are shown below. Which one do you want to select?

- 0] Exit to PRELIMINARY-DESIGN
- 1] COORDINATE
- 2] MATERIAL
- 3] SECTION
- 4] Load-Case
- 5] BUILDING

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 1

The menus of COORDINATE are shown below. Which one do you want to select?

- 0] Exit to COORDINATE
- 1] Xgrid
- 2] Ygrid
- 3] Zgrid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 1

The menus of Xgrid are shown below. Which one do you want to select?

- 0] Exit to Xgrid
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 7

\*\* The start of grid Xgrid \*\*

[Ques.] Enter data of Xgrid direction.

- (a) From Grid No. :0
- (b) To Grid No. :5
- (c) Span Magnitude :720

; 동일간격의 그리드 재생

Xgrid[1] is asserted with 720.

Xgrid[2] is asserted with 1440.

Xgrid[3] is asserted with 2160.

Xgrid[4] is asserted with 2880.

Xgrid[5] is asserted with 3600.

**\*\* The end of grid Xgrid \*\***

Please, press <Return> Key to continue.

; 다른방향에 대해서도 동일함(생략)

; 부재단면의 데이터베이스의 로딩

The menus of PRELIMINARY-DESIGN are shown below. Which one do you want to select?

- 0] Exit to PRELIMINARY-DESIGN
- 1] COORDINATE
- 2] MATERIAL
- 3] SECTION
- 4] Load-Case
- 5] BUILDING

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 3

; 철골보단면

The menus of SECTION are shown below. Which one do you want to select?

- 0] Exit to SECTION
- 1] Slab-CON
- 2] Beam-STL
- 3] Beam-CON
- 4] Column-STL
- 5] Column-CON
- 6] Wall-CON

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 2

The menus of Beam-STL are shown below. Which one do you want to select?

- 0] Exit to Beam-STL
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 5

**\*\* The start of load Beam-STL \*\***

Enter filename to load<beam-stl.dat>

beam-stl.dat is opened.

beam-stl.dat is being loaded.

**\*\* The end of load Beam-STL \*\***

Please, press <Return> Key to continue.

; 다른 부재에 대해서도 동일(생략)

; 예비설계의 수행

The menus of PRELIMINARY-DESIGN are shown below. Which one do you want to select?

- 0] Exit to PRELIMINARY-DESIGN
- 1] COORDINATE
- 2] MATERIAL
- 3] SECTION
- 4] Load-Case
- 5] BUILDING

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 5

The menus of BUILDING are shown below. Which one do you want to select?

- 0] Exit to BUILDING
- 1] generate
- 2] Candidate
- 3] Evaluation
- 4] Factor

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : 1

; 건물형상의 재생 및 분석, 평가과정의 진행

\*\* The start of generate BUILDING \*\*

; 각 레벨에서의 후보재생

Frame:Ygrid:Wall is not your Steel.

Frame:Xgrid:Wall is not your Steel.

Candidate Layout : Xgrid : none

Candidate Layout : Xgrid : all

Candidate Layout : Ygrid : none

Candidate Layout : Ygrid : all

Candidate Layout : Ygrid : user : 2

Candidate Compose : Xgrid : user : 1

Candidate Layout : Xgrid : user : 2

Candidate Compose : Ygrid : user : 1

Candidate Compose : Xgrid : none : 0

Candidate Compose : Xgrid : all : 3

Candidate Compose : Xgrid : ext : 1

Candidate Compose : Xgrid : int : 1

Compose:Xgrid:int:1 is same as :user:.

Candidate Compose : Ygrid : none : 0

Candidate Compose : Ygrid : all : 5

Candidate Compose : Ygrid : ext : 1

Candidate Compose : Ygrid : ext : 2

Candidate Compose : Ygrid : int : 1

Compose:Ygrid:int:1 is same as :user:.  
 Candidate Compose : Ygrid : int : 3  
 Candidate Material : Steel,RC,Composite  
 Material:Composite is not yours.  
 Material:RC is not yours.  
 Candidate Floor : RC-slab,PC-slab,RC-on-stl-deck,PS-con  
 Candidate Frame : Xgrid : Rigid, Brace, Wall  
 Candidate Frame : Ygrid : Rigid, Brace, Wall  
 Frame:Ygrid:Wall is not your Steel.  
 Frame:Xgrid:Wall is not your Steel.  
 Frame:Xgrid:Wall is not your Steel.  
 Frame:Ygrid:Wall is not your Steel.  
 Candidate Layout : Xgrid : ext : 1  
 Candidate Layout : Xgrid : ext : 2  
 Candidate Layout : Xgrid : int : 2  
 Layout:Xgrid:int:2 is same as :user:.  
 Candidate Layout : Xgrid : int : 4  
 Candidate Layout : Ygrid : ext : 1  
 Candidate Layout : Ygrid : int : 2  
 Layout:Ygrid:int:2 is same as :user:.  
 Layout:Xgrid:int:4 is too more reinforced.

; 각 레벨후보의 조합

Now, The Candidates of buildings are generated!!!

; 후보의 제시는 가장 적절한 순서로 제시함

=====

BUILDING CANDIDATE PROFILE[424]

=====

1) Material : Steel  
 2) Floor : RC-slab  
 On Xgrid,  
 3) Frame : Brace  
 4) Layout : CLASS:ext,RANGE:1:: F nil nil nil nil F  
 5) Compose : CLASS:ext,RANGE:1::1-2 3-4  
 On Ygrid,  
 6) Frame : Brace  
 7) Layout : CLASS:user,RANGE:2:: nil F F nil  
 8) Compose : CLASS:user,RANGE:1::3-4  
 Do you want to analyze this building?[Y/N]:y

; 선택을 하게되면 해당 후보에 대해 분석 및 평가과정을 수행  
 ; 추론과정의 설명을 위하여 규칙의 적용과 사실의 변동을 보임.  
 ; 내용 : 부재의 그룹핑을 설정함.

==> f-153 (Candidate gen1 analysis g1 g2 g3 g4 g5 g6 g7 g8)  
 FIRE 5 Slabs-Grouping: f-153 f-114 f-118 f-149

Slabs[1]:Bi,0-5,0-3,1-30  
 ==> f-154 (Slabs 1 Bi 0 5 0 3 1 30)  
 FIRE 6 Columns-Grouping: f-153 f-114 f-118 f-149  
 Columns[1]:Bi,0-5,0-3,0-5  
 ==> f-155 (Columns 1 Bi 0 5 0 3 0 5)  
 Columns[2]:Bi,0-5,0-3,5-10  
 ==> f-156 (Columns 2 Bi 0 5 0 3 5 10)  
 Columns[3]:Bi,0-5,0-3,10-15  
 ==> f-157 (Columns 3 Bi 0 5 0 3 10 15)  
 Columns[4]:Bi,0-5,0-3,15-20  
 ==> f-158 (Columns 4 Bi 0 5 0 3 15 20)  
 Columns[5]:Bi,0-5,0-3,20-25  
 ==> f-159 (Columns 5 Bi 0 5 0 3 20 25)  
 Columns[6]:Bi,0-5,0-3,25-29  
 ==> f-160 (Columns 6 Bi 0 5 0 3 25 29)  
 Columns[7]:Bi,0-5,0-3,29-30  
 ==> f-161 (Columns 7 Bi 0 5 0 3 29 30)

; 건물형상을 구성하는 2-D 프레임을 재생하면서, 해당프레임을  
 ; 구성하는 요소(슬라브, 보, 기둥등)의 그룹 및 개별요소를  
 ; 재생함.

FIRE 7 Floor-Generation: f-153 f-114 f-118 f-149  
 ==> f-162 (Floor 1 Bi 0 5 0 3 1 30)  
 FIRE 8 Element-Constraint-r3: f-162  
 ==> f-163 (Slab gen2 Bi 0 1 0 1 1 30 nil nil nil nil nil nil DESIGN ...)  
 ==> f-164 (Slab gen3 Bi 0 1 1 2 1 30 nil nil nil nil nil nil DESIGN ...)  
 ==> f-165 (Slab gen4 Bi 0 1 2 3 1 30 nil nil nil nil nil nil DESIGN ...)

---- 이하 생략

==> f-175 (Slab gen14 Bi 4 5 0 1 1 30 nil nil nil nil nil nil DESIGN ...)  
 ==> f-176 (Slab gen15 Bi 4 5 1 2 1 30 nil nil nil nil nil nil DESIGN ...)  
 ==> f-177 (Slab gen16 Bi 4 5 2 3 1 30 nil nil nil nil nil nil DESIGN ...)  
 FIRE 9 Beams-Grouping: f-153 f-4 f-114 f-118 f-149  
 Beams[1]:Xgrid,0-5,0-3,1-29  
 ==> f-178 (Beams 1 Xgrid 0 5 0 3 1 29)  
 Beams[2]:Xgrid,0-5,0-3,30-30  
 ==> f-179 (Beams 2 Xgrid 0 5 0 3 30 30)  
 FIRE 10 Frame-NO-Generation: f-153 f-4 f-114 f-118 f-149  
 ==> f-180 (Frame-NO 1 Xgrid 0 0 0 3 0 30 nil)  
 ==> f-181 (Frame-NO 2 Xgrid 1 1 0 3 0 30 nil)  
 ==> f-182 (Frame-NO 3 Xgrid 2 2 0 3 0 30 nil)  
 ==> f-183 (Frame-NO 4 Xgrid 3 3 0 3 0 30 nil)  
 ==> f-184 (Frame-NO 5 Xgrid 4 4 0 3 0 30 nil)  
 ==> f-185 (Frame-NO 6 Xgrid 5 5 0 3 0 30 nil)  
 FIRE 11 Frame-ID-Generation: f-153 f-4 f-5 f-6 f-114 f-118 f-149

```

==> f-186 (Frame-ID 1 Xgrid 1 1 0 3 0 30 nil nil Rigid)
==> f-187 (Frame-ID 2 Xgrid 0 0 0 3 0 30 nil nil Brace 1 2 3 4)
FIRE 12 Frame-Generation: f-153 f-4 f-5 f-6 f-180 f-187
<== f-180 (Frame-NO 1 Xgrid 0 0 0 3 0 30 nil)
==> f-188 (Frame-NO 1 Xgrid 0 0 0 3 0 30 2)
FIRE 13 Element-Constraint-r5: f-188 f-187
==> f-189 (Beam gen17 Xgrid 0 0 0 1 30 30 nil nil nil nil nil nil DESIGN ...)
==> f-190 (Beam gen18 Xgrid 0 0 1 2 30 30 nil nil nil nil nil nil DESIGN ...)
==> f-191 (Beam gen19 Xgrid 0 0 2 3 30 30 nil nil nil nil nil nil DESIGN ...)

```

--- 이하생략

```

==> f-248 (Braces 1 Xgrid 0 5 0 3 0 30)
FIRE 42 Braces_Walls-Grouping: f-153 f-7 f-114 f-118 f-149
==> f-249 (Braces 1 Ygrid 0 5 0 3 0 30)

```

; 재생된 요소의 재료성질을 정의함.

```

FIRE 43 Element-Material: f-246 f-153 f-2
<== f-246 (Brace gen58 Ygrid 2 3 2 2 0 30 nil nil nil nil DESIGN ...)
Brace[gen58:Ygrid,2-3,2-2,0-30] is set to material[1,0].
==> f-250 (Brace gen58 Ygrid 2 3 2 2 0 30 1 nil nil nil DESIGN ...)
-----

```

```

FIRE 44 Element-Material: f-244 f-153 f-2
<== f-244 (Column gen56 Ygrid 4 4 2 2 0 30 nil nil nil nil DESIGN ...)
Column[gen56:Ygrid,4-4,2-2,0-30] is set to material[1,0].
==> f-251 (Column gen56 Ygrid 4 4 2 2 0 30 1 nil nil nil DESIGN ...)
-----

```

```

FIRE 48 Element-Material: f-239 f-153 f-2
<== f-239 (Beam gen51 Ygrid 4 5 2 2 30 30 nil nil nil nil nil nil DESIGN ...)
Beam[gen51:Ygrid,4-5,2-2,30-30] is set to material[1,0].
==> f-255 (Beam gen51 Ygrid 4 5 2 2 30 30 1 nil nil nil nil nil nil DESIGN ...)

```

; 각 요소의 부재력, 단면을 설정하고 평가과정을 수행.  
; 다른 건물형상에 대해 반복.

```

=====
BUILDING CANDIDATE PROFILE[424]
=====

```

- 1) Material : Steel
- 2) Floor : RC-slab  
On Xgrid,
- 3) Frame : Brace
- 4) Layout : CLASS:ext,RANGE:1:: F nil nil nil nil F
- 5) Compose : CLASS:all,RANGE:3::1-4  
On Ygrid,

6) Frame : Brace  
7) Layout : CLASS:user,RANGE:2:: nil F F nil  
8) Compose : CLASS:user,RANGE:1::3-4  
Do you want to analyze this building?[Y/N]:n

; 이하생략  
; 건물후보의 분석이 끝남

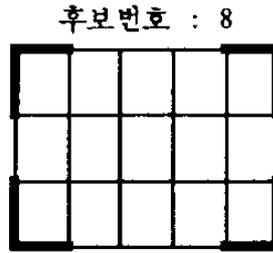
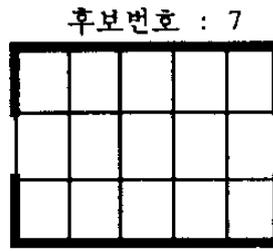
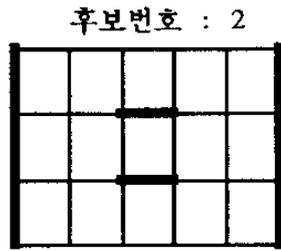
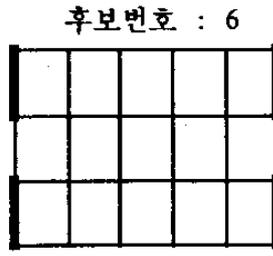
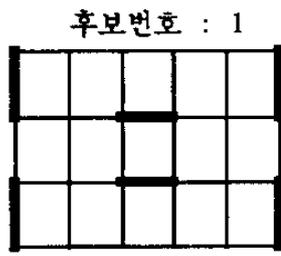
**\*\* The end of generate BUILDING \*\***  
Please, press <Return> Key to continue.

; 2,3의 선택으로 후보들의 구성 및 평가치를 리스트함.  
The menus of BUILDING are shown below. Which one do you want to select?  
0] Exit to BUILDING  
1] generate  
2] Candidate  
3] Evaluation  
4] Factor  
Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : 0

; 상위레벨로의 이동  
; 이하생략  
; 작업의 수행완료

### 7.2.1 결과토론

본 예제에서 입력한 건물에 대하여 건물의 형상후보는 그림 7.1과 같이 43종류가 생성되었다. 후보의 순서는 개략적인 평가에 의하여 가능성이 높은 순서로 제시되는데 사용자가 선택을 하게되면 자세한 평가가 수행된다. 그리고 제시되는 평가치는 분석요인의 종류 및 중요도에 따라 다를 수도 있으므로 본 연구에서는 이를 극복하기 위하여 각 요인별로 적절한 가중치를 사용하였으며 이는 사용자가 직접 입력할 수 있다. 예를들면, 수평변위, 건설속도, 코어링정도, 내부공간상태에 대해서는 5.0, 보 깊이와 기둥의 면적에 대해서는 3.0, 그리고 프레임 및 기둥의 응력상태에 대해서는 1.0으로 하였다.



●  
●  
●  
●

이하생략 (후보 9 ~ 후보 43)

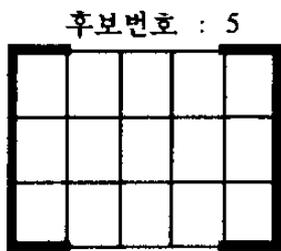


그림 7.1 건물 후보 형상

## 7.3 구조물의 이상화

### 7.3.1 실행예

예비설계단계를 수행하지 않은 상태에서 사용자가 직접 해석데이터를 입력하기 위해서는 이 모듈을 사용하여야 한다. 구조해석과정과 관련한 데이터의 입력과정을 설명하기 위하여 본 연구에서는 그림 6.2 모델링의 예제건물을 대상으로 한다. 각 과정의 설명은 다음과 같다.

```
strsys2% "k-clips":
```

```
; 타이틀 표시 생략
```

```
; 배치파일의 실행
```

```
K-CLIPS> "(batch "lll")"
```

```
1
```

```
K-CLIPS> (load "control.rul")
```

```
*****
```

```
K-CLIPS> (load "control.fct")
```

```
$$$$$$$$$$
```

```
K-CLIPS> (load "idealize.rul")
```

```
*****
```

```
K-CLIPS> (reset)
```

```
K-CLIPS> (run)
```

```
; 타이틀 표시 생략
```

```
; 세부설계단계의 선택
```

```
The menus of K-CLIPS are shown below. Which one do you want to select?
```

```
0] Exit to K-CLIPS
```

```
1] Planning
```

```
2] PRELIMINARY-DESIGN
```

```
3] DETAILED-DESIGN
```

```
4] EVALUATION
```

```
5] PRESENTATION
```

```
Otherwise, phase 'help' will be supplied.
```

```
Enter your selection : "3"
```

```
; 모델링 부분 선택
```

The menus of DETAILED-DESIGN are shown below.

Which one do you want to select?

- 0] Exit to DETAILED-DESIGN
- 1] IDEALIZED-MODEL
- 2] STRUCTURAL-ANALYSIS
- 3] STEEL-MEMBER-DESIGN
- 4] CONCRETE-MEMBER-DESIGN
- 5] FOUNDATION-DESIGN

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

; '도움' 기능의 선택

The menus of IDEALIZED-MODEL are shown below.

Which one do you want to select?

- 0] Exit to IDEALIZED-MODEL
- 1] COORDINATE
- 2] MATERIAL
- 3] SECTION
- 4] LOAD
- 5] GEOMETRY
- 6] Load-Case

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "?"

; 설명 부분

\*\* The start of help IDEALIZED-MODEL \*\*

In this module, You can design building in terms of  
COORDINATE, MATERIAL, SECTION, LOAD, Load-Case and  
ELEMENT, ELEMENTS(ELEMENT GROUPING)

Notice!!! Units;KMS(Kg,cm,sec)Xgrid

\*\* The end of help IDEALIZED-MODEL \*\*

Please, press <Return> Key to continue.

; 그리드 좌표계 선택

The menus of IDEALIZED-MODEL are shown below.

Which one do you want to select?

- 0] Exit to IDEALIZED-MODEL
- 1] COORDINATE
- 2] MATERIAL
- 3] SECTION
- 4] LOAD
- 5] GEOMETRY
- 6] Load-Case

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "1"

The menus of COORDINATE are shown below. Which one do you want to select?

- 0] Exit to COORDINATE
- 1] Xgrid
- 2] Ygrid
- 3] Zgrid

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "?"

\*\* The start of help COORDINATE \*\*

You can design building geometry in this module.  
This menu serves you to define your coordinate system  
which will be used to describe element connectivity,  
and building loads.

\*\* The end of help COORDINATE \*\*

Please, press <Return> Key to continue.

The menus of COORDINATE are shown below. Which one do you want to select?

- 0] Exit to COORDINATE
- 1] Xgrid
- 2] Ygrid
- 3] Zgrid

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "1"

; 등간격 그리드 좌표계의 일괄처리

The menus of Xgrid are shown below. Which one do you want to select?

- 0] Exit to Xgrid
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "7"

\*\* The start of grid Xgrid \*\*

[Ques.] Enter data of Xgrid direction.

(a) From Grid No. : "0"

- (b) To Grid No. : "2"
- (c) Span Magnitude : "600"

Xgrid[1] is asserted with 600.  
Xgrid[2] is asserted with 1200.

**\*\* The end of grid Xgrid \*\***

Please, press <Return> Key to continue.

; 입력의 확인

The menus of Xgrid are shown below. Which one do you want to select?

- 0] Exit to Xgrid
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

**\*\* The start of list Xgrid \*\***

/NO/COORDINATE/  
/0/0/  
/1/600/  
/2/1200/

**\*\* The end of list Xgrid \*\***

Please, press <Return> Key to continue.

; 상위 레벨로의 이동

The menus of Xgrid are shown below. Which one do you want to select?

- 0] Exit to Xgrid
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

The menus of COORDINATE are shown below. Which one do you want to select?

- 0] Exit to COORDINATE
- 1] Xgrid
- 2] Ygrid
- 3] Zgrid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "2"

; 데이터의 입력

The menus of Ygrid are shown below. Which one do you want to select?

- 0] Exit to Ygrid
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "2"

\*\* The start of insert Ygrid \*\*

Enter Ygrid ID/NO Value/list to insert : "1"

ENTER Ygrid[1] COORDINATE<nil>: "800"

\*\* The end of insert Ygrid \*\*

Please, press <Return> Key to continue.

The menus of Ygrid are shown below. Which one do you want to select?

- 0] Exit to Ygrid
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

The menus of COORDINATE are shown below.

Which one do you want to select?

- 0] Exit to COORDINATE

- 1] Xgrid
- 2] Ygrid
- 3] Zgrid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "3"

The menus of Zgrid are shown below. Which one do you want to select?

- 0] Exit to Zgrid
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "2"

\*\* The start of insert Zgrid \*\*

Enter Zgrid ID/NO Value/list to insert : "1"  
ENTER Zgrid[1] COORDINATE<nil>: "300"

\*\* The end of insert Zgrid \*\*

Please, press <Return> Key to continue.

The menus of Zgrid are shown below. Which one do you want to select?

- 0] Exit to Zgrid
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

The menus of COORDINATE are shown below.

Which one do you want to select?

- 0] Exit to COORDINATE
- 1] Xgrid
- 2] Ygrid
- 3] Zgrid

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

The menus of IDEALIZED-MODEL are shown below.

Which one do you want to select?

- 0] Exit to IDEALIZED-MODEL
- 1] COORDINATE
- 2] MATERIAL
- 3] SECTION
- 4] LOAD
- 5] GEOMETRY
- 6] Load-Case

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "2"

; 재료성질의 입력

The menus of MATERIAL are shown below. Which one do you want to select?

- 0] Exit to MATERIAL
- 1] Steel
- 2] Concrete

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

; 내장된 재료성질

The menus of Steel are shown below. Which one do you want to select?

- 0] Exit to Steel
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

\*\* The start of list Steel \*\*

/NO/ELASTIC/RHO/POISSON/YIELD/

/1/2.1000e+06/7.5/0.30000001/3000/

\*\* The end of list Steel \*\*

Please, press <Return> Key to continue.

The menus of Steel are shown below. Which one do you want to select?

- 0] Exit to Steel
- 1] list
- 2] insert
- 3] delete

- 4] modify
- 5] load
- 6] save

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

The menus of MATERIAL are shown below. Which one do you want to select?

- 0] Exit to MATERIAL
- 1] Steel
- 2] Concrete

Otherwise, phase 'help' will be supplied.

Enter your selection<0> :

**; 건물의 형상입력**

The menus of IDEALIZED-MODEL are shown below.

Which one do you want to select?

- 0] Exit to IDEALIZED-MODEL
- 1] COORDINATE
- 2] MATERIAL
- 3] SECTION
- 4] LOAD
- 5] GEOMETRY
- 6] Load-Case

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "5"

**; 요소단위로 입력**

The menus of GEOMETRY are shown below. Which one do you want to select?

- 0] Exit to GEOMETRY
- 1] FRAME
- 2] ELEMENT
- 3] GROUPING

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "2"

**; 슬라브 요소의 입력**

The menus of ELEMENT are shown below. Which one do you want to select?

- 0] Exit to ELEMENT
- 1] Slab
- 2] Beam
- 3] Column
- 4] Brace
- 5] Wall

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

The menus of Slab are shown below. Which one do you want to select?

- 0] Exit to Slab
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-
- 8] assign

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "2"

\*\* The start of insert Slab \*\*

; 그리드좌표계에 의한 다중 입력

```
Enter Slab ID/NO Value/list to insert : "temp"
ENTER Slab[temp] DIR<nil>: "Bi"
ENTER Slab[temp] Xgrid<nil>: "0"
ENTER Slab[temp] Xgrid<nil>: "2"
ENTER Slab[temp] Ygrid<nil>: "0"
ENTER Slab[temp] Ygrid<nil>: "1"
ENTER Slab[temp] Zgrid<nil>: "1"
ENTER Slab[temp] Zgrid<nil>: "1"
ENTER Slab[temp] Steel<nil>:"<CR>"
ENTER Slab[temp] Concrete<nil>:"<CR>"
ENTER Slab[temp] Slab-STL<nil>:"<CR>"
ENTER Slab[temp] Slab-CON<nil>:"<CR>"
ENTER Slab[temp] Slab-DL<nil>: "1" ; 정의되지 않은 데이터임
ENTER Slab[temp] Slab-LL<nil>: "1" ; 같음.
ENTER Slab[temp] FORCE<nil>:"<CR>"
```

; 각 요소별로 그리드 좌표계로 자동분리됨  
Slab[temp] has multi-dimensional grids. It is divided.

; 미지의 하중데이터가 입력되었음을 통고하고  
; 사용자에게 입력을 요함

???)> INDEXING ERROR :

Slab[gen2] has undefined Slab-LL[1].

You must

1. delete Slab[gen2] or
2. insert Slab-LL[1].

...Which one do you want?[1/2] : "2"

ENTER Slab-LL[1] W<nil>: "0.05"

???> INDEXING ERROR :  
Slab[gen2] has undefined Slab-DL[1].  
You must

1. delete Slab[gen2] or
  2. insert Slab-DL[1].
- ...Which one do you want?[1/2] : "2"

ENTER Slab-DL[1] W<nil>: "0.04"

\*\* The end of insert Slab \*\*

Please, press <Return> Key to continue.

; 확인

The menus of Slab are shown below. Which one do you want to select?

- 0] Exit to Slab
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-
- 8] assign

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

\*\* The start of list Slab \*\*

/ID/DIR/Xgrid/Xgrid/Ygrid/Ygrid/Zgrid/Zgrid/  
Steel/Concrete/Slab-STL/Slab-CON/Slab-DL/Slab-LL/FORCE/  
/gen1/Bi/0/1/0/1/1/1/nil/nil/nil/1/1/nil/  
/gen2/Bi/1/2/0/1/1/1/nil/nil/nil/1/1/nil/

\*\* The end of list Slab \*\*

Please, press <Return> Key to continue.

; 재료성질을 슬라브요소에 부여함

The menus of Slab are shown below. Which one do you want to select?

- 0] Exit to Slab
- 1] list

- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-
- 8] assign

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "8"

\*\* The start of assign Slab \*\*

; 슬라브의 위치

Enter DIR : "Bi"

Enter Xgrid : "0"

Enter Xgrid : "2"

Enter Ygrid : "0"

Enter Ygrid : "1"

Enter Zgrid : "1"

Enter Zgrid : "1"

; 요소성질 정의

Possible properties are

1)Steel,2)Concrete,3)Slab-STL,4)Slab-CON,5)Slab-DL,6)Slab-LL.

Enter your selection : "1"

; 부과될 요소성질번호

Enter Steel ID to be assigned to Slab : "1"

\*\* The end of assign Slab \*\*

Please, press <Return> Key to continue.

; 확인

The menus of Slab are shown below. Which one do you want to select?

- 0] Exit to Slab
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-
- 8] assign

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"  
\*\* The start of list Slab \*\*

/ID/DIR/Xgrid/Xgrid/Ygrid/Ygrid/Zgrid/Zgrid/  
Steel/Concrete/Slab-STL/Slab-CON/Slab-DL/Slab-LL/FORCE/  
/gen1/Bi/0/1/0/1/1/1/1/1/1/nil/nil/1/1/nil/  
/gen2/Bi/1/2/0/1/1/1/1/1/1/nil/nil/1/1/nil/

\*\* The end of list Slab \*\*

Please, press <Return> Key to continue.

The menus of Slab are shown below. Which one do you want to select?

- 0] Exit to Slab
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-
- 8] assign

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

; 기동요소의 선택

The menus of ELEMENT are shown below. Which one do you want to select?

- 0] Exit to ELEMENT
- 1] Slab
- 2] Beam
- 3] Column
- 4] Brace
- 5] Wall

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "3"

The menus of Column are shown below. Which one do you want to select?

- 0] Exit to Column
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-

8] assign  
Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "2"  
\*\* The start of insert Column \*\*

; 다중입력

```
Enter Column ID/NO Value/list to insert : "temp"
ENTER Column[temp] DIR<nil>: "Bi"
ENTER Column[temp] Xgrid<nil>: "0"
ENTER Column[temp] Xgrid<nil>: "2"
ENTER Column[temp] Ygrid<nil>: "0"
ENTER Column[temp] Ygrid<nil>: "1"
ENTER Column[temp] Zgrid<nil>: "0"
ENTER Column[temp] Zgrid<nil>: "1"
ENTER Column[temp] Steel<nil>:"<CR>"
ENTER Column[temp] Concrete<nil>:"<CR>"
ENTER Column[temp] Column-STL<nil>:"<CR>"
ENTER Column[temp] Column-CON<nil>:"<CR>"
ENTER Column[temp] FORCE<nil>:"<CR>"
```

; 각 요소별 분리

Column[temp] has multi-dimensional grids. It is divided.

\*\* The end of insert Column \*\*

Please, press <Return> Key to continue.

; 확인

The menus of Column are shown below. Which one do you want to select?

- 0] Exit to Column
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-
- 8] assign

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

\*\* The start of list Column \*\*

```
/ID/DIR/Xgrid/Xgrid/Ygrid/Ygrid/Zgrid/Zgrid/
Steel/Concrete/Column-STL/Column-CON/FORCE/
```

/gen3/Bi/0/0/0/0/0/1/nil/nil/nil/nil/  
/gen4/Bi/0/0/1/1/0/1/nil/nil/nil/nil/  
/gen5/Bi/1/1/0/0/0/1/nil/nil/nil/nil/  
/gen6/Bi/1/1/1/1/0/1/nil/nil/nil/nil/  
/gen7/Bi/2/2/0/0/0/1/nil/nil/nil/nil/  
/gen8/Bi/2/2/1/1/0/1/nil/nil/nil/nil/

**\*\* The end of list Column \*\***

Please, press <Return> Key to continue.

; 재료성질의 부여

The menus of Column are shown below. Which one do you want to select?

- 0] Exit to Column
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-
- 8] assign

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "8"

**\*\* The start of assign Column \*\***

Enter DIR : "Bi"

Enter Xgrid : "0"

Enter Xgrid : "2"

Enter Ygrid : "0"

Enter Ygrid : "1"

Enter Zgrid : "0"

Enter Zgrid : "1"

Possible properties are

1)Steel,2)Concrete,3)Column-STL,4)Column-CON.

Enter your selection : "1"

Enter Steel ID to be assigned to Column : "1"

Steel[1] is assigned to Column[gen3].

Steel[1] is assigned to Column[gen4].

Steel[1] is assigned to Column[gen5].

Steel[1] is assigned to Column[gen6].

Steel[1] is assigned to Column[gen7].

Steel[1] is assigned to Column[gen8].

**\*\* The end of assign Column \*\***

Please, press <Return> Key to continue.

The menus of Column are shown below. Which one do you want to select?

- 0] Exit to Column
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save
- 7] grid-
- 8] assign

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

The menus of ELEMENT are shown below. Which one do you want to select?

- 0] Exit to ELEMENT
- 1] Slab
- 2] Beam
- 3] Column
- 4] Brace
- 5] Wall

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "<CR>"

; 요소그룹 데이터

The menus of GEOMETRY are shown below. Which one do you want to select?

- 0] Exit to GEOMETRY
- 1] FRAME
- 2] ELEMENT
- 3] GROUPING

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "3"

The menus of GROUPING are shown below. Which one do you want to select?

- 0] Exit to GROUPING
- 1] Slabs
- 2] Beams
- 3] Columns
- 4] Braces
- 5] Walls

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "3"

The menus of Columns are shown below. Which one do you want to select?

- 0] Exit to Columns
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "2"

\*\* The start of insert Columns \*\*

Enter Columns ID/NO Value/list to insert : "1"

ENTER Columns[1] DIR<nil>: "Bi"

ENTER Columns[1] Xgrid<nil>: "0"

ENTER Columns[1] Xgrid<nil>: "2"

ENTER Columns[1] Ygrid<nil>: "0"

ENTER Columns[1] Ygrid<nil>: "1"

ENTER Columns[1] Zgrid<nil>: "0"

ENTER Columns[1] Zgrid<nil>: "1"

\*\* The end of insert Columns \*\*

Please, press <Return> Key to continue.

The menus of Columns are shown below. Which one do you want to select?

- 0] Exit to Columns
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "<CR>"

The menus of GROUPING are shown below. Which one do you want to select?

- 0] Exit to GROUPING
- 1] Slabs
- 2] Beams
- 3] Columns
- 4] Braces
- 5] Walls

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "<CR>"

The menus of GEOMETRY are shown below. Which one do you want to select?

- 0] Exit to GEOMETRY
- 1] FRAME
- 2] ELEMENT
- 3] GROUPING

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "<CR>"

; 하중조합 데이터의 선택

The menus of IDEALIZED-MODEL are shown below.

Which one do you want to select?

- 0] Exit to IDEALIZED-MODEL
- 1] COORDINATE
- 2] MATERIAL
- 3] SECTION
- 4] LOAD
- 5] GEOMETRY
- 6] Load-Case

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "6"

The menus of Load-Case are shown below.

Which one do you want to select?

- 0] Exit to Load-Case
- 1] list
- 2] insert
- 3] delete
- 4] modify
- 5] load
- 6] save

Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "1"  
\*\* The start of list Load-Case \*\*

/NO/DL/LL/BL/WL/EL/

\*\* The end of list Load-Case \*\*

; 이하생략

### 7.3.2 결과토론

위의 예에서 보이는 바와 같이 구조해석을 위한 건물의 데이터의 처리는 사용자가 원하는 기능을 선택함으로써 원하는 입력을 할 수 있다. 이와같은 방법은 기존의 배치(batch) 양식에 의한 방법과 비교하여 다음과 같은 장점을 가진다.

1. 사용자의 데이터 수정이 에디터에 의한 입력파일의 작성과정 없이 즉시 가능하다.
2. 기존의 절점 및 요소의 자동재생방법과 더불어 본 연구에서는 그리드의 좌표계를 기준으로 하여 그리드의 입력, 3차원적인 요소의 입력 및 제거기능(그리드의 범위로 입력함), 요소성질(재료, 단면 및 요소하중등)의 3차원적인 부여기능을 이용한다.
3. 프레임 단위로 입력을 함으로서 해당 프레임을 구성하는 각 요소, 즉 보, 기둥, 트러스 및 벽체요소를 자동적으로 재생할 수 있다.

## 7.4 구조해석

### 7.4.1 실행예

본 내용은 사용자가 BUILDS-A를 이용하여 구조해석을 수행하고자 하는 경우 입력파일의 작성을 지원해주는 부분의 실행결과이다. 실행과정에 관한 설명은 다음과 같다.

```
strsys2% "k-clips"  
  
; 타이틀 표시 생략  
  
K-CLIPS> "(batch "III")"  
1  
K-CLIPS> (load "control.rul")  
*****  
K-CLIPS> (load "control.fct")  
$$$$$$$$$$  
K-CLIPS> (load "builds-a.rul")
```

\*\*\*\*\*

K-CLIPS> (reset)

K-CLIPS> (run)

; 타이틀 표시 생략

The menus of K-CLIPS are shown below.

Which one do you want to select?

- 0] Exit to K-CLIPS
- 1] Planning
- 2] PRELIMINARY-DESIGN
- 3] DETAILED-DESIGN
- 4] EVALUATION
- 5] PRESENTATION

Otherwise, phase 'help' will be supplied.

Enter your selection : "3"

; 구조해석 과정의 선택

The menus of DETAILED-DESIGN are shown below.

Which one do you want to select?

- 0] Exit to DETAILED-DESIGN
- 1] IDEALIZED-MODEL
- 2] STRUCTURAL-ANALYSIS
- 3] STEEL-MEMBER-DESIGN
- 4] CONCRETE-MEMBER-DESIGN
- 5] FOUNDATION-DESIGN

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "2"

; 입력파일 작성 선택

The menus of STRUCTURAL-ANALYSIS are shown below.

Which one do you want to select?

- 0] Exit to STRUCTURAL-ANALYSIS
- 1] create-inputfile
- 2] check-inputfile
- 3] analyze
- 4] review-results
- 5] REFINE-MODEL

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "1"

\*\* The start of create-inputfile STRUCTURAL-ANALYSIS \*\*

; 각 카드에서 요구되는 데이터의 처리과정

## 1. CONTROL INFORMATION CARD

Number of stories in the complete building(not including the footing line).

NST="20"

Number of frames with different properties or different vertical loading.

NDF="3"

Total number of frame or shear wall elements in the structure

NTF="5"

Total number of load conditions

NLD="3"

Analysis type code:

EQ. 0; Static loads only

EQ. 1; Mode shapes and frequencies only

EQ.-1; Input and/or generation of approximate mode shapes and periods plus static analysis using lateral force cases(A and B) for the solution of the gross response quantities.

EQ. 2; Type 0 + Type 1

EQ. 3; Lateral earthquake response spectrum analysis for the solution of individual frame displacements and member forces in addition to analysis type 2, above

EQ. 4; Lateral earthquake time history response analysis for the solution of individual frame displacements and member forces in addition to analysis type 2, above

EQ. 5; This option is not available for use

EQ. 6; Lateral earthquake response spectrum analysis for the solution of the gross response quantities.

EQ.-6; Input and/or generation of approximate mode shapes and periods plus lateral earthquake response spectrum analysis for the gross response quantities

EQ.-7; Type 3 + Type 6, above

EQ. 8; Lateral earthquake time history response analysis for the solution of the gross response quantities.

EQ.-8; Input and/or generation of approximate mode shapes and periods plus lateral earthquake time history response analysis for the gross building response quantities

EQ. 9; Type 4 + Type 8, above

NAT="0"

Number of frequencies to be calculated  
NFQ="0"

Allowable story degrees of freedom:  
EQ.0; X,Y translations + story rotations  
EQ.1; X translation only for sym. buildings only  
EQ.2; Y translation only for sym. buildings only  
NSD="0"

Each lateral loadcase force generation code  
For Loadcase A.....  
EQ. 0; No force generation.  
EQ. 1; Generate forces in X direction.  
EQ. 2; Generate forces in Y direction.  
EQ. 3; Generate forces in both X and Y directions.  
KODE1="0"

For Loadcase B.....  
KODE2="0"

Element stress ratio calculation code  
EQ. 0; No stress ratios calculated  
EQ. 1; Stress ratios calculated for column, beam, and  
diagonal brace elements.  
KODE3="0"

Execution code  
EQ. 0; Full Execution  
EQ. 1; Data check  
ICHECK="0"

Acceleration of gravity for use in calculation of P-delta effects  
GRAV="980.0"

Building identification information to be printed with the output  
BHED="This is a test run"

; 두번째 입력카드의 재생 및 데이터 처리

## A2.0 STORY DATA

Two cards per story level is entered in sequence  
from top to bottom of the structure. Now,  
a. First Card (1 story)

Five characters to be user for level identification  
ID="roof"

Story height[distance from the floor(or roof) level  
to the floor(or foundation) level below]  
HEIGHT="390"

Translation mass  
Mt="0"

Rotational mass moment of inertia about a vertical axis  
through the center of mass  
Mr="0"

Distance to the center of mass measured from the reference point  
Mtx="0"

Mty="0"

External story stiffness in the each direction  
Xk="0"

Yk="0"

; 세번째 카드

b. Second Card (1 story)  
Lateral loadcase A:  
Enter load in each direction  
Fxa="8000"

Fya="6000"

Enter coordinate at the point of load  
Xa="2000"

Ya="1500"

Lateral loadcase B:  
Enter load in each direction  
Fxb="0"

Fyb="0"

Enter coordinate at the point of load  
Xb="0"

Yb="0"

; 네번째 카드

## A2.0 STORY DATA

Two cards per story level is entered in sequence from top to bottom of the structure. Now,

a. First Card (2 story)

Five characters to be user for level identification

ID="19th"

; 이하 생략

; 구조해석의 실행

The menus of STRUCTURAL-ANALYSIS are shown below.

Which one do you want to select?

- 0] Exit to STRUCTURAL-ANALYSIS
- 1] create-inputfile
- 2] check-inputfile
- 3] analyze
- 4] review-results
- 5] REFINE-MODEL

Otherwise, phase 'help' will be supplied.

Enter your selection<0> : "3"

\*\* The start of analyze STRUCTURAL-ANALYSIS \*\*

Enter input file name : "test.dat"

test.dat is opened.

Enter output file name : "test.out"

test.out is opened.

Your building is being analyzed.

\*\* The end of analyze STRUCTURAL-ANALYSIS \*\*

Please, press <Return> Key to continue.

The menus of STRUCTURAL-ANALYSIS are shown below.

Which one do you want to select?

- 0] Exit to STRUCTURAL-ANALYSIS
- 1] create-inputfile
- 2] check-inputfile

3] analyze  
4] review-results  
5] REFINE-MODEL  
Otherwise, phase 'help' will be supplied.  
Enter your selection<0> : "<CR>"

; 이하생략

#### 7.4.2 결과토론

위의 예제에서는 사용자가 구조해석을 수행하기 위하여 곧바로 입력파일을 작성하고자 하는 경우 시스템의 실행결과를 보이고 있다. 본 연구에서는 예비설계과정 및 이상화과정을 거쳐 작성된 데이터가 기억공간에 있을 경우에는 사용자가 정의한 데이터를 바탕으로 각 변수의 값을 전달받도록 하고, 값이 미지일 경우에는 사용자로부터 직접 입력을 요구한다. 이러한 과정이 진행되어 감에 따라 필요한 데이터 카드가 재생되고, 해당카드의 변수에 필요한 설명을 제공함으로써 본 시스템을 BUILDS-A의 매뉴얼대신 사용할 수 있다.

## 8. 결 론

건축구조물의 구조해석 및 설계과정을 자동화하기 위하여 본 연구에서는 다음과 같은 결과를 제공하고 있다.

1. 건물의 설계과정에 대한 모델의 설정
2. 자동시스템을 위한 데이터 구조의 완성
3. 사용자와의 메뉴 인터페이스를 위한 메뉴구조완성
4. 예비설계단계의 지원
5. 모델링과정의 지원
6. 입력파일의 재생부분 완성
7. 예비설계 결과의 입력파일로의 전달

위의 연구에 관련하여 요구되는 앞으로의 연구과제는 다음과 같다.

1. 해석결과의 분석(해석결과의 검토부분, 부재의 개선등)
2. 그래픽을 이용한 결과의 제시
3. BUILDS-S/C/F/B/D의 지원시스템의 지원

## 9. 참고 문헌

### A. 인공 지능에 관한 문헌

1. P.Harmon and D. King, *Artificial Intelligence in Business : Expert System*, John Wiley and Sons, New York, 1985.
2. Hayes-Roth F, Waterman D.A, Lenat D.D., *Building Expert System*, Addison-Welsey, 1983.
3. D.A.Waterman, *A Guide to Expert System*, Addison-Wesley, US, 1986.
4. B.G.Buchanan and E.H.Shortlife, *Rule-based Expert Systems*, Addison-Wesley, US, 1984.
5. S.Weiss and C.Kulikowski, *A Practical Guide to Designing Expert Systems*, Rowman and Allandheld, Totowa,NJ, 1984.
6. C.Townsend and D.Feucht, *Design and Programming Personal Expert Systems*, TAB, 1986.
7. L.Brown, R.Farrel, and E.Kant etc., *Programming Expert System in OPS5*, Addison-Welsey, 1986.
8. F.Chehayeb and J.Conner, *GEPSE-A Computer Environment for Engineering Problem Solving*, Research Report R86-11, Dept. of Civil Engineering, MIT, 1985.
9. 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.

10. Mark H. and Richer, An Evaluation of Expert System Development Tools, Expert System, July, Vol.3, No.3, 1986.

11. Phillip J.L., William J.G., and Del D., Expert System Tools for Civil Engineering Applications, Expert Systems in Civil Engineering, 1986.

12. Joseph C. Giarrantano, CLIPS User's Guide, 1987.

13. Sobelman and Krekelberg, Advanced C : Techniques and Applications, QUE, 1985.

14. M.Waite, S.Prate, and D.Martin, C Primer Plus, Howard W.Sams & Co.,Inc., 1984.

15. B.W.Kernigham and D.M.Ritchie, The C programming Language, Prentice-Hall, 1978.

16. Herbert Schildt, C : The Complete Reference, McGraw-Hill, 1987.

## B.I-BUILDS구축에 관한 문헌

1. D.Sriram, M.I.Maher and S.J.Fenves, Knowledge-based Expert System in Structural Design, Computers and Structures, Vol.20, No.1-3, pp. 1-9, 1985.
2. D.Sriram, DENSTINY : A Model for Integrated Structural Design, Artificial Intelligence, Vol.1, No.2, pp.109-116, 1986.
3. M.L. Maher, Expert System for Civil Engineering : Technology and Application, ASCE.
4. M.L. Maher and S.J.Fenves, HI-RISE : A Knowledge-Based Expert System for the Preliminary Structural Design of High Rise Building, 1985.
5. M.F.Rooney and S.E.Smith, Artificial Intelligence in Engineering Design, Computers and Structures, Vol.16, No.1-4, pp.279-288, 1985.
6. Furuta,H.. Tu,K.S. and James,T.P., Structural Engineering Applications of Expert Systems, Journal of Computer Aided Design, Vol.17, No.9, pp410-419, 1985.
7. W.J.Rasdof and G.C.Salley, Generative Engineering Database-Toward Expert Systems, Computers and Structures, Vol.20, No.1-3, pp11-15, 1985.
8. S.J.Fenves and J.H.Garrett, Knowledge Based Standard Proceeding, Artificial Intelligence, Vol.1, No.1, pp.3-14, 1986.
9. C.K.Soh and A.K.Soh, Example of Intelligent Structural Design System, Journal

of Computing in Civil Engineering, Vol.2, No.4, October, 1988.

10. M.S.Jones and V.E.Saouma, Prototype Hybrid Expert System for R/C Design, Journal of Computing in Civil Engineering, Vol.2, No.4, April, 1986.

11. M.R.Wigan, Engineering Tools for Building Knowledge-based System on Microsystem, Microcomputers in Civil Engineering, pp.52-68, 1986.

11.1 D.Sriram, Knowledge-Based Approaches For Structural Design, Topics in Engineering Vol.1 edited by C.A. Brebbia and J.J.Connor, Computational Mechanics, 1987

11.2 M.L.Maher, Problem Solving using Expert System Techniques, Expert System in Civil Engineering edited by C.N.Kostem and M.L.Maher, April 1986

12. C.K.Choi and E.D.Kim, BUILD-A : A Three Dimensional Analysis Program for Building Structure, Proceedings of the Second International Conference on Civil and Structural Engineering Computing : CIVIL-COMPO 85, London, England, 1985.

13. E.D.Kim, Linear 3-D Analysis of Building Structures, Thesis of M.S. Degree, Department of Civil Engineering, KAIST, Seoul, Korea, 1986.

14. C.K.Choi and H.W.Lee, Three-Dimensional Analysis and Optimal Design of Building Structures, Proceedings of 10th Triennial CIB Congree, Washington,D.C., 1986.

15. 이 환우, 규격부재로 이루어진 프레임의 최적화, 석사학위논문, 한국과학기술원, 1986.
16. 최 창근, 박 효경, 철근콘크리트 구조의 최적설계, SEMR89-04, 1989.
17. 건설부, 건축물의 구조기준등에 관한 규칙.
18. 대한건축학회, 철근콘크리트 구조계산 기준.
19. 김 규석, 철골구조학, 기문당, 1987
20. 대한건축학회, 강구조 계산 기준
21. 김 우석, 철골구조, 형설출판사, 1986
22. 김 현산, 건축구조계획, 문운당, 1988
23. Wolfgang Schueller, High-rise Building Structures, John Wiley and Sons, 1977.
24. T.Y.Lin and S.D.Stotesbury, Structural Concepts and Systems for Architects and Engineers, John Wiley and Sons, 1981.
25. Kenneth R. Lauer, Structural Engineering for Architects, McGraw-Hill, 1981.
26. ACI Committe 442, Response of Buildings to Lateral Forces, ACI Journal, 1971.
27. ASCE, Monograph on Planning and Design of Tall Buildings, Vol.PC (1981), Vol.SC (1980), Vol.CL (1980), Vol.SB (1979), Vol.CB (1978), New York.

28. M.Sefik, Planning with Constraints(MOLGEN 1), Artificial Intelligence, Vol. 16, pp.111-140, 1981.
29. 최 무혁, 건축계획 과정에서의 평가모델 선정에 관한 연구, 대한건축학회지, 24권 96호, 1980.
30. Harty,N., An Aid to Preliminary Design, Preceedings of 2nd Interational Conference on Application of A.I. in Eng. Problems, Cambrige Massachusetts, 1987.
31. Mikell P.,Emory W.Z., CAD/CAM : Computer Aided Design and Manufacturing, Prentice-Hall, 1984.
32. C.K.Choi and E.D.Kim, A Preliminary Model of I-BUILDS : An Intelligent Building Design System, Proceedings of Second International Conference on Application of Artificial Intelligence in Engineering, Knowledge Based Expert System in Engineering : Planning and Design, Edited by D.Sriram and R.A.Adey, Computational Mechanics Publications, Boston,USA, 1987.
33. 한국과학기술원, 인공지능을 이용한 고도의 구조해석/설계용 전문가시스템의 개발 (I,II), 과학기술처, 1988-89
34. Stanley W. Crawley and al., Steel Buildings Analysis and Design, John Wiely and Sons, 1984.
35. Wolfgang Schueller, Horizontal-Span Building Structures, John Wiley and Sons,

1983.

36. C.V.Clarke and R.Muller, GIFTS-1100 : Graphics Oriented Interactive Finite Element Time-Sharing System, Finite Element Systems edited by C.A.Brebbia, Computational Mechanics Centre, Southampton, p.p. 225-240, 1982.

### PART III 교량설계 전문가시스템

## 1. 서론

### 1.1 연구개요

일반적으로 교량의 예비설계에 있어서 적절한 설계를 위해서는 법규, 지형조건, 환경조건, 경제조건등의 여러조건을 주시해 가면서 설계자의 상당한 숙련 및 경험이 필요하고 또한 고도의 지식도 함께 요구된다. 그러나 국내의 교량설계에서는 여러 제약조건에 의해 교량계획단계에서 경제성, 시공성, 안전성등의 여러 조건에 대한 충분한 검토가 부족한 실정이다. 또한 교량의 경간분할이나 가교위치, 상부구조형식 및 공법, 기초형식등을 결정할 수 있는 관련지침이나 기준이 미흡한 관계로 설계자들은 기존에 설치된 유사한 교량의 설계도면을 참고하거나 상급자의 경험에 전적으로 의존하고 있는 실정이다. 이러한 교량계획 설계지침과 기준의 미비는 교량설치 계획에 있어서 통일성 결여와 설계, 시공단계에서 심한 경제적 손실을 초래하여 전체공사비의 과다책정 원인으로 되고 있다. 그럼에도 불구하고 현재 국내에는 이러한 지침이나 기준이 절대적으로 부족하고 이에 대한 연구수행 실적마저 전무한 실정이다. 따라서 교량계획 단계에서의 합리적이며 종합적인 기준 및 이에 대한 요구가 절실히 요망되고 있다.

### 1.2 연구목적

현재 토목공학에 있어서의 구조해석이나 계산작업의 면에서 컴퓨터의 이용은 거의 확립되었다고 말해도 과언은 아니다. 현재에는 단순한 기술적인 측면 뿐만 아니라 보다 넓은 범위에 있어서 컴퓨터를 이용하려는 노력이 행해지고 있다. 1960년대 후반에 미국에 있어서 그의 개념이 생겨난 CAD(Computer Aided Design) 시스템을 포함한 자동설계의 보급은 그 현상의 하나이고, 현재 토목공학의 각 분야에서 설계지원 시스

템으로서 이용되고 있다. 그러나 교량등의 특정구조물의 예비설계단계인 보다 상위의 단계에 있어서는 컴퓨터의 이용은 아직까지는 미흡하게 행해지고 있다. 그래서 본 연구에서는 교량의 예비설계분야인 교량상부구조형식의 선정, 가설공법의 선정 및 기초형식의 선정과 결정된 교량상부구조의 해석 및 설계분야를 현재 의사결정 지원시스템으로 많이 이용되는 인공지능(Artificial Intelligence)분야중에서 특히 주목을 받고 있는 전문가 시스템(Expert System)을 구축함으로써 사용자들이 쉽고 간편하게 실무에 적용할 수 있도록 함으로써 실무에 큰 도움이 되도록 하는 것이다.

### 1.3 연구배경

전문가 시스템이란 해결해야할 문제에 속하는 영역의 전문지식(전문적 사실과 경험적 지식)을 데이터베이스(Data Base)화 하여 비전문가가 이것을 이용해서 매우 충분한 문제를 전문가와 동등한 정도의 수준으로 해결하는 것을 목표로 하는 시스템이다. 최근에 비교적 성공적인 결실을 거둔 전문가 시스템을 보면 의사의 진단을 위한 MYCIN, 광택탐사를 위한 PROSPECTOR 및 컴퓨터의 설치계획을 위한 R1등이 있으며, 구조공학 분야에서도 수치계산을 주로 행하는 기존의 컴퓨터 이용방법을 기호처리(symbolic processing)에 바탕을 둔 새로운 기법을 도입하여 컴퓨터의 이용효율을 향상시키려는 노력이 이루어지고 있다. 이에 대한 전문가 시스템으로는 중층 건물의 개략설계 지원을 위한 HI-RISE, 구조해석법의 선정을 위한 SACON, 내진손상도 해석을 위한 SAGE, 재료의 파괴해석을 위한 FAX 및 구조안정성 평가를 위한 HOWSAFE, DAPS등이 있다.

또한 교량의 개략설계지원을 위한 전문가 시스템으로는 M. Biswas 와 J.G. Welch에 의한 BDES(A Bridge Design Expert System)와 T. Pagnoni에 의한 Expert Systems for Bridge Engineering 및 H. Furuta 와 N.Shiraishi에 의한 Application of Expert

Systems to Structural Engineering등이 있다.

#### 1.4 연구범위

본 연구는 교량의 예비설계에서부터 해석 및 설계를 하는 과정을 목적으로 하는 시스템으로서 예비설계 단계에서는 상부구조형식의 선정, 가설공법의 선정, 경간분할의 결정, 교각형식의 결정, 기초구조형식의 결정, 선정결과의 평가 및 선정된 상부공형식에 대한 해석 및 설계를 지원하는 것으로 이에 대한 전체 시스템 명칭을 BADES( A Bridge Analysis and Design Expert System)으로 하고 다음과 같은 부시스템을 포함한다. 상부구조의 형식선정은 BADES-S, 가설공법의 선정은 BADES-C, 기초구조의 형식선정은 BADES-F, 선정된 상부구조에 대한 해석 및 설계는 BADES-A 및 BADES-D로 한다.

## 2. BADES의 배경

### 2.1 예비설계의 일반조사사항

교량을 계획, 설계, 시공하는 데에는 잘못된 조사에 의해 공기 및 공사비등에 차질을 가져오는 경우가 종종 있으므로 사전조사가 절대적으로 중요하다. 또한 교량을 건설하는데 있어 큰 변경없이 합리적이고 경제적인 계획, 설계, 시공을 수행하기 위하여 구조물의 규모, 중요성 및 교량 설치지점의 상황등을 정확하게 판단할 수 있게 하는데 조사의 목적이 있다. 일반적으로 교량의 계획에서 시공까지 각 단계별로 계획조사, 설계조사, 시공조사를 행해야 하며, 조사의 내용은 교량규모에 따라 차이가 있으며, 상, 하부공은 상호 연관을 가지며 계획조사 단계에서 효율적인 조사를 위해서 상, 하부공의 조사계획을 따로 마련하여 중복되거나 누락되지 않도록 작성하여야 한다.

표 2.1 교량설계의 분류

구 분	정 의	내 용
1. 일반도 작성	*교량설치 지점에서 비교 대상이 되는 여러가지 형식의 구조물 일반도를 작성하는 작업으로 구조물의 전체적인 개념을 도면에 나타내는 설계	*교량 종, 평면도 및 일반도 *토질주상도 *수량 및 공비(기존 자료 추정가능)
2. 계획설계	*구조물의 형식비교에 필요한 자료를 작성하는 설계	*주요단면 크기결정 *철근배치 *말뚝보수 *개략수량 산출 *구조계산(기존자료 추정가능)
3. 기본설계	*중요하지 않은 세부 설계는 생략한 설계	*구조체 치수결정 *난간, 배수장치, 신축이음 받침등의 형식 결정 *기타 주요단면크기 결정 *사용재료 결정 *시공법 결정 *수량산출(기존자료 추정가능)
4. 상세설계	*구조물을 정확히 시공 하는데 필요한 모든 것을 작성하는 설계	*교량 상세도면 작성 *시공계획서 *수량 및 공비

일반적으로 교량의 최종도면이 완성되기까지는 다음과 같이 그 내용 및 중요도에 따라 표2.1과 같이 4단계로 분류할 수 있다.

교량의 설계는 표2.1에서와 같이 일반도 작성에서 상세설계로 진행되는 것으로 개략 설계에서 점차 세부설계로 발전시켜 나간다. 교량의 설계시 표준설계가 응용될 수 있는 교량 및 중소교량은 표준도 및 과거의 자료를 참고하면 시간을 크게 줄일 수 있다.

## 2.2 BADES의 개요

일반적으로 교량설계등에서 보여진 현행의 구조물 설계의 최종단계에 위치하는 상세설계에서 가장 중요한 부분은 역학적 검증이지만 동일역학 모델로 설계, 계산등이 행해지고 있는 것은 아니다. 결국의 교량의 현행 역학설계는 교량상부구조의 각 부재, 하부구조, 기초구조, 기초주변의 지반등이 동일모델로 풀리는 만큼 안정한 구조물을 도출하기 위한 역학적 설계는 구조물 설계의 대전제라고 할 수 있다.

이처럼 교량설계의 경우에서도 구조물의 구성요소를 가능한 많이 고려하고 목적변수인 평가요소에 대해서는 역학적 안정성외에 경제성, 기능성, 환경적합성등의 항목의 각 지표를 시계열적인 측면과 함께 내부, 외부의 영향효과의 관점에서도 동일수준에서 논할 필요가 있을 것이다. 아울러 구조물의 설계기술도 높은 수준이 되어 이와같은 수준에서 각 평가항목을 습득하는 것은 1인의 기술자에 있어서는 어렵다고 할 수 있다.

그러나 지금까지의 공학설계에 컴퓨터를 활용하려는 노력을 분석해보면 주로 컴퓨터가 가지는 특성, 즉 빠른 연산속도, 컴퓨터 그래픽스에 의한 시각적 영상효과, 그리고 많은 양의 데이터 저장에 중점을 두어왔기 때문에 위에서 언급한 바와 같은 여러 측면에서의 경험적 판단에는 적용하기가 어려웠고, 다만 세부단계에서 물량측정에 의한 경제성 문제에 국한된 결과를 바탕으로 설계자는 자신의 경험에 의존, 그 적절성을

판단하여 설계를 완성하였다. 이러한 문제는 인공지능의 기법의 발달로 인간의 경험에 의존한 지식 즉, 개념과 추론에 의한 결과의 도출과정등도 컴퓨터에 의한 설계과정의 한부분으로서 도입하게 됨으로서 자동화된 설계시스템을 구현하기 위한 기술적 토대를 마련하게 되었다.

일반적으로 지식을 습득해나가는 과정은 주로 정립된 이론에 근거하여 이루어지므로 처음으로 실제적인 업무에 나서는 사람은 경험의 부족상태에 있다. 구조공학분야에서도 처음에는 구조역학의 기본적인 이론에만 익숙해져 있기 때문에 여러조건이 주어지는 구조물의 실제설계, 특히 적절한 구조시스템을 선정하는 문제에는 많은 어려움이 뒤따른다. 이러한 분야는 그 이전의 축적된 경험을 요하는 부분이므로 지식의 체계화로 전문가 시스템의 도입이 필요함을 적절히 설명한다고 볼 수 있다.

본 연구는 위와 같은 제반 설계조건, 구조요소의 역학적 특성등에 관한 기존의 이론적 지식과 이 분야의 전문적인 경험적 지식을 바탕으로 자동화된 교량설계의 일관시스템을 개발하려는 노력의 하나로서, 특히 교량의 설계과정이 여러분야의 참가자들이 상호 충분한 정보교환을 거듭하게 됨으로써 다방면의 전문가들의 협력에 의하여 진행이 되므로 이러한 일련의 설계과정을 일관된 모델로 정립하는데 그 목적이 있다. 이러한 전문가시스템을 이용함으로써 짧은 시간에 사람이 생각해 볼 수 있는 가능성보다 더 많은 때로는 가능한 모든 후보의 정보를 체크해 볼 수 있으므로 교량의 설계에 커다란 도움을 줄 수 있다.

### 2.3 BADES의 설계모델

이상과 같은 배경 및 관점에서 본 연구에서는 교량의 예비설계를 대상으로 각 전문가의 지식, 경험을 토대로 한 부시스템(sub system)을 설정하고, 또한 그것들을 합성한 전체시스템(total system)의 구축을 목표로 하는 것이다. 물론 각 부시스템간에는

동일지표로 할 수 없는 것등의 문제점이 있다. 이상과 같은 사항을 고려한 예비설계의 모델은 그림 2.1과 같다.

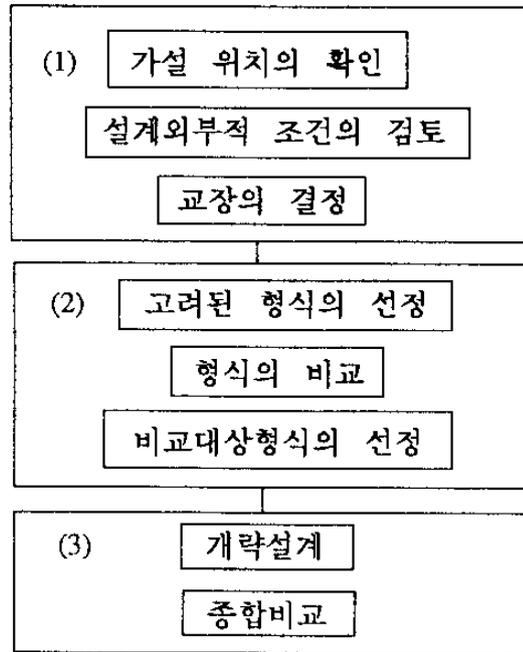


그림 2.1 예비설계의 모델

그림 2.1에서 (1) 교량 가설지점의 지형, 지질, 인접노선계획의 상황, 하천개수계획 및 하천의 관리조건등을 토대로 교장을 결정하는 작업이다. (2) 교종, 경간분할, 구조 등에 대해서 충분한 검토를 행하여 적당하다고 생각되는 교종에서 하부구조, 상부구조 및 가설공법을 2-3종류를 선정하는 작업이다. (3) (2)에서 선정된 종류에 대해서 개략 설계를 행하여 경제성, 구조특성등에 대해서 종합비교를 행한다.

본 연구에서는 이같은 예비설계내에서 (2)에서의 사용을 전제로 한 prototype의 시스템구축을 위주로 한다.

## 2.4 BADES의 구성

전문가 시스템은 앞에서 설명된 바와 같이 전문적인 지식을 지식베이스(knowledge base)로서 저장하고 그 저장된 것을 이용하여 문제해결을 하는 시스템이다. 여기서 전문적인 지식은 사실(fact)에 관한 지식과 전문가의 경험으로부터 얻은 지식(heuristic knowledge)을 가리킨다. 일반적으로 전문가 시스템의 구축에 있어서 요구되는 것으로는 지식의 획득, 지식의 표현, 지식의 이용 및 사용자와 기계와의 인터페이스(interface)의 4가지를 들 수 있다. 그중에서 가장 중요한 문제는 지식의 획득이다. 지식중에서 전문가의 경험적 지식을 얻어내는 일은 매우 어렵고 이와 같은 지식은 아무리 전문가라 해도 표현상의 어려움이 있기 때문에 이 경험적 지식을 어떻게 알기쉽게 체계화하여 정립할 것인가 하는 문제가 지식베이스 구축의 성공여부를 좌우한다.

전문가 시스템의 대표적인 지식표현형식으로서 생산시스템(production system)이 있는데 이 시스템에는 IF(조건부), THEN(실행부)의 형태를 가지는 생산규칙(production rule)을 모은 지식베이스가 있다. 이 시스템의 특징은 지식을 이해하기 쉽고 정의, 변경 및 확장이 용이하다는 것이다.

본 연구에서는 생산시스템 방식에 적합한 전문가 시스템 개발용 도구(tool)인 K-CLIPS(KAIST-C Language Integrated Production System)를 이용하여 본 시스템을 구축하고자 한다. 본 시스템에서의 지식베이스의 구축은 시방서, 교량실시설계 보고서 및 교량전문가등의 경험을 토대로 한 의견을 참고로 하여 지식을 체계화 하였으며, BADES의 통합적인 개념은 그림 2.2와 같다.

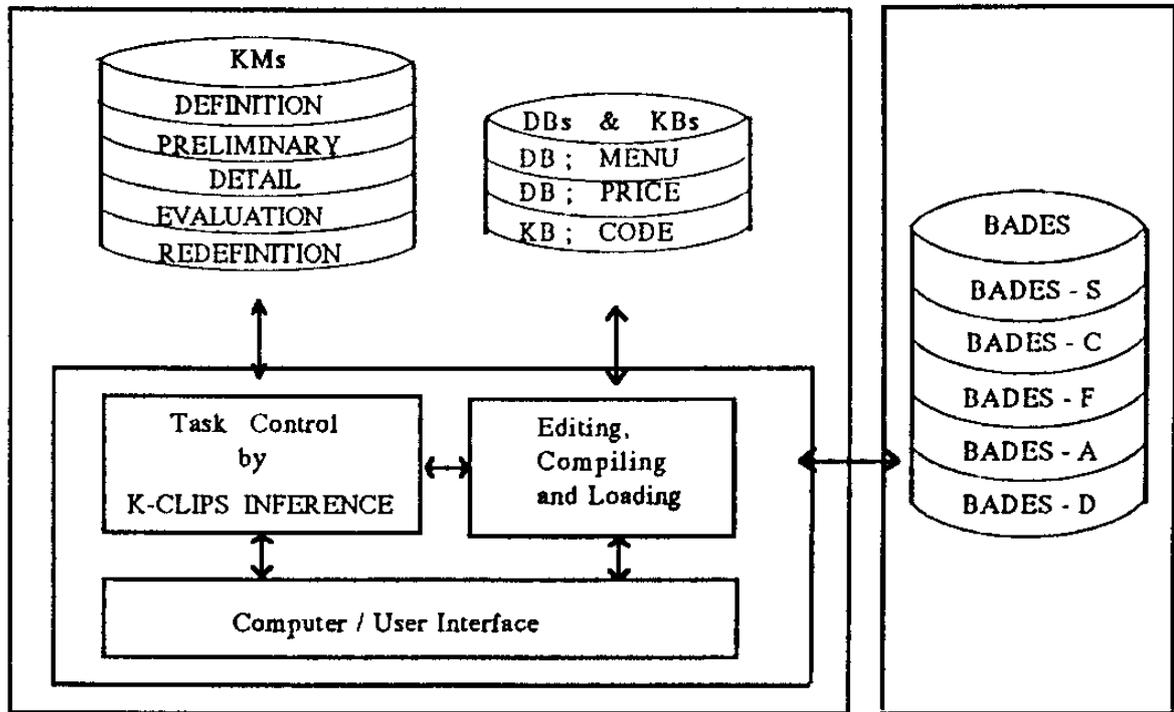


그림 2.2 BADES의 구조

## 2.5 BADES의 범위

BADES의 최종목적은 모든 형태의 교량설계에 적용되는 시스템을 완성하는 것을 목적으로 한다. 본 연구에서 고려하고 있는 범위로는 BADES의 부시스템에서 교량의 예비설계에 해당하는 교량상부구조 형식의 선정(BADES-S), 선정된 상부구조 형식에 대한 가설공법의 선정(BADES-C) 및 기초형식의 선정(BADES-F)으로 되어 있으며, 본 시스템에서 고려하고 있는 상부구조, 가설공법 및 기초의 형식은 표 2.2, 표 2.3과 표 2.4와 같다.

표 2.2 상부구조형식의 종류

steel superstructures	concrete superstructures	pc superstructures
simple plate girder continuous psate girder simple box girder continuous box girder simple truss continuous truss langer truss - upper deck - through deck arch	slab hollow slab rahmen t-beam	simple girder continuous girder simple box girder continuous box girder beam

표 2.3 가설공법의 종류

steel superstructures	pc superstructures
staging method - with truck crane - with cable crane - with stiffleg derrick - with floating crane cable erection method erection truss method pushing out method - with erection truss - with erection nose - with barge - with movable bent cantilever method - with truck crane - with cable crane - with stiffleg derrick - with floating crane large block method - with truck crane - with floating crane - with barge	temporary bent method movable scaffolding method incremental launching method free cantilever method progressive placing method precast girder method precast segment method - with truck crane - with floating crane - with cable crane - with tower crane

표 2.4 기초형식의 종류

shallow foundation rc pile pc pile steel pile benoto pier earth drill pier reverse circulation pier deep pier open caisson pneumatic caisson
---

### 3. BADES의 구축

#### 3.1. 상부구조형식의 선정시스템 (BADES-S)

##### 3.1.1 상부구조형식의 선정기준

교량을 계획하는 데에는 사전에 각종 예비조사가 행하여지며 그 결과에 따라 노선이 먼저 결정되어진다. 그리고 교량의 개략비교설계를 통하여 교량규모, 경간분할, 구조형식의 결정을 하게 된다. 이때 각각의 비교설계에 대해 기존자료를 최대한 이용함으로써 시간을 줄일 수 있고, 최적의 형식이 결정되면 이에 대한 상세설계로 진행하게 된다. 이상이 교량을 계획하는 일반적인 순서이며 이때 고려해야할 교량형식 선정의 요인은 다음과 같다.

##### (1) 노선선형 및 가교위치

일반적으로 설계자는 주어진 도로 선형조건에 맞춰 설계하는데, 많은 경우에 교량구조물이 도로 전체에서 차지하는 공비 및 공정상의 영향은 크다. 따라서 노선 선정시에는 교량의 설계 및 시공, 경제성등도 같이 고려하여 유리한 선형을 선택하도록 하여야 한다. 특히 장대교인 경우에는 공비의 영향이 크기 때문에 충분히 고려하여 결정해야 한다. 그리고 일단 결정된 노선에 대해서도 용지나 기타 중요요소에 지장을 주지않는 한도 내에서 약간의 수정을 가하여 교량의 설계시공에 유리하게 하는 것이 유효한 방법이 될 수 있다.

##### (2) 외적조건

교량계획을 하는데 있어 외적조건이란 교장, 지간, 교대 및 교각의 위치, 방향, 형하고 및 기초의 근입심등과 같이 외부적 요인에 의해 결정되는 것으로 이것은 지형, 기초지질의 상태등에 따라 변하는 것으로 하천 및 도로 관리자와 충분히 협의 하여야

하고 정확한 기초 지질조사가 수행되어야 한다. 또한 소음, 진동등의 환경조건에 대해서도 형식결정시 충분히 고려해야 한다.

### (3) 안전성과 경제성

교량형식이 구조상 안정한 것인가는 기술적으로 중요한 문제로 구조계산과 설계자의 경험에 의하여 충분히 검토되어야 한다. 또한 이에 못지 않게 중요한 문제는 경제성의 비교로 상부와 하부구조를 합한 총 공사비 측면에서 검토되어야 하고 경제성 비교는 다음과 같은 비교에 의해 수행하는 것이 일반적이다.

- 1) 단순한 형식의 비교에 있어서는 일반도 작성의 단계에서 기존자료나 표준도를 이용하여 결정한다.
- 2) 장대교와 같이 지간이 길거나, 하부조건이 복잡한 경우 1)에 의해 결정이 되지 않는 경우에는 2-3개의 형식에 대해 비교검토를 통하여 결정한다.

### (4) 시공성과 유지관리

공비가 비슷한 경우에는 시공성과 유지관리의 용이성에 대해 검토하여야 한다. 근래 들어 유지관리성의 중요성이 강조되고 있는데 유지관리가 용이한 구조물이 유지비용도 절약될 수 있으므로 이에 대해 충분히 고려해야 한다.

### (5) 표준화

교량의 계획 중간단계에서는 상, 하부 및 기타 부대시설에 따라 각각 독립적으로 최적의 형식 및 교장등이 결정되는데 계획의 최후 단계에서는 교량을 전체적으로 검토하여야 한다.

예를 들면 몇개의 교량을 일시에 시공하는 경우에 있어 다른 조건으로 설계하는 것보다 같은 형식과 경간으로 하는 것이 설계비와 공비의 전체금액 면에서 경제적일 뿐 아니라 시공면에서도 능률이 좋다. 또한 표준화가 되면 설계의 획일화, 조사의 간소화, 시공능률 향상등을 기할 수 있다.

### (6) 안정성과 쾌적성

교량상의 주행중 안정성과 쾌적성을 지배하는 요소는 노선의 선형외에 노면상에 설치된 신축장치등이 있는데 이에대한 충분한 검토를 요한다.

### (7) 미관

교량의 계획에 있어 경제성 뿐만 아니라 주변 지형과의 조화등 미적 요소를 고려하는 것이 필요하다.

## 3.1.2 선정시스템의 구축

본 시스템에서 고려된 상부구조의 형식은 표 2.2에서 나타난 바와 같이 강교와 콘크리트교로 분리할 수 있다. 상부구조의 형식을 선정시 고려할 조건으로는 아주 많이 있지만 본 시스템에서는 그중에서 가장 중요하다고 생각되는 교량가설지점의 위치, 교장 및 중앙지간등을 고려한다. 위의 조건으로부터 몇가지 비교안이 나오게 되면 그 비교안에 대하여 경제성, 시공성, 유지관리, 주행성 및 미관등을 고려하여 평가를 하게 되어 최종적인 교량상부구조 형식이 선정된다. 이에 대한 순서도는 그림 3.1과 같다.

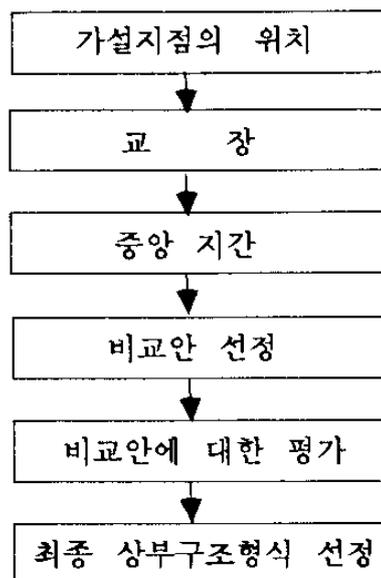


그림 3.1 상부구조형식 선정의 순서도

그림 3.1에서 입력데이터에 대한 rule의 형태는 그림 3.2와 같다. 그림 3.2의 deffacts initial에서 보는 바와 같이 교장의 범위는 10-1000m로 하였고, 교량의 위치는 하천이나 강, 도로, 철도 및 계곡중에서 택일하도록 하였으며, 지간은 8단계로 분류하였으며 이에 대한 범위는 5-150m로 하였다.

또한 비교안을 선정하기 위한 중앙지간은 각 교종에 따라 각각 이상적인 지간이 있는데, 교량계획시 경간분할 단계에서는 하천의 상황, 경제성, 용도, 외관등의 관점에서 검토하는 것이 필요하다. 그렇지만 실제로는 하부구조와의 관계, 교장등의 제한때문에 생각한 것처럼 쉽게 정해지지 않는다. 그래서 본 시스템에서는 문헌분석 및 전문가의 의견을 참고로 하여 표 3.1과 같이 각 교종별로 표준적용지간의 범위를 정했다.

표 3.1 각 교종별 표준적용지간

형 식		지 간 (m)														
		10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
RC 교	slab															
	hollow slab															
	rahmen															
	t-beam															
PC 교	simple girder															
	continuous girder															
	simple box girder															
	continuous box girder															
	beam															
STEEL 교	simple plate girder															
	continuous plate girder															
	simple box girder															
	continuous box girder															
	simple truss															
	continuous truss															
	langer(upper deck)															
	langer(through deck)															
	arch															

(PC continuous box girder 교의 경우 가설공법에 따라 지간을 150m까지 할 수 있다)

```

(deffacts w.f " "
  (servicebility.weighting.factor nil)
  (constraint servicebility.weighting.factor range 0 1)
  (aesthetic.weighting.factor nil)
  (constraint aesthetic.weighting.factor range 0 1)
  (maintenace.weighting.factor nil)
  (constraint maintenace.weighting.factor range 0 1)
  (fabrication.weighting.factor nil)
  (constraint fabrication.weighting.factor range 0 1)
  (economic.weighting.factor nil)
  (constraint economic.weighting.factor range 0 1))

(deffacts initial " "
  (main.span.length nil nil 5 10 20 30 40 60 80 100 150)
  (constraint main.span.length range 5 150)
  (total.length nil nil)
  (constraint total.length range 10 1000)
  (bridge.site nil nil)
  (constraint bridge.site oneof stream road railway valley))

;
; USER INPUT
;
(defrule USER-INPUT1 "range"
  ?rem1 <- (constraint ?att ?ctyp&range ?min ?max)
  ?rem2 <- (?att nil $?list)
  =>
  (retract ?rem1 ?rem2)
  (bind ?inp nil)
  (while (or (not (numberp ?inp)) (< ?inp ?min) (> ?inp ?max))
    (format t "Enter %s<%d-%d>:" ?att ?min ?max)
    (bind ?inp (read)))
  (assert (?att ?inp $?list)))

(defrule USER-INPUT2 "oneof"
  ?rem1 <- (constraint ?att ?ctyp&oneof $?menu)
  ?rem2 <- (?att nil $?list)
  =>
  (retract ?rem1 ?rem2)
  (bind ?inp nil)
  (while (not (member ?inp $?menu))
    (format t "Enter %s<" ?att)
    (bind ?len (length $?menu))
    (bind ?i 1)
    (while (< ?i ?len)
      (format t "%s," (nth ?i $?menu))
      (bind ?i (+ ?i 1)))
    (format t "%s>:" (nth ?len $?menu))
    (bind ?inp (read)))
  (assert (?att ?inp $?list)))

```

그 림 3.2 입력데이터의 규칙

또한 비교안에 대한 평가요인인 경제성, 시공성, 유지관리, 주행성 및 미관등은 시방서, 교량실시설계보고서, 참고문헌 및 전문가의 의견을 참고로 하여 5, 3, 1, 0의 4단계로 평가점수를 부여하였다. 이러한 각 비교안별 평가요인에 대한 평가점수는 표 3.2에서와 같이 하나의 화일로 데이터베이스화하여 보관하도록 되어 있어 효율적인 관리를 유지하도록 하였다.

표 3.2에서 [1]의 경우는 지간이 5-10m인 경우에 비교안이 RC slab와 RC rahmen으로서 이에대한 각 평가항목별 평가점수는 nil다음의 list로서 경제성, 시공성, 유지관리성, 미관 및 주행성의 순으로 되어 있다.

경제성에 대해서는 각 교종별의 지간으로부터 구한  $m^2$ 당의 회귀식을 작성하여 각 비교안을 평가하도록 되어 있고, 시공성에 대해서는 안전성, 난이도, 공기 및 가설공비등을 판정요소로 들 수 있을 것이다. 가설공비는 경제성에 포함되기 때문에 시공성이라는 면에서는 제외하고, 안전성 및 난이도는 평가방법이 대단히 어렵지만 과거의 실적수에 의해 평가하고, 공기는 각 교종별 가설공법 및 제 조건에 의해 평가를 하도록 하며, 유지관리에 대해서는 콘크리트교에 대해서는 보통 도장은 불필요하므로 유지비용은 적은 것으로 고려하여 강교보다는 좋다고 생각되고, 강교는 도장이 정기적으로 행해지는 것에 의해 콘크리트교보다는 낮게 고려하여 평가하도록 하였으며, 주행성에 대해서는 이의 우열을 판단하는 것은 대단히 곤란하지만 진동, 충격을 가능한 적게 하는 것, 즉, 신축계수가 적은 것이 주행성을 양호하게 한다. 따라서 교량을 가능한 길게 연속시킨다든지, 지간장을 길게 하는 것등이 고려된다. 본 시스템에서는 조인트수를 고려해서 판정한다.

마지막으로 미관에 대해서는 주위에 영향을 미치는 것이 대단히 중요하지만 선정요소를 고정화하는 것은 매우 어렵다. 그래서 본 시스템에서는 교종과 가설장소로부터 평가한다.

표 3.2 평가요인별 점수의 Data file (BADES-S.DB)

```

[1]
RC slab      nil 3 5 3 1 3
RC rahmen    nil 5 3 3 5 3
[END]
[2]
RC hollow.slab nil 1 1 3 5 3
RC rahmen      nil 3 3 5 5 3
RC t.beam      nil 5 5 3 3 1
[END]
[3]
STEEL simple.plate.girder nil 1 3 1 5 1
RC t.beam              nil 3 5 3 3 1
PC beam                nil 5 5 3 3 3
PC simple.girder       nil 3 3 5 3 1
[END]
[4]
STEEL simple.plate.girder nil 3 5 1 1 1
STEEL continuous.plate.girder nil 3 3 1 3 5
STEEL simple.box.girder nil 3 3 1 3 1
PC beam                nil 5 5 5 3 3
PC simple.girder       nil 3 3 5 3 1
PC simple.box.girder   nil 3 1 5 3 1
[END]
[5]
STEEL continuous.plate.girder nil 3 5 1 1 5
STEEL simple.box.girder nil 1 3 3 3 1
STEEL continuous.box.girder nil 3 3 3 5 5
PC continuous.girder nil 3 5 3 1 5
PC continuous.box.girder nil 5 1 5 5 5
[END]
[6]
STEEL continuous.box.girder nil 3 3 3 3 5
STEEL simple.truss nil 1 1 1 5 1
STEEL continuous.truss nil 1 3 1 5 5
STEEL langer.upper.deck nil 1 1 1 5 3
PC continuous.box.girder nil 5 3 5 3 5
[END]
[7]
STEEL continuous.truss nil 3 3 1 3 5
STEEL langer.through.deck nil 3 3 1 3 3
STEEL arch nil 3 1 3 5 3
PC continuous.box.girder nil 5 1 5 3 5
[END]
[8]
STEEL continuous.truss nil 3 3 1 3 5
STEEL arch nil 3 1 1 5 3
PC continuous.box.girder nil 5 1 5 3 5
[END]

```

또한 각 평가요인에 대한 중요도(weighting factor)를 그림 3.2에서 보는 비와 같이 0-1의 범위의 값을 사용자에게 직접 입력받아 이 중요도에 각 평가요인별 평가점수를 곱해서 비교안별로 점수를 합산해서 합산점수가 높은 순으로 비교안이 나타내어 지도 록 하였다. 사용자는 이 평가된 추천결과중에서 최종적인 상부구조의 형식을 선정하도록 되어 있다.

그림 3.1에 대한 전체적인 입력데이터 및 추천결과외 한 예를 보면 그림 3.3과 같다.

```

CLIPS> (reset)
CLIPS> (run)
Enter bridge.site<stream,road,railway,valley>:road

Enter total.length<10-1000>:50

Enter main.span.length<5-150>:15

Enter economic.weighting.factor<0-1>:0.9

Enter fabrication.weighting.factor<0-1>:0.7

Enter maintenace.weighting.factor<0-1>:0.5

Enter aesthetic.weighting.factor<0-1>:0.7

Enter servicebility.weighting.factor<0-1>:0.5

Candidate hollow.slab is being evaluated.
Candidate rahmen is being evaluated.
Candidate t.beam is being evaluated.
CANDIDATE[gen1]: RC-rahmen:12
CANDIDATE[gen2]: RC-t.beam:12
CANDIDATE[gen3]: RC-hollow.slab:8
Enter your candidate name<gen1> :
19 rules fired
CLIPS> (dribble-off)

```

### 그림 3.3 BADES-S의 추천결과

그림 3.3에서 보면 각 입력데이터에 대한 추천 결과에서 RC-rahmen, RC-t.beam 및 RC hollow의 순으로 점수가 되어 있으며, 사용자는 이 결과를 참고로 하여 교량의 상부구조형식을 선정할 수 있다.

## 3.2. 가설공법의 선정시스템 (BADES-C)

### 3.2.1 개요

본 시스템에서는 일단 BADES-S에서 교량 상부구조의 형식이 선정되어지면 선정된 형식에 따라 가설공법이 선정되도록 되어있으며, 여기서의 적용범위는 상부구조형식이 PC BOX 형식에 한하고 있다. 표 2.3에서의 PC교량의 가설공법을 시공특성에 따라 세부적으로 분류하면 그림 3.4와 같다. 그림 3.4에서 박스거더교 시공시에 주로 적용되는 공법은 동바리공법, 이동식 비계공법, 압출공법, 캔틸레버공법, 전진가설공법 및 프리캐스트 세그먼트공법등이고 프리캐스트 거더공법은 PC I빔이나 T빔등을 경간단위로 가설할 때 사용되는 공법이다.

국내 PC 박스거더교는 1981년에 원효대교가 캔틸레버공법으로 건설된 이후 PC 박스거더교의 건설이 급격히 증가하였으며, 동바리공법과 압출공법이 전체의 83%를 차지하고 특히 압출공법은 1985년의 금곡천교 시공이후 팔목할만한 신장세를 보이고 있다. 동바리공법의 상부구조 시공부분에서의 인건비 비율이 최근에는 65%를 상회하고, 상부구조 노임중 거푸집과 동바리 작업의 비율이 45% 정도의 수준임을 감안할 때 앞으로 재래식 동바리공법보다는 압출공법과 같은 기계화된 최신공법의 적용이 더욱 늘어날 것으로 전망된다.

### 3.2.2 선정시스템의 구축

일반적으로 PC 박스거더교의 공법의 특징중에서 경제성은 공법선정시 중요한 요소가 된다. 공법별 경제성은 경간장, 교장, 교고 및 하부조건등과 밀접한 관계가 있고, 재래식 동바리공법을 제외한 현대식 기계화된 공법에서는 경간장과 교장이 특정공법

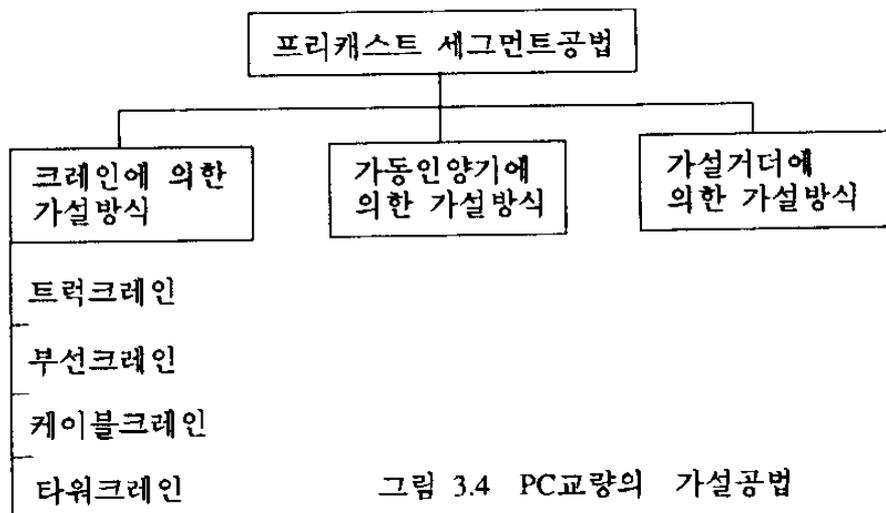
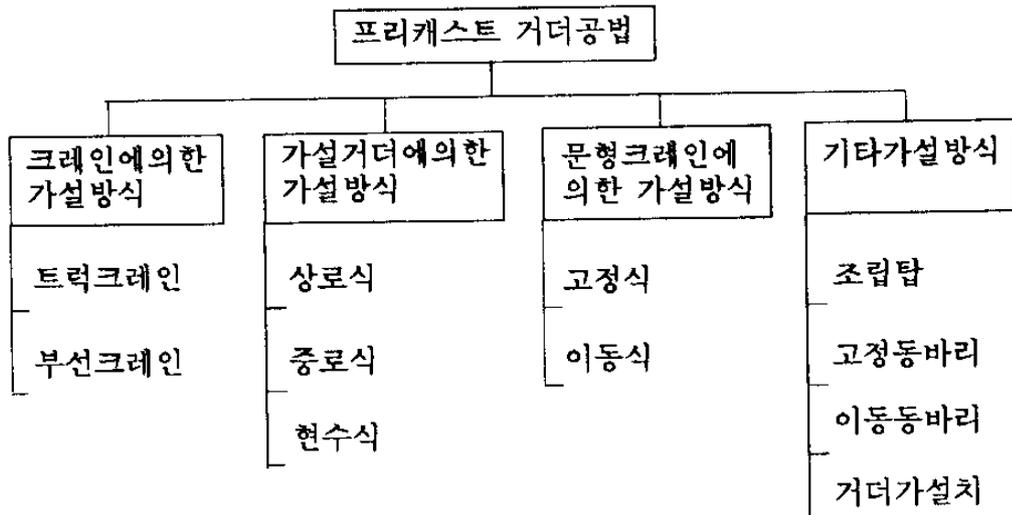
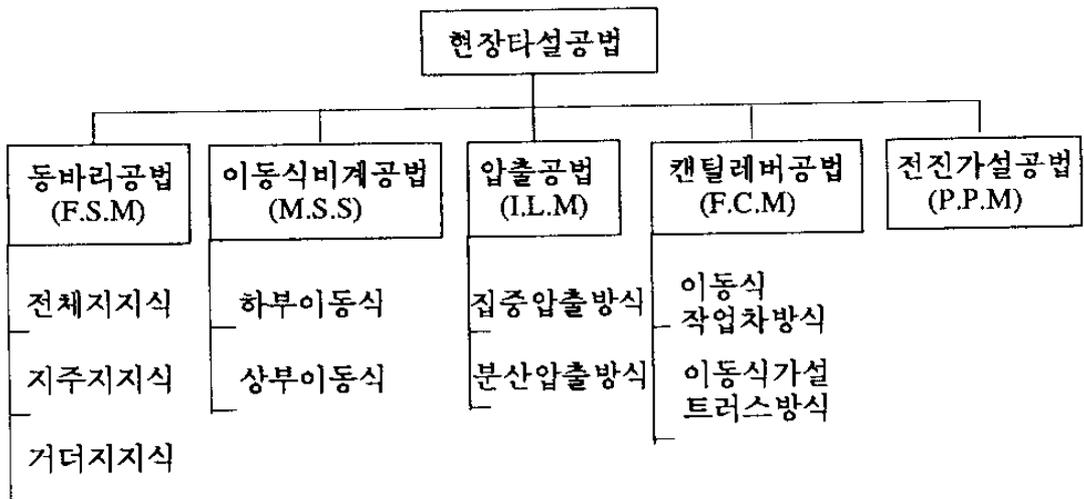


그림 3.4 PC교량의 가설공법

의 경제성 여부에 가장 큰 영향을 미치게 된다. 표 3.3에서는 국내의 시공실적과 외국의 문헌분석 및 시공실적을 토대로 각 공법별로 적용할 수 있는 경간장을 표시하였다.

표 3.3에서 보는 바와 같이 동바리 공법은 최대 경간장이 30m에서 60m 사이인 교량에 사용되었으며, 압출공법은 40m에서 60m 사이인 교량에, 캔틸레버공법은 95m에서 125m 사이인 교량에, 이동식 비계공법은 50m 교량에 각각 적용되었다는 것을 알 수 있다. 또한 표 3.4에서는 국내의 시공실적과 외국의 문헌분석 및 시공실적을 토대로 각 공법별로 적용할 수 있는 교장을 표시하였다.

표 3.4에서 보는 바와 같이 동바리공법은 50m에서 500m 사이의 교량에 적용되며 주로 100m 내외의 교량에 많이 사용되고, 압출공법은 200m에서 1000m 사이의 교량에 적용되며 주로 300m에서 700m 사이의 교량에 많이 사용되고, 캔틸레버공법은 300m에서 1000m 사이의 교량에 적용되고, 이동식 비계공법은 100m 이상의 교량 시공에 이용된다는 것을 알 수 있다.

또한 최대 경간장과 교장뿐만 아니라 현장의 입지조건도 가설공법을 선정할 때 중요한 고려요소가 된다. 현장의 입지조건을 중심으로 한 공법선정개념을 순서도로 표시하면 그림 3.5와 같다.

본 시스템에서는 이상에서 적용성, 경간장, 교장 및 입지조건에 대한 사항을 고려한 지식베이스를 구축하여 선정된 상부구조형식에 대한 가설공법을 선정하도록 하였다. 다음 그림 3.6은 BADES-S에서 선정된 상부구조형식에 대해서 가설공법이 선정되는 입력데이터 및 추론결과를 나타낸다.

그림 3.6에서 BADES-S 시스템에서 CANDIDATE[gen1]:PC-continuous box girder 형식이 상부구조형식으로 선정되었다면, 그 후의 BADES-C의 입력데이터를 받아 추론한 결과 최종적인 가설공법은 Precast Segment 공법으로 선정된 결과를 나타낸다.

표 3.3 공법별 적용 경간장

공법 \ 경간장(m)	20	40	60	80	100	120	140	160	180	200	220	240	260
	동바리공법												
이동식비계 공법 압출공법													
캔틸레버 공법 전진가설 공법 프리캐스트 거더공법 프리캐스트 세그먼트공법													

----- 적용가능 경간장

———— 시공실적 경간장

표 3.4 공법별 적용 교장

공법 \ 교장(m)	100	200	300	400	500	600	700	800	900	1000
	동바리공법									
이동식비계 공법 압출공법										
캔틸레버 공법 전진가설 공법 프리캐스트 거더공법 프리캐스트 세그먼트공법										

----- 적용가능 교장

———— 시공실적 교장



```

CLIPS> (reset)
CLIPS> (run)
Enter bridge.site<stream,road,railway,valley>:road

Enter total.length<10-1000>:500

Enter main.span.length<5-150>:50

Enter economic.weighting.factor<0-1>:1

Enter fabrication.weighting.factor<0-1>:0.8

Enter maintenace.weighting.factor<0-1>:0.7

Enter aesthetic.weighting.factor<0-1>:0.7

Enter servicebility.weighting.factor<0-1>:0.5

Candidate continuous.plate.girder is being evaluated.
Candidate simple.box.girder is being evaluated.
Candidate continuous.box.girder is being evaluated.
Candidate continuous.girder is being evaluated.
Candidate continuous.box.girder is being evaluated.
CANDIDATE[gen1]: PC-continuous.box.girder:16
CANDIDATE[gen2]: STEEL-continuous.box.girder:12
CANDIDATE[gen3]: PC-continuous.girder:12
CANDIDATE[gen4]: STEEL-continuous.plate.girder:10
CANDIDATE[gen5]: STEEL-simple.box.girder:7
Enter your candidate name<gen1> :
Is it possible to place temporary bent for erection[y/n]?n

Is there a yard for fabrication near the road[y/n]?n

Is there a yard for fabrication near the site[y/n]?y

Can the crane or the like enter into the working area[y/n]?y

Is the height suitable for crane erection[y/n]?y

Can we select the self-propelled crane according to
the member size and member weight[y/n]?y

>>** You can select the PC BOX GIRGER type by PRECAST
    SEGMENT construction method.
30 rules fired
CLIPS> (dribble-off)

```

### 그 립 3.6 BADES-C의 추론결과

### 3.3 기초형식의 선정시스템 (BADSE-F)

#### 3.3.1 개요

일반적으로 한개의 기초형식에 대해 여러 지반개량공법에 의해 많은 기초형식이 존재하지만 그것을 전부 고려하면 시스템이 매우 커지게 되어 한정된 시간내에 이루는 것은 곤란하기 때문에 본 시스템에서는 기술적인 사항에 중점을 두어 크게 직접기초, 말뚝기초, 피어기초 및 케이슨기초로 분류하고(표 2.4참조), 그 안에서 다시 형식을 분류하는 방식으로 구축하였다.

일반적으로 기초형식을 선정하는 순서는 다음과 같다.

(1) 기초구조물이 지지하는 상부구조물의 재원 파악

(2) 지반조건(개략의 상황) 조사

(3) 기초형식에 대한 제1차 예상

(4) 지반조사

\* 직접기초 - 표준관입시험, 시료채취 및 역학시험, 평판재하시험

\* 말뚝 및 케이슨기초 - 표준관입시험, boring, 시료채취 및 역학시험

압밀층의 실내압 및 특성시험

(5) 시공환경조사 - 공해방지조례에 의한 제한, 부식환경

(6) 제 1차 형식선정

(7) 선정대상중에서 다음 조건에서 비교검토대상을 선정

\* 상부구조의 하중규모

\* 지지층의 깊이

\* footing의 지하수면에서의 깊이

\* 작업면적의 대소

- \* 공기의 우선여부
- \* 지하수의 수두, 유속
- (8) 공사비, 공기의 산출
- (9) 최종적인 형식결정

### 3.3.2 선정시스템의 구축

본 시스템에서는 각 기초형식에 대한 평가요인으로서 크게 구조물의 특성, 지형 및 지질조건, 시공조건 및 환경조건에 의해 5, 3, 1, 0의 4단계의 점수를 부여하여 평가하도록 되어있다. 각 형식별 평가요인에 대한 평가점수는 표 3.5와 같이 하나의 화일로 데이터베이스화하여 보관하며, 또한 사용자에게 각각의 평가요인에 대해서 질문을 하여 해당되는 요인을 선정하는데 입력데이터의 형태는 일가 선택형으로 나타내었으며, 선정된 평가요인들의 각 기초형식별로 최종합계 평가점수가 높은 순서로 추론결과가 나타나게 되었다. 이에 대한 입력데이터 및 추론결과는 그림 3.7과 같다.

표 3.5에서 end.bearing.type.Df:0-5m에서 friction.type까지는 지지방식에 관한 사항이며, small.span.length:below 20m에서 large까지는 하중규모로서 연직하중과 수평하중을 나타내며, flatted:below:30degree에서 concave.and.convex까지는 지지충면의 상태를 나타내고, 그 후의 사항들은 지하수의 상태, 지표면의 상태 및 환경조건등을 나타낸다. 또한 각 점수에 대한 list의 순서는 직접기초, RC, PC 및 Steel pile, RCD, BENOTO 및 Earth drill pier, open 및 pneumatic caisson순으로 되어 있다.

표 3.5 평가요인별 점수의 Data file (BADES-F.DB)

```

[end.bearing.type.Df:0~5m]
5 3 1 1 0 0 1 3 1 0
[END]
[end.bearing.type.Df:5~15m]
3 3 3 3 1 3 3 5 5 5
[END]
[end.bearing.type.Df:15~25m]
1 3 5 5 5 5 3 5 5 5
[END]
[end.bearing.type.Df:25~40m]
0 0 3 5 5 3 1 0 5 3
[END]
[end.bearing.type.Df:40~50m]
0 0 1 5 5 1 0 0 1 1
[END]
[end.bearing.type.Df:50~60m]
0 0 0 5 5 0 0 0 1 0
[END]
[partial.friction.type]
1 5 5 3 3 3 3 3 3 3
[END]
[friction.type]
0 5 5 5 0 0 0 0 0 0
[END]
[small.span.length:below20m]
5 3 5 3 3 3 3 3 1
[END]
[medium.span.length:20~50m]
5 3 5 5 5 5 3 3 5 3
[END]
[large.span.length:above50m]
5 1 3 3 3 3 3 3 5 5
[END]
[small]
5 5 5 3 3 3 5 3 3 1
[END]
[large]
5 1 3 3 3 3 3 3 5 5
[END]
[flatted:below30degree]
5 5 5 5 5 5 5 5 5 5
[END]
[inclined:above30degree]
5 1 3 3 1 5 1 5 1 3
[END]
[concave.and.convex]
5 1 3 3 5 5 3 5 3 3

```

(표 3.5 계속)

```
[END]
[cray.Nvalue:below4]
1 3 5 5 5 5 5 5 5
[END]
[cray.Nvalue:4~10]
1 1 3 5 5 5 3 3 5 3
[END]
[cray.Nvalue:10~20]
1 0 1 3 3 5 3 3 3 3
[END]
[cray.Nvalue:above20]
3 0 0 0 3 3 1 1 1 3
[END]
[sand.Nvalue:below15]
3 1 3 5 5 5 5 5 5
[END]
[sand.Nvalue:15~30]
1 0 1 3 5 5 3 5 3 5
[END]
[sand.Nvalue:above30]
0 0 0 1 3 3 1 3 1 5
[END]
[loose.sand]
3 1 3 5 3 1 1 3 1 5
[END]
[no.gravel,float.stone]
5 3 3 5 5 5 5 5 5
[END]
[gravel:below10cm]
5 0 0 1 3 3 1 3 5 5
[END]
[gravel:10~30cm]
5 0 0 1 0 1 0 3 3 3
[END]
[gravel:above30cm]
3 0 0 0 0 1 0 1 1 3
[END]
[G.W.L.is.above.the.bottom.of.the.footing]
1 3 5 5 5 5 3 0 3 5
[END]
[G.W.L.is.below.the.bottom.of.the.footing]
3 5 5 5 5 5 1 5 5
[END]
[G.W.L.is.below.the.end.bearing.pile]
5 5 5 5 5 5 3 5 5
[END]
[pressed.ground.water.is.0~2m.from.the.surface.of.the.earth]
1 5 5 5 1 1 1 0 3 5
[END]
[pressed.ground.water.is.above2m.from.the.surface.of.the.earth]
0 3 5 5 0 0 0 0 1 3
[END]
[movable.ground.water.velocity.above3m/min]
0 3 5 5 0 0 0 0 1 3
[END]
```

(표 3.5 계속)

```
[flatted:below10degree]
0 0 0 5 5 3 1 0 1 3
[END]
[inclined:above10degree]
3 3 3 3 3 3 3 3 3 3
[END]
[concave.and.convex]
1 1 1 1 1 1 1 1 1 1
[END]
[site.on.the.water.depth.of.water:below5m]
3 3 5 5 5 0 0 0 5 5
[END]
[site.on.the.water.depth.of.water:above5m]
0 1 1 1 1 0 0 0 5 5
[END]
[yes2]
5 0 0 0 5 3 3 5 3 3
[END]
[no2]
5 5 5 5 5 5 5 5 5 5
[END]
[yes3]
3 0 0 1 5 5 3 1 1 3
[END]
[no3]
5 5 5 5 5 5 5 5 5 5
[END]
[yes4]
5 1 1 1 3 1 1 5 3 3
[END]
[no4]
5 5 5 5 5 5 5 5 5 5
[END]
[yes5]
1 3 5 5 5 3 3 0 5 0
[END]
[no5]
5 5 5 5 5 5 5 5 5 5
[END]
```

```

CLIPS> (reset)
CLIPS> (run)
Select support.type :
  1] end.bearing.type.Df:0~5m
  2] end.bearing.type.Df:5~15m
  3] end.bearing.type.Df:15~25m
  4] end.bearing.type.Df:25~40m
  5] end.bearing.type.Df:40~50m
  6] end.bearing.type.Df:50~60m
  7] partial.friction.type
  8] friction.type
Enter your selection<1-8> :3

Select structural.vertical.load :
  1] small.span.length:below20m
  2] medium.span.length:20~50m
  3] large.span.length:above50m
Enter your selection<1-3> :2

Select structural.lateral.load :
  1] small-
  2] large
Enter your selection<1-2> :1

Select bearing.layer.state :
  1] flatted:below30degree
  2] inclined:above30degree
  3] concave.and.convex
Enter your selection<1-3> :1

Select bearing.layer.soil.condition :
  1] clay.Nvalue:below4
  2] clay.Nvalue:4~10
  3] clay.Nvalue:10~20
  4] clay.Nvalue:above20
  5] sand.Nvalue:below15
  6] sand.Nvalue:15~30
  7] sand.Nvalue:above30
  8] loose.sand
  9] no.gravel,float.stone
  10] gravel:below10cm
  11] gravel:10~30cm
  12] gravel:above30cm
Enter your selection<1-12> :2

Select underground.water.state :
  1] G.W.L.is.above.the.bottom.of.the.footing
  2] G.W.L.is.below.the.bottom.of.the.footing
  3] G.W.L.is.below.the.end.bearing.pile
  4] pressed.ground.water.is.0~2m.from.the.surface.of.the.earth
  5] pressed.ground.water.is.above2m.from.the.surface.of.the.earth
  6] movable.ground.water.velocity.above3m/min
Enter your selection<1-6> :2

```

그림 3.7 BADES-F의 추론결과

(그 림 3.7 계속)

```
Select surface.layer.state :  
  1] flatted:below10degree  
  2] inclined:above10degree  
  3] concave.and.convex  
Enter your selection<1-3> :1
```

```
Select environmental.condition1 :  
  1] site.on.the.water.depth.of.water:below5m  
  2] site.on.the.water.depth.of.water:above5m  
Enter your selection<1-2> :1
```

```
Select environmental.condition2.consider.low.vibration.and.low.noise :  
  1] yes2  
  2] no2  
Enter your selection<1-2> :1
```

```
Select environmental.condition3.consider.influence.adjacent.structure :  
  1] yes3  
  2] no3  
Enter your selection<1-2> :1
```

```
Select environmental.condition4.narrow.the.working.area :  
  1] yes4  
  2] no4  
Enter your selection<1-2> :1
```

```
Select environmental.condition5.consider.damage.of.gas :  
  1] yes5  
  2] no5  
Enter your selection<1-2> :1
```

```
Candidate is being generated.  
CANDIDATE[gen5-R.C.D.pier]:56  
CANDIDATE[gen9-open.caisson]:46  
CANDIDATE[gen4-steel.pile]:45  
CANDIDATE[gen6-BENOTO.pier]:43  
CANDIDATE[gen3-pc.pile]:39  
CANDIDATE[gen10-pneumatic.caisson]:39  
CANDIDATE[gen1-shallow.foundation]:37  
CANDIDATE[gen7-EARTH.DRILL.pier]:35  
CANDIDATE[gen8-deep.pier]:31  
CANDIDATE[gen2-rc.pile]:29  
35 rules fired  
CLIPS> (dribble-off)
```

#### 4. 결론

전문가 시스템의 개발은 전문가 시스템 지원용 도구를 이용하여 지식베이스를 구축하여 이를 이용함으로써 문제를 해결하는 것으로서 지식의 표현형태가 다양하고 또한 지식전체를 체계화하기 쉽다. 이와 같은 점을 고려하면 토목설계의 업무는 거의가 경험적지식 위에서 성립되었기 때문에 전문가 시스템의 응용분야가 대단히 넓다고 생각된다. 현재 토목설계를 대상으로 한 많은 업무가 전문가 시스템의 적용대상으로 거론되는 것과 함께 구체적인 전문가 시스템 개발의 검토단계에 들어가고, 또한 더욱 발전된 단계로 개발되고 있다. 이러한 개발의 결과 토목설계용 전문가 시스템은 적용가능성이 대단히 높은 것으로 입증되기 시작하고 있다.

본 연구에서도 전문가 시스템 개발용 도구인 K-CLIPS를 이용하여 예비설계 시스템의 구축을 한 것으로 토목분야에 있어서는 이 같은 형식, 공법의 선정이라는 전문가 시스템 구축이외에도 설계도면의 검증, 시방서와의 적합, 관련법규와의 검증등과 같이 한개의 제시된 결론 및 가설을 미리 준비된 지식을 토대로 진단, 판단하는 문제등에 대해서 전문가 시스템을 도입해서 해결하고자 하는 범위가 있다.

일반적으로 토목분야에 있어서의 전문가 시스템은 단독시스템이 아니고, 지원시스템으로서 기존의 시스템에 조합되어 가는 것에 의해 적용가능성이 넓어진다고 생각된다. 본 연구에 있어서도 교량의 예비설계 전문가 시스템의 개발은 한개의 독립된 시스템으로서 개발하는 것으로 끝나지 않고, CAD 시스템이나 적산시스템 및 DBMS(Data Base Management System)등을 조합해 가는 것으로 장기간에 걸쳐 축적된 소프트웨어를 유효하게 활용가능하다는 점에서 매우 큰 가치가 있다고 생각된다. 또한 실용적인 전문가 시스템의 차원에서 단지 추론결과를 제시하는 것 이외에 왜 그러한 결과가 선정되었는가? 라는 등의 추론과정의 출력이나, 추론결과의 근거의 출력이 설명이 중

요하다고 생각되며, 추론결과의 확실성을 사용자에게 제공하기 위하여 certainty factor 등의 확률적 사항에 대한 판한 표현방법을 고려해야겠다.

마지막으로 상부구조형식의 선정시 평가항목들은 예비설계에 있어서 반드시 의식적으로 고려하지 않아도 될 지 모르지만, 경험적인 종합판단에서는 숙련기술자의 판단기준의 잠재화된 부분을 추출할 필요가 있어 포함할 필요가 있으며, 그것이 곤란한 경우에는 잠재화된 부분을 현재부분으로 잘 처리하지 않으면 안된다. 또한 평가항목에 관해서도 5항목 이외에 안전성, 환경적합성등을 고려하지 않으면 안될 것이다. 또한 지식의 애매모호한 과정등은 기술자로서의 경험뿐만 아니라 그 사람의 성격이 크게 영향을 주므로 시방서나, 문헌등에서도 애매한 과정을 추출하는 것은 매우 어렵다. 그래서 지금부터 충실한 시스템으로서 완성되기 위해서는 전문가와의 지속적인 대화와 앙케이트등의 조사에 의해 더욱 지식을 광범위하게 수집하는 것이 필요하다.

## 5. 참고 문헌

1. P.Harmon and D. King, Artificial Intelligence in Business : Expert System, John Wiley and Sons, New York, 1985.
2. Hayes-Roth F, Waterman D.A, Lenat D.D., Building Expert System, Addison-Welsey, 1983.
3. D.A.Waterman, A Guide to Expert System, Addison-Wesley, US, 1986.
4. B.G.Buchanan and E.H.Shortlife, Rule-based Expert Systems, Addison-Wesley, US, 1984.
5. S.Weiss and C.Kulikowski, A Practical Guide to Designing Expert Systems, Rowman and Allandheld, Totowa,NJ, 1984.
6. C.Towmsend and D.Feucht, Design and Programming Personal Expert Systems, TAB, 1986.
7. L.Brown, R.Farrel, and E.Kant etc., Programming Expert System in OPS5, Addison-Welsey, 1986.
8. R.Forsyth, Expert systems : Principles and Case Studies, Chapman and Hall Computing, 1984.
9. C.K. Choi and E.d.Kim, A Preliminary model of I-BUILDS: An Intelligent Building Design system, Proceedings of 2nd International Conference on Application of A.I. in Eng. Problems, Cambridge Massachusetts, U.S.A., 1987.
10. E.H. Shortliffe, Computer-Based Medical Consultations : MYCIN, American Elsevier, N.Y., 1976.
11. R.O.Duda et al, A computer-Based Consultant for Mineral Exploration, Final Report SRI Project 6415, SRI International, 1979.

12. J. Mcdermodott, R1 : A Rule-Based Configure of Computer system, Technical Report CMU-CS- 80-119, Department of computer science, Carnegie-Mellon Univ. Pittsburgh, 1980.
13. M.L. Maher and S.J. Fenves, HI-RISE : A Knowledge-Based Expert System for the Preliminary Structural Design of Hi Rise Buildings. Report No. R-85-146, department of Civil Eng., Carnegie-Mellon Univ., Pittsburg, PA, 1985.
14. J.S. Bennet and R.S. Englemore, SACON : A Knowledge-Based Consultant for Structural Analysis, Proceedings of the Sixth International Joint Conference on A.I., Tokyo, August, pp.47-49, 1979.
15. R.J.Melosh, a. Verma and H-Y.Chang, SAGE : A Consultant for Earthquake Engineering, Third Chautauqua on Productivity in Eng. and Design(Ed.H.Shaeffler) Kiawah Island, SC, Millford, NH:Wallace Press, 1984.
16. J.Morrill and D.Wright, A Knowledge-Based Program for Diagnosing metallurgical failures, J. Comput. Struct. 24(2) 305-311, 1986.
17. R.E. Levitt, HOWSAFE : A Microcomputer-based expert system to evaluate the safety of a construction firm. In : Expert systems in civil Eng. (Eds. C.N. Kostem; M.L.Maher). New York: American Society of Civil Engineers, pp 55 - 66, 1986.
18. T.J. Ross, F.S. Wong et al., An expert system for damage assessment of protective structure, Expert system in civil Eng. (Eds. C. N. Kostem, M.L Maher) New York, ASCE, pp.109-120, 1986.
19. Andrew K.C. Wang and Tung Au, Parallel Processing Approach to Bridge Planning, Journal of Structural Division, ASCE, vol.97, no.ST1, 1971.
20. Mrinmay Biswas and James G. Welch, BDES : A Bridge design Expert System, Engineering with Computers 2, pp.125-136, 1987.

21. T. Pagnoni, Expert System for Bridge Engineering, International Bridge Conference, Pittsburgh, Pennsylvania, 1985.
22. H. Furuta and N. Shiraisha, Application of Expert System to Structural Engineering, Proceedings of Korea-Japan Joint Seminar, Seoul, Korea, 1988.
23. F.Chehayeb and J.Conner, GEPSE-A Computer Environment for Engineering Problem Solving, Research Report R86-11, Dept. of Civil Engineering, MIT, 1985.
24. 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.
25. Mark H. and Richer, An Evaluation of Expert System Development Tools, Expert System, July, Vol.3, No.3, 1986.
26. Phillip J.L., William J.G., and Del D., Expert System Tools for Civil Engeneering Applications, Expert Systems in Civil Engineering, 1986.
27. Joseph C. Giarrantano, CLIPS User's Guide, 1987.
28. Sobelman and Krekelberg, Advanced C : Techniques and Applications, QUE, 1985.
29. M.Waite, S.Prate, and D.Martin, C Primer Plus, Howard W.Sams & Co.,Inc., 1984.
30. B.W.Kernigham and D.M.Ritchie, The C programming Language, Prentice-Hall, 1978.
31. Herbert Schilt, C : The Complete Reference, McGraw-Hill, 1987.
32. J.M. Barker, Post-Tensioned Box Girder Bridge Manual, Post-Tensioning Institute, 1978.
33. 토목구조물 설계제산에 시리즈, 건설도서, 1988.

34. 신기화, 최신도로 설계요령, Vol.3,no.2, pp.1-8, 1980
35. 최무혁, 건축계획 과정에서의 평가모델 설정에 관한 연구, 대한건축 학회지, 24권 96호, 1980.
36. 유신설계공단 etc., 중부고속도로건설 실시설계 보고서, 한국 도로 공사, 1985.
37. 유신설계공단,etc., 호남고속도로건설 실시설계 보고서, 한국 도로 공사, 1986.
38. 유신설계공단 한강복선철교 교량시설조사설계 기본설계 보고서, 철도 청, 1986.
39. 한국과학기술원, 인공지능을 이용한 고도의 구조해석/설계용 전문가시스템의 개발(I,II), 과학기술처, 1988.

## 부록 1 : K-CLIPS의 사용법

## 목 차

1. 서론
2. K-CLIPS의 개요
  - 2.1 사실
  - 2.2 규칙
  - 2.3 K-CLIPS의 실행
  - 2.4 보고서에서의 사용기호
  - 2.5 타 프로그램 언어와의 연계성
  - 2.6 고도의 프로그램 기법
3. 패턴매칭
  - 3.1 LHS 조건문
    - 3.1.1 패턴매치
      - a) 문자패턴
      - b) 와일드카드
      - c) 변수
    - 3.1.2 패턴의 조절
      - a) 패턴바인딩
      - b) 패턴의 제한
    - 3.1.3 규칙의 특성
      - a) 규칙의 우선권 선언
  - 3.2 RHS 실행문
    - 3.2.1 사실의 제어

- a) 새로운 사실을 만들
- b) fact-list에서 사실을 제거
- c) 바인딩의 사용
- d) 다중필드함수

### 3.2.2 K-CLIPS의 입력/출력 시스템

- a) 입출력 지시명
- b) 일반적인 입출력함수

### 3.2.3 그 밖의 실행문

- a) system
- b) K-CLIPS의 중지
- c) 심볼의 생성
- d) call
- e) if..then..else
- f) while
- g) 문자열의 결합

## 3.3 계층적인 지식을 표현하는 기능

### 3.3.1 클래스의 정의

### 3.3.2 클래스의 슬롯관리

- a) 슬롯 생성
- b) 슬롯 제거
- c) 슬롯 추가
- d) 슬롯 수정

### 3.3.3 클래스의 패턴매치

a) 클래스의 패턴문장

b) 클래스의 조건함수

### 3.4 함수를 정의하는 기능

3.4.1 함수의 정의

3.4.2 그밖의 보조함수

### 3.5 K-CLIPS의 계산함수

3.5.1 기본함수

a) 기본적인 산술함수

b) 확장된 산술함수

### 3.6 K-CLIPS의 인터페이스 함수

3.5.1 윈도우함수

3.5.2 메뉴함수

3.5.3 그래픽함수

### 3.7 작업수행시 고려할 사항

3.6.1 패턴의 순서

3.6.2 fact-list와 사실과의 비교

## 4. 사용자를 위한 인터페이스 기능

### 4.1 K-CLIPS 명령문

4.1.1 환경제어 명령문

4.1.2 시스템 상태 명령문

4.1.3 디버깅 명령문

4.1.4 메모리 관리 명령문

4.1.5 그밖의 명령문

## 4.2 HELP 시스템

### 4.2.1 온 라인 HELP 시스템

- a) HELP 기능 사용방법
- b) HELP 파일의 조사

### 주.1 K-CLIPS의 설치

### 주.2 K-CLIPS의 SYNTAX

### 주.3 K-CLIPS의 MicroEmacs 에디터

### 주.4 K-CLIPS의 런-타임 프로그램

### 주.5 K-CLIPS의 에러메시지

## 1. 서론

전문가 시스템이 최근들어 상당히 주목을 받고 있으나, 연구 개발된 전문가시스템을 보급시키기 위해서는 불편한 점이 상당히 뒤따른다. 이러한 사항은 LISP언어의 문제가 큰 것에 기인하는 데, 우선 LISP은 기존의 컴퓨터에 대부분이 내장되어 있지 않으며, 둘째는 LISP를 사용하여 만든 전문가시스템 개발용 도구와 전용기계가 너무 비싸며 또한 LISP언어는 기존의 언어와는 통합이 어려운 점들을 들 수 있다.

이러한 문제점을 해결하기 위하여 NASA/Johnson Space Center의 AIS(Artificial Intelligence Section)에서는 CLIPS(C language Integrated Production System)이라는 C로 만든 전문가시스템 개발용 도구를 개발하였다. 이 CLIPS는 높은 이식성과 낮은 가격, 그리고 다른 시스템과의 쉬운 통합등을 목적으로 개발되었다. CLIPS의 기본적인 특징으로는 Rete Algorithm을 이용한 전진추론 규칙기반 시스템이다. 또한 CLIPS는 여러 종류의 하드웨어에서 적절히 수행될 수 있어 대부분의 NASA 분소, 대학, 그리고 개인 연구소에서 사용되고 있다.

K-CLIPS는 기존의 CLIPS가 제공하는 지식표현의 기법 이외에 계층적인 지식을 보다 효율적으로 처리하는 프레임 표현방식을 가능케하고 또한 사용자와의 인터페이스를 위한 그래픽 라이브러리와 윈도우/메뉴 시스템을 추가한 CLIPS의 KAIST Version이다.

## 2. K-CLIPS의 개요

### 2.1 사실 (facts)

사실은 K-CLIPS 시스템의 데이터 표현의 기본 형태이며, 각 사실은 fact-list 라 불리우는 사실목록에 위치하게 되고 이것은 정보의 기본단위이다. 규칙(rules)은 사실들의 유무에 기초하여 적용되는 데, 한 개의 사실은 빈칸(space)으로 분리된 임의 갯수의 필드(field)로서 구성되며 사실의 갯수는 오직 컴퓨터의 기억용량에 따라 제한된다. 사실들은 실행(run)을 하기전에 상위레벨에서 fact-list에 추가될 수 있고 또한 규칙의 실행(action)에 따라 추가 또는 삭제될 수 있다. 사실들은 이미 fact-list에 기록이 되어 있는 것과 똑같은 때는 새로이 추가되는 사실들은 무시된다.

사실내의 각 필드는 숫자(number), 단어(word) 또는 문자열(string)중의 하나가 될 수 있다.

1) 숫자 필드 : 숫자는 1 - 0, 소숫점, 부호(+ 또는 -) 그리고 가끔 부호와 함께 표시되는 지수표현의 e 등으로만 구성된 것이다. 모든 숫자는 소숫점이 주어져 있는지에 관계없이 실수형태로 저장된다. 그러므로 유효숫자의 크기는 기계의 종류에 따르며 반올림오차(roundoff error)도 마찬가지이다.

예) 237    15.0    +12.9    -32.3e-7

2) 단어 필드 : 단어는 알파벳 문자로 시작되고 뒤따르는 어떠한 문자로도 구성될 수 있는 문자집합을 일컫는다. 즉 단어는 특별한 기능을 가진 기호 (<, (, ), :, ;, ^, 공란(space) 등)를 제외하고는 감탄부호(!)에서 Tilde(~)에 이르기까지의 모든 ASCII문자를 사용하는 것이 가능하다. 단지 유의할 것은 K-CLIPS는 케이스 센시티브(case sensitive)하므로 대문자와 소문자는 다르게 처리된다.

예) fool Hello p378-hi KAIST

3) 문자열 필드 : 문자열은 인용부호(")로 둘러싸인 문자의 집합을 일컫는다. 단 인용기호 그 자체는 백슬래시(backslash, \)를 앞에 뒀으로서 문자열에 포함시킬 수 있다. 이러한 문자열은 동등한 문자열에 대해서만 매칭(matching)이 가능하다.

예) "fool" "a and b" "number 1" "a\"quote"

K-CLIPS의 사실은 이러한 세개의 필드를 사용하여 임의로 구성될 수 있다. 프로그램의 편의를 위하여 fact-list의 첫번째 사실은 reset 이라는 명령문에 의하여 자동적으로 (initial-fact)라는 이름으로 입력하게 되는데 이는 매칭하는 데 사용되거나 또는 사실을 제거할 때 사용된다.

초기의 여러가지 사실들은 fact-list에 추가하기위한 명령문으로는 deffacts가 있는데, 이렇게 입력된 사실들은 다른 사실처럼 제거되거나 패턴매칭(pattern matching)될 수 있으며 이들은 또한 reset 명령문 이후에 자동적으로 fact-list에 입력된다.

[Syntax]

(deffacts <name> ["<comment>"]

[ ... ] )

여기서 <name>는 사실의 집합을 구분짓기 위한 이름으로 단어로 이루어진다. 코멘트는 선택적으로 사용되고 인용부호로 둘러싸여 사용된다. deffacts의 수와 하나의 deffacts문에서 입력되는 사실의 수는 제한이 없다.

예) (deffacts initial-facts "This is an example"

(A is B)

(B is C))

## 2.2 규칙(rules)

K-CLIPS의 지식표현의 기본되는 방법은 규칙이다. 규칙은 조건(conditions)부분과 조건이 만족이 될 경우에 취해질 행위(actions)부분으로 구성된다. 전문가시스템을 개발하려는 사람은 관심분야의 문제를 풀기 위한 방법으로 규칙을 정의하게 되는데, 전문가시스템의 모든 규칙을 통틀어 지식베이스(knowledge base)라 한다. K-CLIPS는 시스템의 현재상태에 규칙을 적용시키고 행위부분을 실행하는 추론엔진을 제공한다. 현재의 상태는 fact-list로서 표현된다.

이러한 규칙은 defrule에 의하여 정의되는데 한개의 규칙은 적어도 한개의 조건문과 행위문을 가져야 한다. 그리고 규칙이 가질 수 있는 조건문과 행위문의 갯수에 대해서는 제한이 없다.

### [Syntax]

```
(defrule <name> ["<comment>"]  
  [ ...          ]  
  =>  
  [ ...          ]  
)
```

여기서 <name>은 규칙의 이름이며 word로 표현된다. 코멘트(comment)는 따옴표로 둘러싸인 어떠한 문자열이 되어도 좋다. LHS는 규칙의 조건을 나타내는 한개 또는 그 이상의 패턴으로 구성되며 이들의 사이에는 묵시적으로 and에 의하여 연결된

다. RHS는 규칙의 LHS가 만족이 될 때 수행되는 한개 또는 여러개의 행위문으로 이루어진다. LHS와 RHS를 구분하기 위하여 =>이 사용된다.

예)

```
(defrule example-rule "This is an Example"
```

```
  (A is B)
```

```
  (B is C)
```

```
  =>
```

```
  (assert (A is C)))
```

### 2.3 K-CLIPS의 실행(Execution)

규칙으로 지식베이스가 구축되고 초기의 fact-list가 준비되면 K-CLIPS는 규칙을 실행시키기 위한 준비를 한다. 이전의 순차적인 언어(procedural programming language)는 프로그램의 시작, 끝 그리고 일련의 중간과정이 프로그래머에 의하여 명시적으로 정의되어 있지만 K-CLIPS에서는 이러한 순서가 이전의 프로그램과 동일하지 않다. 즉, 지식(규칙)과 데이터(사실)을 분리하여 K-CLIPS가 제공한 추론엔진에 의하여 실행이 이루어지게 된다. 이러한 실행순서는 다음과 같다.

(첫 단계) 조건이 만족되는 규칙을 찾기 위하여 지식베이스를 조사한다.

(둘째 단계) 조건이 만족되는 규칙은 agenda라는 규칙의 집합체에 스택(stack)의 형식으로 저장된다. 이때 만약 agenda에 새로이 추가되는 규칙이 이전에 있던 규칙보다 우선순위가 높으며 이전 규칙의 상위에 위치된다.

(셋째 단계) agenda의 제일 상단에 있는 규칙을 선택하여 RHS 부분을 실행시킨다.

그러면 RHS 결과로 생긴 새로운 데이터의 상태에 따라 새로운 규칙들이 agenda에 추가되거나(activated) 제거되게(deactivated) 된다. 이러한 과정은 반복되어 모든 규칙이 작용되거나(fire) 규칙의 작용횟수의 제한에 도달하거나 규칙의 RHS에서 halt 함수를 만날 때까지 진행된다.

## 2.4 보고서에서의 사용기호

본 보고서에서 쓰여지는 심볼에 대한 약속은 다음과 같다. <name>과 같은 한 개의 '<' 으로 둘러싸인 심볼은 사용자에게 의하여 정의될 수 있는 하나의 필드를 나타낸다. ["< >"]로 둘러싸인 부분은 옵션으로 추가될 수 있다는 것을 의미한다. 기타 공백문자는 각 필드를 분리시키는 역할을 하며 K-CLIPS는 세미콜론(;)에 뒤따르는 모든 문장을 무시하게 되므로 프로그램의 설명에 유용하게 쓰일 수 있다.

## 2.5 타 프로그램 언어와의 연계성

전문가시스템을 사용할 경우 두 가지의 연계성을 고려할 수 있다. 첫째가 다른 시스템에서 K-CLIPS를 하나의 서브루틴(subroutine)으로 사용하는 방법이고 둘째는 K-CLIPS의 규칙에서 타시스템의 함수들을 외부함수로 사용하는 방법이다. 현재 K-CLIPS는 두가지 방법이 다 사용될 수 있도록 설계되었다.

타 시스템에 K-CLIPS를 포함시킬 경우 이미 존재하는 시스템과 K-CLIPS와의 보다 쉬운 연계성을 가질 수 있다. 이 방법은 전문가시스템이 큰 사업의 작은 부분을 차지할 경우나 다른 함수와 데이터를 상호 교환이 필요할 경우 매우 유용하다.

또한 K-CLIPS의 규칙을 수행하는 동안에 규칙의 LHS와 RHS에서 외부함수를 불러 사용할 수 있다. 규칙의 LHS에서는 규칙의 조건을 테스트하기 위한 조건연산식(predicate)로서 사용되며, RHS에서는 에러가 발생했을 때 사용자에게 경고를 하는

함수와 같이 여러가지 추론을 위해 사용할 수 있다. 그리고 사용자가 정의한 외부함수는 상위레벨(top level)에서 불러어질 수 있다. K-CLIPS의 변수와 문자 상수(literal values)는 외부함수와 K-CLIPS사이에 상호 교환이 가능하다.

## 2.6 고도의 프로그램 기법

주어진 문제의 해결을 위해서는 그 문제의 특성을 헤아려 봄이 바람직하다. 이러한 문제들의 분석결과로서 가장 기본적인 것이 문제의 해결전략이라고 볼 수 있다.

해결전략의 예를 들어보면 그 부류는 크게 두가지로 나누어 진다. 몇가지 초기 상태로 부터 어떠한 제약조건들을 만족시키는 해결책을 찾는 문제에는 전진추론(forward chaining)이 적합하고, 반대의 경우 즉, 초기의 상태가 많고 결과 또는 목표가 적은 문제에는 후진추론(backward chaining)이 적합하다. 후자는 흔히 특정한 목표를 가정하여 그 목표를 달성하기 위한 부차적인 목표(sub-goal)를 만족시키는 방식으로 접근해 가는 데 최종적으로 특정한 사실에 대한 진위에 따라 가정의 진위가 판단된다.

위와 같은 문제해결의 방법에 못지않게 중요한 것이 전문가시스템의 하나인 설명기능(explanation facility)을 들 수 있다. 이 설명기능은 후진추론의 경우에는 실현하기 쉬운데 이때는 보통 부목표(sub-goal)을 제시함으로써 이루어 진다. K-CLIPS에서는 설명기능을 추가하기 위하여 규칙의 우선권(priority)를 최대한 활용할 수 있게 하였다.

규칙의 우선권은 salience를 이용하여 salience가 낮은 것은 보다 높은 규칙이 적용된 이후에 적용될 것이므로 순차적인 작업에 적합하고 높은 것은 우선적으로 적용되므로 설계조건등과 같이 제약사항을 적용하는 데 유리하다.

### 3. 패턴매칭 (Pattern Matching)

#### 3.1 LHS 조건문

이 절에서는 규칙의 LHS에서 사용되는 패턴의 문법(syntax)에 대해 언급한다. 이러한 패턴은 규칙이 수행되기 전에 만족해야 할 조건을 표현한다.

##### 3.1.1 패턴 매치

###### a) 문자패턴 (literal pattern)

가장 기본적인 패턴은 매치하고자 하는 사실을 정확하게 정의하는 것이다. 이것을 문자패턴(literal pattern) 이라고 부른다. 문자패턴의 모든 필드(field)는 fact-list상에 기입된 패턴의 모든 필드와 서로 매치되는가를 판단한다. 이러한 문자패턴내에서는 변수가 없다.

예)

규칙의 LHS 패턴	fact-list내의 사실	매치?
(group left town)	(group left town)	Y
(group left town)	(group in town)	N
(group in Houston)	(group in Houston)	Y
(group in Houston)	(group in HOUSTON)	N
(group in my home state)	(group in "my home state")	N
(data "a red flag")	(data "a red flag")	Y
(data 1)	(data 1.0)	Y
(data 100.321)	(data 100.3)	N

문자열내에 있는 공백(space)과 제어문자(control character)에 주의하고 소숫점 이하 3째자리 이상의 수치값의 매치는 많은 에러를 유발한다.

b) 와일드카드(wildcards)

K-CLIPS에서는 패턴상의 필드를 표현하는 방법으로 두가지의 와일드카드를 사용할 수 있다. 즉 물음표(?)와 달러-물음표(\$?) 와일드카드가 있다. 물음표 와일드카드(question wildcard)는 사실의 하나의 필드에 저장된 값(숫자,단어,문자열)과 매치된다. 그래서 이를 단일필드 와일드카드(single-field wildcard)라고 한다. 달러-물음표 와일드카드는 사실의 영 이상의 필드와 매치된다. 그래서 이를 다중필드 와일드카드라 일컫는다. 하나의 패턴에 단일필드 와일드카드와 다중필드 와일드카드(multifield wildcard)를 혼합하여 사용할 수 있다.

예)

규칙의 LHS내의 패턴	fact-list내의 사실	매치?
(data red ?)	(data red)	N
(data red ?)	(data red green)	Y
(data red ?)	(data res "green")	Y
(data red ?)	(data red green green)	N
(data red ?)	(data green red)	N
(data red \$?)	(data red)	Y
(data red \$?)	(data red green)	Y
(data red \$?)	(data red "green")	Y
(data red \$?)	(data red green green)	Y
(data red \$?)	(data green red)	N
(data ? ?)	(data red)	N
(data ? ?)	(data red green)	Y
(data ? ?)	(data red "green")	Y
(data ? ?)	(data red green green)	N
(data ? ?)	(data green red)	Y
(data ? \$?)	(data red)	Y
(data ? \$?)	(data red green)	Y
(data ? \$?)	(data red "green")	Y

(data ? \$?)	(data red green green)	Y
(data ? \$?)	(data green red)	Y
(data \$? ?)	(data red)	Y
(data \$? ?)	(data red green)	Y
(data \$? ?)	(data red "green")	Y
(data \$? ?)	(data red green green)	Y
(data \$? ?)	(data green red)	Y

다중필드 와일드카드와 문자필드(literal field)을 혼합하여 사용하면 보다 유용한 형태의 패턴을 만들 수 있다. 예를 들어 첫번째 필드이외에 YELLOW라는 단어를 가진 모든 사실을 찾고자 한다면 패턴은 다음과 같이 쓸 수 있다.

(data \$? YELLOW \$?)

매치될 수 있는 사실의 예는 다음과 같다.

(data YELLOW blue red green)  
 (data YELLOW red)  
 (data red YELLOW)  
 (data YELLOW)  
 (data YELLOW data YELLOW)

마지막 사실은 YELLOW가 두번 나타나기 때문에 두번 매치될 수 있다. 다중필드 와일드카드는 하나의 사실로 부터 매치될 수 있는 모든 가능한 조합을 만들기 때문에 단일필드 와일드카드로 원하는 매치를 이룰 수 없는 패턴의 경우에만 제한적으로 사용해야 한다. 다중필드 와일드카드의 매치시간은 단일필드 와일드카드에 비하여 많이 소모된다.

### c) 변수 (variable)

#### c.1) 제한조건이 없는 변수

와일드 카드 기호와 단어로 이루어진 것을 변수라고 하는데 이것은 사실의 필드

LHS내의 패턴	fact-list내의 사실	?x의 값	\$?y의 값
(data ?x \$?y)	(data red)	red	- nothing -
(data ?x \$?y)	(data red green)	red	green
(data ?x \$?y)	(data red "green")	red	"green"
(data ?x \$?y)	(data red green green )	red	green green
(data ?x \$?y)	(data green red)	green	red

LHS내의 패턴	fact-list내의 사실	\$?x의 값	?y의 값
(data \$?x ?y)	(data red)	- nothing -	red
(data \$?x ?y)	(data red green)	red	green
(data \$?x ?y)	(data red "green")	red	"green"
(data \$?x ?y)	(data red green green)	red green	green
(data \$?x ?y)	(data green red)	green	red

같은 변수명을 가진 변수는 대입된 값이 같아야 매치가 될 수 있다. 이런 사항은 단일필드 변수와 다중필드 변수 둘 다 적용된다. 그리고 또한 여러개의 패턴에서도 마찬가지로 적용된다.

단일패턴의 예)

LHS내의 패턴	fact-list내의 사실	매치?
(data ?x red ?x)	(data blue red blue)	Y
(data ?x red ?x)	(data blue red green)	N
(data ?x red ?x)	(data blue blue blue)	N
(data ?x red ?x)	(data blue red "blue")	N

다중필드의 예)

LHS내의 패턴들	fact-list내의 사실	fires?
(defrule .... (data red ?x) (data purple ?x) => ....	(data red green) (data purple green)	Y
(defrule .... (data red ?x) (data purple ?x) => ....	(data red green) (data purple blue)	N

```
(defrule ....
  (data red $?x)           (data red blue green)      Y
  (data purple $?x)        (data purple blue green)
  => ....
```

```
(defrule ....
  (data red $?x)           (data red blue green)      N
  (data purple $?x)        (data purple blue green)
  => ....
```

## c.2) 제한조건이 있는 변수

### c.2.1) 논리연산자(logical operators)

한 패턴내에 제한조건이 있는 변수는 세가지의 논리연산자를 사용할 수 있다. 즉 & (AND), | (OR) 와 ~ (NOT) 이 있다. 논리연산자는 패턴매치를 하는 동안 특별한 필드의 값을 제한하기 위하여 어떠한 형태나 갯수로 혼합하여 사용된다. 여러 개로 이루어진 제한조건의 평가는 왼쪽에서 오른쪽으로 고려해야 한다.

#### [Syntax]

```
<value1>&<value2>      : AND 연산자
<value1>|<value2>     : OR 연산자
~<value>               : NOT 연산자
```

AND 연산자는 다른 논리연산자와 변수를 함께 사용할 수 있다. 이 경우 논리연산자는 변수 뒤에 위치한다.

#### [More Syntax]

```
?x&<value1>|<value2>  : 변수와 함께 사용한 OR 연산자
?x&~<value>           : 변수와 함께 사용한 NOT 연산자
```

여기서 변수가 처음 언급될 경우는 필드는 단순히 논리연산자에 따라 제한되고

그 결과값이 변수에 저장된다. 그러나 변수가 이전에 언급되었을 경우에는 변수가 포함된 제한조건에 이 논리연산자에 따른 제한을 붙여야 한다. 즉, 필드는 이미 대입된 값과 동일한 값을 가져야 되고 또한 논리연산자에 의해 정의되는 제한조건을 만족되어야 매치가 된다.

예)

LHS내의 패턴	fact-list내의 사실	매치?
(data ~red)	(data red)	N
(data ~red)	(data green)	Y
(data red blue)	(data red)	Y
(data red blue)	(data blue)	Y
(data red blue)	(data green)	N
(data ~red&~blue)	(data red)	N
(data ~red&~blue)	(data blue)	N
(data ~red&~blue)	(data green)	Y
(data ~red&blue get)	(data red)	N
(data ~red&blue get)	(data blue)	Y
(data ~red&blue get)	(data get)	Y
(data ~red&blue get)	(data green)	N

변수를 가진 단일 패턴의 예)

LHS내의 패턴	fact-list내의 사실	?X의 값	매치?
(data ?x&~red)	(data red)	----	N
(data ?x&~red)	(data green)	green	Y
(data ?x&red blue)	(data red)	red	Y
(data ?x&red blue)	(data blue)	blue	Y
(data ?x&red blue)	(data green)	----	N

변수를 가진 이중패턴의 예)

LHS내의 패턴	fact-list내의 사실	?X의 값	매치?
(defrule ... (data A ?x)	(data A red)	red	N

```

(data B ?x&~red)      (data B red)
=> .....

(defrule ...
  (data A ?x)          (data A red)      red      N
  (data B ?x&~red)    (data B green)
=> ....

(defrule ...
  (data A ?x)          (data A blue)     blue     Y
  (data B ?x&~red)    (data B blue)
=> ....

(defrule ...
  (data A ?x)          (data A red)      red      Y
  (data B ?x&red|blue) (data B red)
=> ....

(defrule ...
  (data A ?x)          (data A red)      red      N
  (data B ?x&red|blue) (data B blue)
=> ....

(defrule ...
  (data A ?x)          (data A red)      red      N
  (data B ?x&red|blue) (data B green)
=> ....

```

### c.2.2) 조건함수(predicate functions)

논리연산자보다 더 복잡한 필드의 제한이 필요할 경우 조건함수를 사용할 수 있다. 이 조건함수는 필드값이 함수에서 정의된 제한조건을 만족하는가를 체크한다. 만약 만족한다면 함수는 TRUE값을 가지게 되고 다음 패턴을 계속 매치하게 되며 그렇지 않으면 FALSE값을 가지며 패턴매치가 중지된다. 조건함수는 콜론을 지닌 AND 연산자(&:)와 함께 사용된다. 필드의 값은 변수에 대입된 값을 만족해야 되고 조건함수도 만족해야 한다.

[Syntax]

?x&:(<function> <<argument>>)

다중 조건함수(multiple predicate function)는 하나의 필드를 보다 많이 제한하기 위해 사용될 수 있다. 이 함수는 왼쪽에서 오른쪽으로 평가되고 각각 &:에 의해 분리된다. K-CLIPS에서는 여러개의 조건함수를 가지고 있고 사용자가 이러한 조건함수를 만들 수 있다. 다음은 K-CLIPS가 지니고 있는 조건함수들이다.

조건함수	용도
(numberp <arg>)	<arg>의 값이 숫자인가?
(evenp <arg>)	<arg>의 값이 짝수인가?
(oddp <arg>)	<arg>의 값이 홀수인가?
(stringp <arg>)	<arg>의 값이 문자열인가?
(wordp <arg>)	<arg>의 값이 단어인가?

예)

LHS내의 패턴	fact-list내의 사실	매치?
(data ?x&:(numberp ?x))	(data 2)	Y
(data ?x&:(numberp ?x))	(data red)	N
(data ?x&:(numberp ?x)&:(oddp ?x))	(data 2)	N
(data ?x&:(stringp ?x))	(data 2)	N
(data ?x&:(stringp ?x))	(data red)	Y

c.2.3) 패턴확장연산자(pattern expansion operator)

이것은 변수와 함께 필드값을 제한하지 않고 단지 외부함수로만 필드값을 제한한다. 사용된 외부함수의 반환값(return value)은 패턴의 필드로 사용한다. 그러므로 패턴 확장연산자(=)는 하나의 패턴내에서 외부함수를 사용을 허용하게 한다. 그러나 외부함수에 의해 반환되는 값은 문자열, 단어, 숫자 중 하나 이어야 한다. 그리고 이런 값은

연산자가 사용된 위치에서 직접 대입되고 패턴을 매치할 경우 이 값이 매치된다.

[Syntax]

`=(<function> <<args...>>)`

예)

LHS내의 패턴	반환 값	fact-list	matches?
<code>(data =(my_fun))</code>	2	<code>(data 2)</code>	Y
<code>(data =(my_fun))</code>	2	<code>(data red)</code>	N
<code>(data =(my_fun))</code>	red	<code>(data 2)</code>	N
<code>(data =(my_fun))</code>	red	<code>(data red)</code>	Y

또한 패턴확장연산자는 변수와 다른 논리연산자와 함께 사용할 수 있다.

LHS내의 패턴	fact-list내의 사실	matches?
<code>(defrule ... (data ?y) (data ?x =&gt; ....</code>	<code>(data 4)</code>	Y

#### c.2.4) test 함수

test함수는 앞서 사용한 패턴의 변수를 제한하는 것으로 이 test 조건을 만족해야 다음 패턴을 계속 패턴매치할 수 있다. 조건함수처럼 사용자가 이미 변수에 대입된 값을 비교할 수 있고 변수의 수치적인 비교, 복잡한 논리비교 또는 등가비교가 가능하다. 또한 test함수는 외부함수를 사용하여 사용자의 요구에 따른 변수의 비교를 가능하게 해준다. K-CLIPS는 여러개의 논리함수,비교함수와 산술함수를 제공한다. K-CLIPS에서 정의된 함수의 모든 리스트는 주.2 에 나타났다.

## [Syntax]

(test (<defined-function> [<<arguments>>]))

test는 함수를 사용하는 특수한 형태이기 때문에 test함수를 사용하지 않는 패턴들은 각 패턴의 첫번째 필드에는 test라는 단어(word)를 사용할 수 없다. 또한 규칙의 LHS의 첫번째 패턴에서는 test 함수를 사용할 수 없다. test내에 존재하는 함수는 다시 test함수를 사용할 수 있고 이 경우는 내부부터 먼저 평가된다.

예) 다음 예제는 ?y와 ?X의 차이가 3보다 큰가를 체크한다.

LHS내의 패턴	fact-list내의 사실	fires?
(defrule ... (data ?x) (test (>= (- ?y ?x) 3)) => ....	(data 6) (data 9)	Y

다음 예제는 두점을 연결하는 직선의 기울기가 양수인가를 체크한다.

LHS내의 패턴	fact-list내의 사실	fires?
(defrule ... (point ?a ?x1 ?y1) (point ?b&~?a ?x2 ?y1) (test (< 0 (/ (- ?y2 ?y1) (- ?x2 ?x1)))) => ....	(point 1 4.00 7.00) (point 2 5.00 9.00)	Y

### 3.1.2 패턴의 조절

#### a) 패턴바인딩(pattern bindings)

retract와 같이 일부 RHS의 실행문(action)을 취할 때 하나의 패턴(즉 하나의 사

실)을 전부 다루어야 할 때가 있다. 이 경우 K-CLIPS에서는 변수에 한 패턴을 전부 대치시킨다.

[Syntax]

```
?<var-name><-(<<fields>>)
```

"<-" 는 변수에 한 패턴을 대치시키는 부호이다.

예)

```
(defrule dummy
  (data 1)
  ?fact<-(dummy pattern)
=>
  (retract ?fact))
```

b) 패턴의 제한

K-CLIPS에서 규칙의 LHS는 만족해야할 조건을 표현한 일련의 패턴들로 이루어져 있다. 그리고 이런 패턴을 모두 만족하면 agenda에 등록되어 진다. K-CLIPS에서는 모든 규칙의 LHS 패턴은 암시적인 and 로 구성되어 있다. 즉 이것은 규칙이 수행되기 전에 LHS의 모든 조건을 만족해야 한다는 것을 의미한다. 그러므로 명시적으로 and 를 정의할 필요가 없다. 그러나 이러한 기본적인 사항을 변경하거나 다른 논리적 조건의 정의가 가능하도록 설계되었다.

논리패턴블록(logical pattern blocks)은 패턴들을 논리합(OR)나 논리곱(AND)을 사용하여 여러 개로 구성될 수 있다. 이런 블록은 LHS상의 하나의 조건으로 간주되고 규칙이 수행되기 위해서는 모든 다른 조건들도 만족되어야 한다. 논리블록은 다른 논리블록들을 가질 수 있고 블록내의 패턴들도 필드제한에 사용된 것을 사용할 수 있

다. K-CLIPS에서는 이런 블록을 괄호로써 제한한다.

### b.1) 논리합(OR)블록

논리합 블록은 블록내의 여러 패턴중 하나라도 만족되면 이 블록의 조건을 만족하게 되고 또한 다른 LHS의 조건들도 다 만족된다면 규칙의 수행이 가능하게 된다. 논리합 블록내에 여러 개의 패턴을 사용함으로써 여러개의 유사한 규칙을 사용한 효과를 나타낼 수 있다.

```
[Syntax]
(defrule <name>
  [(< additional patterns >>)]
  (or (<<pattern 1>>)
      .
      .
      .
      (<<pattern n>>))
  [(< additional patterns >>)]

=>
(<<actions>>))
```

만약 논리합 블록내에서 만족되는 패턴의 수가 여러 개라면 그 조합의 수만큼 규칙은 수행이 될 수 있다.

예)

```
(defrule system-fault
  (error-status unknow)
  (or (temp high)
      (valve broken)
      (pump off))
  =>
  (fprintout t "The system has a fault." crlf))
```

위의 예는 다음의 세개로 분리된 규칙들과 동일한 것이다.

```
(defrule system-fault
  (error-status unknow)
  (pump off)
  =>
  (fprintout t "The system has a fault." crlf))
```

```
(defrule system-fault
  (error-status unknow)
  (valve broken)
  =>
  (fprintout t "The system has a fault." crlf))
```

```
(defrule system-fault
  (error-status unknow)
  (temp high)
  =>
  (fprintout t "The system has a fault." crlf))
```

## b.2) 논리곱(AND)블록

논리곱 블록은 and 와 or 조건을 조합하기 위해 제공되는 것으로 논리합 블록내에서 패턴들의 논리적인 조합을 허용한다. 논리곱 블록내의 모든 패턴들이 만족되어야 이 블록의 조건이 만족된다. 그리고 다른 LHS의 조건들도 만족되면 규칙은 수행되어진다.

### [Syntax]

```
(defrule <name>
  [(< additional patterns >>)]
  (or (and (<<pattern 1>>)
          .
          .
          (<<pattern n>>))
      (<<other patterns>>))
  [(< additional patterns >>)]

  =>
  (<<actions>>))
```

예)

```
(defrule system-flow
  (error-status confirmed)
  (or (and (temp high)
           (valve closed))
      (and (temp low)
           (value open))))
=>
(fprintout t "The system is having a flow problem." crlf))
```

### b.3) 패턴 부정 (pattern negation)

가끔 정보의 부족이 의미를 가질 경우가 있다.(예를 들어 규칙의 실행조건이 fact-list내에 사실이 존재하지 않을 경우) 이런 기능은 패턴 부정(not)을 이용할 수 있다.

[Syntax]

```
(defrule <name>
  (<<pattern 1>>)
  (not (<<pattern 2>>))
  [(<< additional patterns >>)]
=>
  (<<actions>>))
```

하나의 패턴에서는 한번만 부정할 수 있고 여러 개의 패턴을 부정할 때는 여러 개의 not 문장을 사용해야 한다. 논리곱 블록과 논리합 블록은 not 패턴 내부에 존재할 수 없다. 단지 논리곱 블록과 논리합 블록내에 not 패턴이 존재할 수 있다. 또한 논리곱 블록과 논리합 블록내에 not 패턴을 사용할 경우에도 많은 주의를 요한다. 왜냐하면 생각하지 못한 패턴을 만족시켜 주므로 많은 규칙을 수행가능하게 해 주기 때문이다. 마찬가지로 부정패턴(negated pattern)내에 사용한 바인딩된 변수도 주의를 필요로 한다. 그래서 부정패턴내에서 처음 대치되는 변수는 단지 그 패턴내에서만 사용된다. K-CLIPS에서는 부정패턴은 달리 처리되므로 부정패턴을 포함한 규칙을 만들었거나 로드한 후에는 항상 reset을 사용해야 한다.

예))

```
(defrule high-flow-rate
  (temp high)
  (valve open)
  (not (error-status confirmed))
  =>
  (fprintout t "Recommand closed of valve due to high temp" crlf))
```

```
(defrule check-value
  (check-status ?value)
  (not (valve-broken ?valve))
  =>
  (fprintout t "Device " ?valve "is OK" crlf))
```

```
(defrule double-pattern
  (data red)
  (not (data red ?x ?x))
  =>
  (fprintout t "No patterns with red green green!" crlf))
```

패턴 부정의 부정은 사용할 수 없다. 즉 (not (not (<<pattern>>)))은 허용할 수 없다.

### 3.1.3 규칙의 특성

여기서는 규칙의 특성을 정의하는 것에 관하여 언급하고자 한다. 규칙의 특성은 LHS에서 declare 을 사용함으로써 선언할 수 있다. 하나의 규칙에 여러개의 declare 을 선언할 수 있다. 그러나 모든 declare 문장은 LHS의 첫 패턴이전에 위치해야 한다.

[Syntax]

```
(declare (<something> [<<args>>])
```

현재로는 단지 salience 만이 가능하다.

a) 규칙의 우선권 선언

saliency 문장은 사용자가 규칙의 우선권의 결정을 가능하게 해 준다. agenda내에의 규칙들이 존재할 때 우선권이 높은 순서에 따라 수행된다. 동일한 우선권을 가질 경우는 나중에 선언된 규칙이 먼저 수행한다.

[Syntax]

```
(declare (saliency <num>))
```

여기서 <num>는 정수이다. 만약 표시하지 않으면 0 가 된다. saliency 값의 최대값은 10000 이며 최소값은 -10000 이다.

예)

```
(defrule test
  (declare (saliency 99))          ; saliency 선언
  (initial-fact)                  ; initial-fact 매치
  =>
  (fprintout t "Have a saliency value of 99." crlf))
```

## 3.2. RHS 실행문

이 절에서는 규칙의 RHS에서 활용할 수 있는 실행문에 관해 언급하고자 한다.

### 3.2.1 사실의 제어

다음 실행문은 RHS에서 가장 일반적인 것으로 사실을 만들고 제거하는 함수이다.

#### a) 새로운 사실을 만들

##### a.1) assert에 의한 사실 추가

assert 실행문은 fact-list에 사실을 첨가하는 함수로서 하나의 사실은 하나의 assert 실행문에 의해 만들어 진다. 규칙의 RHS에서는 여러개의 assert 실행문을 사용할 수 있다.

[Syntax]

```
(assert (<<pattern>>))
```

선언된 사실에는 변수, 함수의 호출, 문자상수가 포함될 수 있다.

이미 fact-list에 존재하는 사실을 새로 첨가할 수 없다.

예)

```
(assert (data red)) ; 단순한 사실 선언
```

```
(defrule close-value  
  (temp ?sensor high)  
  (valve ?v open)
```

```
(not (error-status confirmed))
=>
(assert (Set ?v close))           ; 변수를 사용한 사실 선언
(fprintout t "Close valve due to high temp" crlf))
```

예 : (외부함수를 호출)

```
(assert (data red =(+ ?x ?y)))    ; 호출 함수를 사용한 사실 선언

(defrule user-close-value
  (goal close valve)
  (input-from user)
  =>
  (fprintout t "Which valve should be closed?")
  (assert (close-valve =(read))))
```

#### a.2) str\_assert에 의한 사실 추가

str\_assert 실행문은 fact-list에 사실은 추가하는 것은 assert 실행문과 유사하다. 그러나 str\_assert 실행문은 하나의 문자열을 사용하고 이것이 사실로 첨가 되기 전에 인용부호는 제거된 후 여러개의 필드로 나누어 진다. assert 실행문과 같이 하나의 사실 추가는 하나의 str\_assert 실행문에 의해 이루어 진다.

[Syntax]

```
(str_assert <quoted-string>)
```

또한 fact-list에 존재하고 있는 사실은 다시 첨가할 수 없고 필드내에 문자열을 가지고 자 할때는 백슬래쉬(\)를 사용해야 한다. 이 실행문은 특히 나중에 언급할 readline 함수와 같이 사용하면 효율적이다.

예)

문장	추가되는 사실
(str_assert "status valve open")	(status valve open)

```
(not (error-status confirmed))
=>
(assert (Set ?v close))           ; 변수를 사용한 사실 선언
(fprintout t "Close valve due to high temp" crlf))
```

예 : (외부함수를 호출)

```
(assert (data red =(+ ?x ?y)))    ; 호출 함수를 사용한 사실 선언

(defrule user-close-value
  (goal close valve)
  (input-from user)
  =>
  (fprintout t "Which valve should be closed?")
  (assert (close-valve =(read))))
```

#### a.2) str\_assert에 의한 사실 추가

str\_assert 실행문은 fact-list에 사실은 추가하는 것은 assert 실행문과 유사하다. 그러나 str\_assert 실행문은 하나의 문자열을 사용하고 이것이 사실로 첨가 되기 전에 인용부호는 제거된 후 여러개의 필드로 나누어 진다. assert 실행문과 같이 하나의 사실 추가는 하나의 str\_assert 실행문에 의해 이루어 진다.

[Syntax]

```
(str_assert <quoted-string>)
```

또한 fact-list에 존재하고 있는 사실은 다시 첨가할 수 없고 필드내에 문자열을 가지고 자 할때는 백슬래쉬(\)를 사용해야 한다. 이 실행문은 특히 나중에 언급할 readline 함수와 같이 사용하면 효율적이다.

예)

문장	추가되는 사실
(str_assert "status valve open")	(status valve open)

```
(str_assert "light\red\")
```

```
(light "red")
```

## b) fact-list에서 사실을 제거

retract 실행문은 fact-list에서 사실을 제거하는 함수이다. 하나의 retract 실행문으로 여러 개의 사실들을 제거할 수 있다. 또한 이러한 사실을 제거하면 그 사실에 종속되는 모든 규칙들을 자동적으로 agenda에서 제거시킨다.

### [Syntax]

```
(retract ?<fact-var> [ ?<<fact-vars>> ] | [<<fact-num>> ])
```

제거하고자 하는 사실은 3.1.2에서 언급한 패턴바인딩 변수(?<fact-var>)나 fact-list의 순서를 표시하는 숫자(time tag)로 나타낸다. 프로그램이 수행중일 경우에는 숫자를 알기 어렵기 때문에 규칙의 LHS에서 대치된 변수를 사용하여 사실을 제거시켜야 한다.

예)

```
(defrule change-valve-status
  ?f1<-(valve ?v open)
  ?f2<-(Set ?v close)
  =>
```

```
(retract ?f1 ?f2)
(assert (valve ?v close)))
```

## c) 바인딩의 사용

규칙의 RHS에서 새로운 변수를 만들거나 변수의 값을 변경시키는 것은 매우 중요한데 K-CLIPS에서는 이런 기능을 bind 함수가 수행하여 준다.

[Syntax]

(bind ?<var-name> <value> or <variable> or (<defined-function>))

여기서 <var-name>은 변수명이며 이것에는 문자상수, 다른 변수 또는 호출 함수로부터 반환되는 값이 대입될 수 있다.

예)

```
(bind ?input (read))
(bind ?value (+ ?x ?y))
(bind ?value (+ (* ?y ?z) (* ?x ?z)))
(bind ?new ?old)
(bind ?number 14)
```

d) 다중필드함수(multifield functions)

다음에 언급하는 것은 여러개의 필드로 이루어진 변수들을 처리하는 함수이다.

d.1) 요소수의 결정

length 함수는 다중필드 변수(multifield variable)에 치환된 값의 요소(element) 수를 구한다.

[Syntax]

(length <variable>)

여기서 <variable>은 다중필드 변수(\$?)이어야 한다. 만약 변수가 다중필드변수가 아니면 -1 값을 가진다.

예)

fact-list안에 다음과 같은 사실이 있다고 하면

```
(data a b c d e f g)
```

```
(defrule test-length  
  (data $?a)  
  =>  
  (bind ?num (length $?a))  
  (fprintout t crlf "Length of data is " ?num))
```

규칙의 수행 결과

```
Length of data is 7
```

#### d.2) 요소의 내용을 출력

nth 함수는 다중필드함수로 부터 특정한 요소를 구할 수 있다. 이 함수는 LISP의 nth 함수와 유사하다.

[Syntax]

```
(nth <number> <variable>)
```

여기서 <number>는 1 부터 변수가 가지는 최대의 필드수까지 사용할 수 있으며 <variable>는 다중필드 변수(\$?)이어야 한다. 만약 <number>가 변수의 요소수보다 큰 수일 경우는 NIL이라는 심볼이 반환된다.

예)

fact-list에 다음과 같은 사실이 존재한다면

```
(data a b c d e f g)
```

```
(defrule test-length  
  (data $?a)  
  =>  
  (bind ?c (nth 3 $?a))  
  (fprintout t crlf "Third element in data is " ?c))
```

규칙의 수행 결과

Third element in data is c

### d.3) 요소의 발견

member 함수는 사용자가 원하는 요소가 다중필드변수에 있는 것인가를 결정해 준다.

[Syntax]

```
(member <atom> $?<var>)
```

여기서 <atom>은 문자상수 또는 변수이고 <var>는 다중필드변수이어야 한다. 만약 <atom>이 <var>의 요소이면 member 함수는 그 요소의 위치를 표시해 준다. 그렇지 않으면 0 가 반환된다.

예)

fact-list에 다음과 같은 사실이 있다고 한다면

```
(data red "green" blue brown)
```

```
(defrule test-member
  (data $?a)
  =>
  (bind ?val (member blue $?a))
  (fprintout t "Blue is item " ?val "in the list" crlf))
```

규칙의 수행 결과

Blue is item 3 in the list

### d.4) 다중필드 변수의 비교

이 함수는 하나의 다중필드변수가 다른 변수의 부분집합인지를 체크한다. 즉, 처

음 변수의 모든 요소가 두번째 변수의 요소로 이루어져 있는가를 체크한다.

[Syntax]

```
(subset $?<list1> $?<list2>)
```

이 함수를 사용하기 전에 두 변수는 이미 값이 치환되어 있어야 한다. <list1>이 <list2>의 부분집합이면 이 함수는 TRUE가 반환되고 그렇지 않으면 FALSE가 된다. 리스트내의 요소 위치는 고려할 필요가 없다. 만약 \$?<list1>이 어떤 요소도 가지지 않는다면 subset은 항상 TRUE값을 갖는다.

예)

변수들이 좌측값으로 치환된다면

```
 $?a <-- (hammer drill screwdriver wrench pilers saw)
  $?b <-- (hammer saw drill)
  $?c <-- (wrench crowbar)
```

결과로

```
(subset $?b $?a) --> TRUE
(subset $?c $?a) --> FALSE
```

### 3.2.2 K-CLIPS의 입력/출력 시스템

K-CLIPS는 효율적인 입력,출력을 위해 I/O routing 이라 불리는 시스템을 사용한다.

#### a) 입출력 지시명(logical names)

I/O routing의 기본적인 개념은 입출력 지시명을 사용하는 것이다. 입출력 지시명은 I/O 장치(device)에 대한 상세한 내부사항에 대한 이해없이도 I/O 장치를 지시해

줄 수 있다. K-CLIPS에서는 많은 함수들이 이런 입출력 지시명을 사용한다. 입출력 지시명은 문자상수,숫자 또는 문자열로 이루어 질 수 있다. K-CLIPS에서 미리 정의된 입출력 지시명은 아래에 나타내었다. 이런 입출력 지시명은 K-CLIPS 코드 전반에 걸쳐 사용된다.

이름	내용
wkclips	K-CLIPS 프롬프트(prompt)는 이 입출력 지시명에 보내어 진다.
wdialog	모든 정보는 입출력 지시명에 보낸다.
wdisplay	K-CLIPS의 정보(규칙,사실)를 보기 위한 요구사항은 이 입출력 지시명에 전달된다.
werror	모든 에러 내용은 이 입출력 지시명에 전달 된다.
wtrace	모든 조사된 정보는 이 입출력 지시명에 전달된다.
wagenda	agenda를 나타낼 때 이 입출력 지시명을 이용한다.
stdin	모든 기본적인 입력은 이 logical name을 사용한다.
stdout	printout 함수에서 생긴 모든 출력은 이 입출력 지시명에 보내어 진다. format과 fprintf 함수에서 출력을 stdout에 보내기 위해서는 t 라는 입출력 지시명을 사용해야 한다.

#### b) 일반적인 입출력 함수

K-CLIPS는 미리 정의된 여러 개의 함수에 의해서 일반적으로 사용되는 입출력 능력을 수행하도록 제공하고 있다.

### b.1) open

open 함수는 규칙의 RHS에서 사용되어 파일을 개방(open)하는 함수이며 이는 입출력 지시명을 지니고 있다. 이 함수는 세 개의 항(argument)으로 구성되어 있다.

(1) 열고자 하는 파일의 이름 (2) K-CLIPS의 타 입출력 함수가 이 파일을 액세스하기 위한 입출력 지시명 (3) 모드 선택자 (mode specifier). 모드 선택자는 다음중 하나를 사용해야 한다.

모드	용도
r	파일로부터 입력하기 위해 사용
w	파일에 기록하기 위해 사용
r+	파일로부터 입력하거나 파일에 출력하기 위해 사용
a	파일의 끝에 덧붙여서 기록하기 위해 사용

만약 모든 선택자가 선언되지 않으면 기본적으로 파일로부터 입력만이 가능하다. 이러한 모드선택자는 운영체제(operation system)에는 의미를 지니지 않는다.

### [Syntax]

```
(open "<file-name>" <logical-name> ["<mode>"])
```

<file-name>은 문자열로 구성되고 디렉토리 지시자(directory specifier)를 포함할 수 있다. 입출력 지시명은 이미 정의되지 않은 것을 사용해야 한다.

예)

```
(defrule file-open
  (initial-fact)
  =>
```

```
(open "myfile.clp" myfiletag "r+"))
```

```
(open "MS-DOS\directory\file.clp" file)
```

```
(defrule standard-print  
  (light off)  
  (wire broken)  
  =>  
  (fprintout werror "Light off because wire is broken!"))
```

## b.2) close

close함수는 open함수에 의해 열려진 파일 스트림(file stream)을 폐쇄시킨다. 파일은 logical name에 의해 원하는 스트림(stream)와 연결되어 있다.

### [Syntax]

```
(close [<logical-name>])
```

close가 인수없이 사용되면 현재까지 열려있는 모든 파일을 폐쇄시킨다.

예)

```
(defrule file-open  
  (initial-fact)  
  =>  
  (open "test.clp" data file))
```

```
(defrule file-close  
  (close-file data-file)  
  =>  
  (close data-file))
```

열린 파일은 반드시 폐쇄시켜야 한다. 그렇지 않으면 파일의 내용을 보호할 수 없다.

## b.3) read

read 함수는 선택된 입출력 지시명으로 부터 하나의 필드로 된 입력정보를 받아들인다. 다중필드로 된 단어는 인용부호와 함께 사용하면 읽어들일 수 있다.

[Syntax]

```
(read [<logical-name>])
```

여기서 <logical-name>은 선택적인 파라미터이다. 이것이 정의되면 정의된 입출력 지시명과 연결된 화일에서 입력을 받아들인다. 만약 파라미터가 정의되지 않거나 입출력 지시명이 t 일경우 stdin에서 입력을 받아들인다. 입력에서 받아들이는 개행문자, 공란 또는 탭은 경계를 나타낸다. 입력을 받아들이는 동안에 EOF 문자가 나타나면 read는 EOF 라는 문자가 되돌아 온다. 또한 에러가 발생하면 문자열 "\*\*\*\* READ ERROR \*\*\*\*"가 귀환된다.

예)

```
(defrule fileread
  (initial-fact)
  =>
  (open "data.clp" mydata)
  (bind ?input (read mydata))
  (assert (data ?input)))
```

b.4) readline

readline 함수는 read 함수와 유사하다. 그러나 입력을 단순히 하나의 필드가 아니라 전체 문자열을 사용한다. 일반적으로 read 는 공란, 탭, 개행문자, EOF에 의해 정지되지만 readline은 단지 EOF 문자에 의해 중단된다. 입력에 포함되는 탭이나 공란은 readline에 의해서는 단순히 문자열의 일부로 간주된다. readline의 반환되는 값은 문자열이다.

[Syntax]

```
(readline [<logical-name>])
```

여기서 <logical-name>은 선택적인 파라미터이다. 이것이 정의되면 정의된 입출력 지시명과 연결된 파일에서 입력을 받아들인다. 만약 파라미터가 정의되지 않거나 입출력 지시명이 t 일 경우 stdin에서 입력을 받아들인다. 입력을 받아들이는 동안에 EOF 문자가 나타나면 read는 EOF 라는 문자가 되돌아 온다. 또한 에러가 발생하면 문자열 "\*\*\*\* READ ERROR \*\*\*\*"가 귀환된다.

예)

```
(defrule file=read-line
  (initial-fact)
  =>
  (open "data.txt" mydata)
  (bind ?input-string (readline mydata))
  (assert (data ?input-string)))
```

만약 파일 data.txt 의 첫번째 라인이 다음과 같다면

```
the pretty red dog
```

fact-list에 저장되는 사실은 다음과 같다.

```
(data "the pretty red dog")
```

b.5) printout

printout함수는 표준출력장치(대개 스크린)에 결과를 출력을 하는 역할을 한다. 이 함수는 대치된 변수를 평가하고 그 변수의 결과값을 출력 스트링(output string)형태로 표시한다. printout내에는 여러 개의 항목(item)이 사용될 수 있다.

[Syntax]

```
(member <atom> $?<var>)
```

여기서 <atom>은 문자상수 또는 변수이고 <var>는 다중필드변수이어야 한다. 만약 <atom>이 <var>의 요소이면 member 함수는 그 요소의 위치를 표시해 준다. 그렇지 않으면 0가 반환된다.

예)

fact-list에 다음과 같은 사실이 있다고 한다면

```
(data red "green" blue brown)

(defrule test-member
  (data $?a)
  =>
  (bind ?val (member blue $?a))
  (fprintout t "Blue is item " ?val "in the list" crlf))
```

규칙의 수행 결과

```
Blue is item 3 in the list
```

#### d.4) 다중필드 변수의 비교

이 함수는 하나의 다중필드변수가 다른 변수의 부분집합인지를 체크한다. 즉, 처음 변수의 모든 요소가 두번째 변수의 요소로 이루어져 있는가를 체크한다.

[Syntax]

```
(subset $?<list1> $?<list2>)
```

이 함수를 사용하기 전에 두 변수는 이미 값이 치환되어 있어야 한다. <list1>이 <list2>의 부분집합이면 이 함수는 TRUE가 반환되고 그렇지 않으면 FALSE가 된다. 리스트내의 요소 위치는 고려할 필요가 없다. 만약 \$?<list1>이 어떤 요소도 가지지

```
(printout <item> ... <item> [crlf])
```

여기서 <item>에는 변수, 문자열, 숫자, 문자 또는 호출 함수가 들어 갈 수 있다. crlf는 새로운 라인에서 프린트하라는 개행문자(newline character) 이다. 그리고 crlf는 항목으로 이루어진 리스트의 어떤 위치에서도 사용할 수 있다.

예)

```
(defrule change-valve-status
  ?f1<-(valve ?v open)
  ?f2<-(Set ?v close)
  =>
  (retract ?f1 ?f2)
  (assert (valve ?v close))
  (printout "The valve " ?v "has been closed" crlf))
```

printout함수는 fprintf함수나 format함수로 대체될 수 있다.

#### b.6) fprintf

fprintf함수는 입출력 지시명과 연결된 파일에 출력을 하는 함수로 사용하기 전에 이미 파일은 개방되어 있어야 한다. stdout에 출력하고자 할 때는 입출력 지시명을 t로 사용해야 한다. fprintf함수는 대체된 변수를 평가하여 그 변수의 값을 문자열로 프린트한다. 또한 fprintf도 printout 함수와 마찬가지로 여러 개의 항목을 사용할 수 있다.

[Syntax]

```
(fprintf <logical-name> <item> ... <item> [crlf])
```

여기서 <item>은 문자, 변수, 호출함수를 사용할 수 있다. printout와 마찬가지로 개행문자인 crlf을 항목리스트의 어느 위치에도 사용할 수 있다.

예)

```
(defrule change-valve-status
  ?f1<-(valve ?v open)
  ?f2<-(Set ?v close)
  =>
  (open "out.clp" data "w")
  (retract ?f1 ?f2)
  (assert (valve ?v close))
  (fprintout data "the valve " ?v "has been closed" crlf)
  (fprintout data "Example with \"quoted\" information" crlf)
  (fprintout t "Data written." crlf))
```

#### b.7) format

format함수는 입출력 지시자와 연결된 파일에 원하는 형식의 출력을 시키는 함수이다. 특수한 출력의 형태를 원할 경우 fprintout함수대신에 사용될 수 있다. 사용하기에 복잡하기는 하지만 출력의 형태를 조절하기가 매우 용이하다. format함수는 C 언어의 printf함수와 유사하다.

#### [Syntax]

```
(format <logical-name> "<control-string>" <<parameter-list>>)
```

여기서 <logical-name>은 입출력 지시명이며 만약 t이면 stdout에서 출력된다. <control-string>은 결과를 어떤 형식으로 표현하는가를 나타내며 인용부호 사이의 문자열로 이루어져 있다. <<parameter>>은 출력하고자 하는 변수와 문자를 나타낸다. 또한 <<parameter>>에는 호출함수가 올 수 있으며 출력은 함수의 반환되는 값이다. 현재의 K-CLIPS에서는 <<parameter>>에 다중필드 변수가 올 수 없다.

제어문자열(control string)은 문장과 변환사양(format flag)으로 이루어져 있으며

문장은 선언한 문장 그대로 출력하며 변환사양은 <<parameter>>에 있는 변수의 출력형태를 나타내도록 한다. 첫번째 변환사양은 파라미터 리스트의 첫번째 값을 나타내고 두번째 변환사양은 두번째 값을 나타낸다. 변환사양은 퍼센트 부호(%)와 함께 사용되며 대표적인 형식은 다음과 같다.

%-M.Nx

여기서 x 는 아래의 변환문자중 하나이며 마이너스 부호(-) 는 선택적으로 사용되는 것으로 변환 인수를 왼쪽부터 채워 출력하도록 한다. 그리고 M 과 N 도 선택적인 파라미터로서 필드폭을 결정하고 소숫점이하의 자리수를 나타낸다. 만약 M 이 사용되면 최소한 M 자의 폭으로 취한다. M 자보다 큰값을 요구된다면 필드폭은 필요한 만큼 자동적으로 증가된다. 만약 0과 함께 M이 사용되면 (즉 %07d) 공백에 0을 채워 넣는다.만약 N이 정의되지 않으면 기본적으로 소숫점 이하의 자리수를 6으로 한다.

변환문자

변환의 내용

d	파라미터를 정수형으로 나타낸다.(N은 무의미하다.)
f	파라미터를 실수형으로 표현한다.
e	파라미터를 지수형식으로 표시한다.
g	위의 e나 f 중 짧은 표현을 선택해서 표현한다.
o	파라미터를 부호없는 8 진수로 표현한다. (N은 의미가 없다.)
x	파라미터를 부호없는 16진수로 표현한다. (N은 의미가 없다.)
s	파라미터를 문자열로 출력한다. 인용부호는 제거된다. (N 은 의미가 없으며 채워넣는 문자는 사용할 수 없다.)
n	출력을 새로운 줄에서 시작한다.



format함수는 C 의 sprintf 함수를 바탕으로 만들었기 때문에 sprintf함수가 지지 되지 못하는 시스템일 경우는 format함수를 사용할 수 없다.

### 3.2.3 그 밖의 실행문

이 절에서는 K-CLIPS 규칙의 RHS에 사용되는 그밖의 실행문을 언급하고자 한다.

#### a) system

system은 운영체제(operation system)를 호출하는 함수이다.

#### [Syntax]

(system <<args>>)

#### 예)

```
(defrule print-directory
  (print-directory ?directory)
  =>
  (system "dir " ?directory))
```

#### b) K-CLIPS 의 중지(stopping)

halt실행문은 더 이상의 규칙의 수행이 되지않도록 RHS에서 사용하는 것으로 인자를 필요치 않는다. halt가 호출되면 상위레벨로 되돌아 간다. 만약 agenda내에 수행할 수 있는 규칙이 남아 있으면 이를 다시 수행하기 위해서는 run이라는 명령을 사용해야 한다.

[Syntax]

(halt)

예)

```
(defrule halt-error
  (data ?x)
  =>
  (if (< (abs ?x) 0.00001)
    then
      (fprintout t "Error, number to small" crlf)
      (halt)
    else
      (assert (new_data =(/ 15 ?x))))))
```

c) 심볼의 생성

c.1) Gensym

gensym실행문은 단일 필드로 저장되는 순차적이고 특별한 단어가 반환된다. gensym의 첫번째 호출로 반환되는 값은 gen1 이며 계속적인 호출로 gen2, gen3... 등으로 증가된다. clear함수에 의해서 gensym의 값을 수정할 수 없다. 만약 사용자가 gensym과 같은 기능을 설계하고자할 때 사용자가 이러한 사실을 새로 만들 필요가 없다.

genX

예)

```
(assert (new-id =(gensym) flag1 7))
```

gensym의 처음 호출되었다면 사실은 다음과 같이 저장된다.

(new-id gen1 flag1 7)

c.2) setgen

setgen함수는 gensym의 시작되는 숫자를 지정하는 함수이다.

[Syntax]

(setgen <num>)

여기서 <num> 은 양의 정수이다. gensym을 호출하면 <num>에 선언한 숫자부터 계속적으로 증가된 값이 반환된다.

예)

(setgen 32)

그 다음 gensym을 호출하면 gen32가 반환되고 다시 호출하면 gen33이 된다.

d) call

K-CLIPS 는 규칙의 RHS에서 직접 외부함수를 호출할 수 있다.

[Syntax]

(call (<function-name> [<<args...>>]))

이런 형태의 호출된 함수는 단지 효과적인 경우에만 사용해야 한다. 반환되는 값은 알 수가 없다.

예)

```
(defrule display-valve-system
  (valve ?v open)
  (valve ?v2
   =>
   (call (display-valves ?v1 ?v2)))
```

e) if..then..else

가끔 두개이상의 규칙을 사용하는 대신 규칙의 RHS에서 파라미터를 테스트하여 명령문을 취하는 것이 편리할 때가 있다. K-CLIPS에서는 이런 작용을 if..then..else 구조가 처리해 준다.

[Syntax]

```
(if <predicate-function> [<<args...>>])
  then
    (<<action 1>>)
    .
    (<<action n>>)
  [else
    (<<action 1>>)
    .
    (<<action n>>)]
```

then과 else내에 존재할 수 있는 RHS의 명령문은 여러 개 올 수 있다. else 부분은 선택적으로 취할 수 있다. 모든 조건함수는 이 구조의 조건문에 사용될 수 있으며, 즉 테스트에 이용된다. 그러나 비교하는 변수는 이전에 대치된 값을 지니고 있어야 한다.

예)

```
(defrule closed-valves
  (temp high)
  (valve ?v closed)
```

```

=>
(if (= ?v 6)
  then
    (fprintout t "The special valve " ?v "is closed!" crlf)
    (assert (perform special operation)))
  else
    (fprintout t "Valve " ?v "is normally closed" crlf)))

```

위의 규칙은 두개의 규칙을 하나로 줄이는 효과를 가진다.

```

(defrule closed-values-number-6
  (temp high)
  (valve 6 closed)
  =>
  (fprintout t "The special valve 6 is closed!" crlf)
  (assert (perform special operation)))

(defrule closed-values-other-than-6
  (temp high)
  (valve ?v)
  =>
  (fprintout t "Valve " ?v "is normally closed" crlf))

```

#### f) while

while문은 규칙의 RHS에서 조건문이 거짓일 때까지 반복하는 루프(loop)로서 if..then..else와 유사하게 사용된다.

[Syntax]

```

(while (<predicate-function> [(<<args...>>)])
  (<<action 1>>))
.
.
(<<action n>>))

```

모든 조건함수는 while의 조건문에 사용할 수 있고 while블록에는 if..then..else구조와 또다른 while문을 포함하여 여러 개의 RHS의 명령문을 포함할 수 있다. 루프의 실행에 앞서 테스트문장을 실행하여 만족해야 한다.

예)

```
(defrule open-valves
  (valve-open-through ?v)
  =>
  (while (> ?v 0)
    (fprintout t "Valve " ?v "is open" crlf)
    (bind ?v (- ?v 1))))
```

g) 문자열의 결합(string concatenation)

str\_cat은 단어,숫자와 문자열로 이루어진 두 개이상의 인수를 하나의 문자열로 만드는 함수이다.

[Syntax]

```
(str_cat <item1> <<items>>)
```

각 <item1>은 단어,숫자,문자열,변수,함수중 하나이다.

예)

```
(bind ?new1 (str_cat "foo" bar))
(bind ?new2 (str_cat foo bar))
(bind ?new3 (str_cat 12 34))
```

결과

```
?new1 = "foobar"
?new2 = "foobar"
?new3 = "1234"
```

### 3.3 계층적인 지식을 표현하는 기능

K-CLIPS는 계층적인 지식을 보다 효율적으로 표현하고자 프레임 형태와 유사한 기능을 가지고 있다. 이것은 객체(object)와 객체사이의 관계를 정의하고 객체가 가지고 있는 속성값을 상속할 수 있는 기능을 가지고 있어 상위 클래스(super class)에서 정의한 속성(attribute)은 하위 클래스에서 다시 정의할 필요가 없다.

#### 3.3.1 클래스의 정의

여기서는 클래스를 정의하는 것을 언급하고자 한다. 이 정의에 의해 하나의 객체와 하위 객체사이의 관계를 연결시켜 준다.

[Syntax]

```
(define_class <class> ([<<sub class>>]))
```

여기서 <class>는 정의하는 클래스이고 <subclass>라는 하위 클래스를 갖는다. <class>에 속하는 모든 슬롯은 하위 클래스에 상속된다.

<class>는 단어,숫자,문자열,호출함수를 취할 수 있다.

이 함수는 상위레벨에서 사용하거나 규칙의 RHS에서 사용할 수 있다.

예)

```
(define_class frame (beam column))
```

이 문장은 frame의 하위 클래스에는 beam과 column이 있다는 것을 정의한다.

### 3.3.2 클래스의 슬롯 관리

클래스로 정의된 사실은 frame-lists에 저장된다. 이 절에서는 클래스의 속성을 나타내는 슬롯을 만들거나 수정, 제거하는 작업에 관해 설명하고자 한다. 여기서 설명하는 함수는 상위레벨에서 사용하거나 규칙의 RHS에서 사용할 수 있다.

#### a) 슬롯의 생성

create\_slot 함수는 정의된 클래스의 슬롯과 슬롯값을 정의 한다.

#### [Syntax]

```
(create_slot <class> ^<slot 1> <value 1>
      .
      .
      .
      [^<slot n> <value n>])
```

여기서 <class>는 define\_class 함수에 의해 선언된 클래스이며 ^기호는 다음에 나오는 것은 클래스의 슬롯이라는 것을 알려 준다. 슬롯과 슬롯값은 항상 쌍으로 이루어진다. 그리고 이들은 단어, 숫자, 문자열, 호출함수를 취할 수 있다.

예)

```
(create_slot beam ^material concrete
              ^length 100 )
```

이 문장은 beam이라는 클래스내에 material과 length라는 슬롯을 가지고 그 각각의 값은 concrete와 100을 가진다.

## b) 슬롯의 제거

`remove_slot`은 `frame-lists`에 있는 클래스내의 슬롯을 제거하는 함수로서 아래와 같이 정의된다.

### [Syntax]

```
(remove_slot <class> ^<slot> [<value>])
```

여기서 `<class>`는 선언된 클래스이며 `<slot>`은 제거하고자 하는 슬롯명이다. 여기서 `<value>`을 선언하지 않으면 이 슬롯이 가지고 있는 모든 값을 제거한다. 그러나 특정한 값을 선언하면 선언된 값만 제거한다.

예)

```
(remove_slot beam ^material concrete)
```

## c) 슬롯의 추가

`add_slot`은 새로운 슬롯을 만들거나 이미 만들어진 슬롯의 값을 변경하고자 할 때 사용한다. `create_slot`에 의해 이미 만들어진 슬롯과 슬롯값을 선언하면 이전의 슬롯값을 제거되고 새로 정의한 슬롯값이 기록된다. 그러나 `add_slot`를 사용하면 이전의 슬롯값과 새로 정의한 슬롯값을 이 슬롯에 모두 저장시킨다. 즉 여러 개의 슬롯값을 지닐 수 있다.

### [Syntax]

```
(add_slot <class> ^<slot> <value>)
```

여기서 `<class>`는 선언된 클래스이며 `<slot>`은 이미 선언된 슬롯이거나 새로운

슬롯이다. 그리고 <value>는 슬롯값이다.

#### d) 슬롯값의 수정

modify함수에 의해 슬롯값을 변경시킬 수 있다.

#### [Syntax]

(modify <class> ^<slot> <value 1> => <value 2>)

여기서 <value 1>는 이미 정의된 클래스의 슬롯값이고 <value 2>는 변경하고자 하는 슬롯값이다.

### 3.3.3 클래스의 패턴매치

frame-lists에 저장된 계층적인 사실들은 규칙의 LHS에서 패턴매치에 사용되어야 한다. 여기서 사용할 수 있는 것으로는 두가지의 방법이 있다. 첫째로는 클래스의 패턴문장을 사용하는 방법이고 다른 방법은 test함수와 함께 클래스의 조건함수를 사용하는 방법이 있다.

#### a) 클래스의 패턴문장

##### a.1) SUBCLASS\_OF

SUBCLASS\_OF 함수는 클래스와 클래스사이의 계층관계를 표현한다. 여기서 SUBCLASS\_OF는 대문자를 써야 한다. 그렇지않고 소문자를 사용하면 다음에 언급할 클래스의 조건함수(predicate)가 된다.

[Syntax]

(SUBCLASS\_OF <class 1> <class 2>)

여기서 <class 1>과 <class 2>는 단어와 변수가 올 수 있고 <class 1>의 하위 클래스가 <class 2>인 것을 서술한다. 예를 들어 <class 2>가 변수이면 frame-list에 있는 모든 <class 1>의 하위 클래스는 모두 <class 2>의 값에 대치된다.

#### a.2) SLOT\_VALUE

SLOT\_VALUE함수는 클래스, 슬롯과 슬롯값을 매치시키는 함수로서 다음과 같이 선언된다.

[Syntax]

(SLOT\_VALUE <class> <slot-name> <value>)

여기서 <class>, <slot-name>는 미리 정의한 값이며 변수가 될 수 없다. 그러나 <value>에는 변수가 들어가 frame-lists에 들어있는 모든 값이 변수에 대치되어 진다.

#### b) 클래스의 조건함수(predicate)

여기서 제시하는 클래스의 슬롯 또는 슬롯값의 관계가 frame-lists에 존재하는가를 체크하여 참 또는 거짓으로 반환되는 것을 클래스의 조건함수라고 하는데 여기서는 3 개의 조건함수에 대해 언급하고자 한다. 이 함수를 규칙의 LHS에 사용할 경우는 test함수와 함께 사용하고 RHS에서는 일반적인 조건함수와 동일하게 사용되어 진다.

##### b.1) slot\_of

slot\_of함수는 클래스와 슬롯과의 관계가 frame-lists에 기록된 사실과 동일한 가  
를 체크한다. 동일하면 TRUE, 그렇지 않으면 FALSE 값이 반환된다.

[Syntax]

(slot\_of <class> <slot-name>)

여기서 <class> <slot-name>은 단어이거나 이미 값이 대치된 변수를 사용되어 진  
다.

### b.2) subclass\_of

subclass\_of함수는 패턴문장의 SUBCLASS\_OF와 유사하나 반환되는 값을 가지  
고 있다는 것이 차이점이다.

[Syntax]

(subclass\_of <class 1> <class 2>)

여기서 <class 1>과 <class 2>는 단어나 값이 대치된 변수가 사용될 수 있다. 그러  
나 여기서 새로운 변수를 생성할 수는 없다.

### b.3) slot\_value

frame-lists내에 클래스의 슬롯의 속성값이 존재하면 TRUE가 반환되고 슬롯값이  
존재하지 않으면 FALSE가 반환된다.

[Syntax]

(slot\_value <class> <slot-name>)

여기서 <class 1>과 <class 2>는 단어나 값이 대치된 변수가 사용될 수 있다. 그러나 여기서 새로운 변수를 생성할 수는 없다.

### 3.4 함수를 정의하는 기능

#### 3.4.1 함수의 정의

K-CLIPS는 사용자의 편의를 위해 필요한 함수를 정의할 수 있는 기능이 제공되어진다. 정의된 함수는 K-CLIPS에 기본적으로 정의된 함수와 동일한 것으로 사용된다. 이런 함수의 정의는 defun이라는 함수에 의해 수행된다.

[Syntax]

```
(defun <fun-name>
  (<parameter>)
  [body] )
```

여기서 <fun-name>은 선언한 함수명이며 <parameter>는 선언한 함수 <fun-name>에 사용되는 인수를 정의한다. 그리고 [body]에서는 정의하고자 하는

함수들이 존재할 수 있다.

예)

```
(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 "*** 2 + "
    ?b "*** 2 is "?length crlf))
```

### 3.4.2 그밖의 보조 함수

사용자가 정의한 함수로 조절하기 위한 함수들로 다음과 같다.

#### a) displayfunc

이 함수는 현재까지 정의한 함수의 리스트를 모두 나타낸다.

[Syntax]

(displayfunc)

#### b) remove\_defun

이전에 정의한 함수를 제거할 필요가 있을 때 다음과 같은 함수를 사용한다.

[Syntax]

(remove\_defun <fun-name>)

#### c) ppfunc

이 함수는 선언한 함수의 내용을 알고자 할때 사용한다.

[Syntax]

(ppfunc <fun-name>)

### 3.5 K-CLIPS의 계산 함수(computational functions)

#### 3.5.1 기본 함수(elementary functions)

이 절에서는 K-CLIPS에서 제공하는 기본적인 함수에 관하여 예제와 함께 언급하고자 한다.

심볼	함수	예제	의미
<b>논리함수</b>			
!	not(inverse)	logical	-
&&	and	logical	-
	or	logical	-
not	not(inverse)	logical	-
and	and	logical	-
or	or	logical	-
<b>비교함수</b>			
=	equal(numeric)	(test (= ?x ?y))	?x = ?y
eq	equal(any)	(test (eq ?x ?y))	?x eq ?y
!=	not equal(numeric)	(test (!= ?x ?y))	?x != ?y
neq	not equal(any)	(test (neq ?x ?y))	?x neq ?y
>=	greater than or equal	(test (>= ?x ?y))	?x >= ?y
>	greater than	(test (> ?x ?y))	?x > ?y
<=	less than equal	(test (<= ?x ?y))	?x <= ?y
<	less than	(test (< ?x ?y))	?x < ?y
<b>조건함수</b>			
(numberp <arg>)	(foo ?x&:(numberp ?x))		숫자인가?
(stringp <arg>)	(foo ?x&:(stringp ?x))		문자열인가?
(wordp <arg>)	(foo ?x&:(wordp ?x))		단어인가?
(evenp <arg>)	(foo ?x&:(evenp ?x))		짝수인가?
(oddp <arg>)	(foo ?x&:(oddp ?x))		홀수인가?

#### 3.5.2 산술함수(math functions)

K-CLIPS에서는 수치 계산을 위해 여러개의 함수를 제공하고 있다. 이것은 두가지로 나눌 수 있다. 첫째는 기본적인 산술함수이고 둘째는 확장된 산술함수이다.

a) 기본적인 산술함수

기본적인 산술함수는 덧셈, 뺄셈, 곱셈과 나눗셈의 기본적인 연산을 할 수 있다. 이런 함수의 인수(argument)는 단지 숫자만 허용된다. 만약 계산도중에 인수에 문자열이 오면 에러가 발생된다.

예)

divide	(bind ?var (/ ?x ?y))	; ?var = ?x / ?y
multiply	(bind ?var (* ?x ?y))	; ?var = ?x * ?y
add	(bind ?var (+ ?x ?y))	; ?var = ?x + ?y
subtract	(bind ?var (- ?x ?y))	; ?var = ?x - ?y

b) 확장된 산술함수

K-CLIPS에서는 기본적인 산술함수와 더불어 복잡한 수치계산을 위해 여러가지 함수를 제공하고 있다. 만약 구축하고자 하는 전문가시스템이 이런 확장된 수치계산을 필요하지 않는다면 보다 많은 기억용량을 가지기 위해 이런 함수들의 정의하지 않아도 된다. (clips.h에 EX\_MATH를 0으로 하고 math.c를 생략하고 다른 K-CLIPS의 소스코드를 컴파일 한다.)

[Syntax]

(<fun> <arg>)

예)

(defrule trigtest

```
(number ?x)
=>
(bind ?y (cos ?x)))
```

다음 함수들은 하나의 인수만 필요하다.

함수	반환되는 값
cos	cosine
sin	sine
tan	tangent
sec	secant
csc	cosecant
cot	cotangent
acos	arccosine
asin	arcsine
atan	arctangent
asec	arcsecant
acsc	arccosecant
acot	arccotangent
cosh	hyperbolic cosine
sinh	hyperbolic sine
tanh	hyperbolic tangent
sech	hyperbolic secant
csch	hyperbolic cosecant
coth	hyperbolic cotangent
acosh	hyperbolic arccosine
asinh	hyperbolic arcsine
atanh	hyperbolic arctangent
asech	hyperbolic arcsecant
acsch	hyperbolic arccosecant
acoth	hyperbolic arccotangent

위에서 언급한 함수는 인수가 라디안(radian)값을 가진다.

log	자연 로그
log10	상용 로그
exp	e ** <arg>
sqrt	제곱근
trunc	인수값을 정수형으로 치환
deg-rad	도(degree)을 라디안으로 환산
rad-deg	라디안을 도로 환산

abs

절대값

다음에 언급하는 함수는 여러 개의 인수를 취한다.

함수	인수의 수	반환되는 값
min	하나 이상	모든 인수중 최소값
max	하나 이상	모든 인수중 최대값
pi	없음	파이
mod	2	인수1을 인수2로 나눈 몫
**	2	인수1을 인수2배 만큼 곱한 값

이러한 수치함수는 C 라이브러리 math.h을 사용한다. 만약 사용하는 시스템이 이런 라이브러리를 제공하지 못한다면 사용자는 math.c화일을 수정해야 한다. 시스템에 따라 삼각함수는 clcos, clsin 등으로 사용되는 경우가 있으므로 사용자의 시스템에 알맞는 함수를 호출할 수 있게 수정해야 한다. 위에서 언급한 함수는 실수값(floating-point number)이 반환된다. 대부분의 컴파일러는 이런 함수들을 링크하기 위해 분리된 수치 라이브러리를 포함시켜야 한다. (예로 UNIX의 경우 -lm을 링크할 때 사용한다.)

### 3.6 K-CLIPS의 인터페이스 함수

K-CLIPS에서는 전문가시스템의 사용자의 이해의 폭을 증가시키기기 위해 시스템구축에 활용할 수 있는 윈도우 및 메뉴, 그래픽 함수로 제공한다.

#### 3.6.1 윈도우 함수

K-CLIPS는 필요한 정보를 편리하게 전달할 수 있는 윈도우를 사용할 수 있게 하는 함수를 제공하고 있다.

##### a) window\_open

이 함수는 처음 윈도우를 선언할 경우에 사용한다. 여기서는 열고자 하는 윈도우의 크기와 번호를 결정해야 한다.

[Syntax]

(window\_open <window-id> <left> <top> <right> <bottom>)

<window-id>는 열고자 하는 윈도우의 번호를 말하며 <left> <top> <right> <bottom>은 윈도우의 크기를 나타낸다.

b) window\_close

이 함수는 선택된 윈도우를 닫고자 할 경우 사용한다.

[Syntax]

(window\_close <window-id>)

<window-id>을 지정하지 않고 (window\_close) 만 사용한 경우는 열린 윈도우는 모두 닫는다.

c) window\_move

이 함수는 선언된 윈도우의 위치를 변경시킬 경우에 사용한다.

[Syntax]

(window\_move <window-id> <x> <y>)

<x>, <y>는 옮길 윈도우의 상부좌측의 좌표를 나타낸다.

#### d) cprintout

이 함수는 윈도우에 정보를 프린트하는 함수로서 printout과 유사하다.

#### [Syntax]

```
(cprintout <item> ... <item> [crlf])
```

#### e) window\_remove

이 함수는 메모리에서 윈도우가 가지는 정보를 제거하는 함수이다.

#### [Syntax]

```
(window_remove <window-id>)
```

### 3.6.2 메뉴함수

메뉴함수는 시스템 개발자의 의도에 맞는 정보를 얻고자 할 경우나 사용자가 편리하게 정보를 시스템에 제공하는 입력함수이다.

#### [Syntax]

```
(make_menu <menu-id> <x> <y>  
  [ < argument-1 > ]  
  [ < argument-2 > ]  
  .....  
  [ < argument-n > ] )
```

<menu-id>는 정의하는 메뉴의 번호를 말하며 <x>, <y>는 menu의 상부좌측의 위치를 표시한다. 그리고 원하는 항목을 문자열로 표시하면 하나의 메뉴가 생성된다. 만들어진 메뉴의 각 항목을 결정할 경우는 Down키와 Up키를 사용하여 항목을 결정한

후 리턴키를 사용하면 원하는 항목의 번호가 리턴된다.

### 3.6.3 그래픽함수

그림을 구성하기 위한 함수로서 자세한 사항은 Part 1의 3장을 참조하기 바란다.

#### a) 조절함수

```
(initgraph <color>)  
(closegraph)
```

#### b) 선언함수

```
(setcolor <color>)  
(setlinestyle <linestyle> <thickness>)  
(setfillstyle <pattern> <color>)  
(settextjustify <horiz> <vert>)  
(settextstyle <font> <direction> <charsize>)
```

#### c) 그래픽함수

```
(arc <x> <y> <stangle> <endangle> <radius>))  
(circle <x> <y> <radius>)  
(drawpoly <x1> <y1> ... <xn> <yn>)  
(ellipse <x> <y> <stangle> <endangle> <xradius> <yradius>)  
(line <x1> <y1> <x2> <y2>)  
(linerel <dx> <dy>)  
(lineto <x> <y>)  
(moveto <x> <y>)  
(moverel <x> <y>)  
(rectangle <left> <top> <right> <bottom>)  
(bar <left> <top> <right> <bottom>)  
(bar3d <left> <top> <right> <bottom> <depth>)  
(fillellipse <x> <y> <xradius> <yradius>)  
(fillpoly <x1> <y1> ... <xn> <yn>)  
(floodfill <x> <y> <border>)  
(pieslice <x> <y> <stangle> <endangle> <radius>))  
(sector <x> <y> <stangle> <endangle> <xradius> <yradius>)
```

#### d) 문자함수

```
(outtext "<textstring>")  
(outtextxy <x> <y> "<textstring>")
```

### 3.7 작업수행시 고려할 사항

규칙 기반 시스템의 수행에 영향을 주는 사항은 일반적인 프로그램에 영향을 주는 것과는 상당히 차이가 있다. 가장 중요한 사항은 규칙의 LHS에 올 수 있는 패턴의 순서와 사실의 비교이다. 3.4.1에서는 패턴의 순서에 관해 언급하고 사실의 비교는 3.4.2에서 언급하고자 한다.

#### 3.7.1 패턴의 순서

K-CLIPS는 rete 알고리즘을 바탕으로 한 규칙기반의 언어이다. Rete 알고리즘은 특히 패턴매치를 효과적으로 수행하기 위해 설계되어졌다. K-CLIPS는 적절하게 사용하므로써 소형컴퓨터에서도 원활하게 기능을 수행할 수 있다. 그러나 적절하게 사용하기 위해서는 어떻게 패턴매치 작업이 이루어 지는가를 이해해야 한다.

작업수행에 앞서 사용하는 모든 규칙을 시스템에 로드(load)시켜 규칙의 LHS 패턴의 네트워크(network)을 만든다. fact-list에 사실을 추가할때마다 사실들은 이런 패턴네트워크에 의해 체크가 되어진다. 사실의 패턴형태(필드의 수, 문자필드)가 네트워크내의 어떤 패턴과 매치가 된다면 그 패턴에 대해 규칙은 부분적으로 실행가능하게 (instantiate) 되고 LHS의 모든 패턴이 매치된다면 변수값이 치환되어 진다. 이런 작업은 위에서 아래로 행해진다. 즉, LHS의 첫번째 패턴이 먼저 고려되고 그다음 두번째 패턴이 고려되는 순서를 가진다. 모든 패턴에 대한 필드의 값이 변수들에 적용되어 제한조건을 만족한다면 규칙은 실행이 가능하게 되고 agenda에 기록되어 진다.

위의 언급한 사항으로 K-CLIPS가 어떻게 작업을 수행하는가를 알 수 있다. 여기서 중요하게 고려할 사항은 이러한 패턴매칭은 패턴 네트워크에 의해 이루어지고 있다는 점이다. 기본적인 패턴매칭은 패턴과 대칭할 변수와의 비교하는 시간이 많이 필요하다. 그러므로 단순하면서도 중요한 작업의 인자는 규칙의 LHS에 위치하는 패턴의 순서이다. 그러나 정확하고 빠르게 패턴의 순서를 결정하는 방법은 없다. 아래에서는 적절하게 패턴의 순서를 결정하는 방법 3가지를 언급하고자 한다.

1) 특수한(specific) 것을 먼저 쓰고 일반적인(general) 것을 나중에 둔다. 와일드카드나 바인딩될 변수가 많은 것은 하위 패턴에 둔다. 규칙을 수행하기 위해서는 하나의 패턴에 의해 조절된다면 그런 패턴을 상위에 놓는다. 이런 기법은 전문가시스템의 패턴 제어에 종종 사용된다. 이런 패턴들을 먼저 위치시키면 그 패턴이 나타날 때까지 규칙의 나머지 패턴을 고려하지 않아도 된다. 만약 이런 패턴이 문자패턴이라면 더욱 효과적이다. 만약 변수를 가진 다중 패턴에 의해 규칙의 수행이 좌우된다면 가장 중요한 변수가 있는 패턴을 먼저 위치시킨다.

2) fact-list에 나타나는 횟수가 적은 패턴을 위쪽으로 놓는다. fact-list의 특이한 형태의 패턴들을 빠르게 부분적으로 실행가능하게 함(instantiation)으로 변수를 비교하는 작업을 줄이는 역할을 한다.

3) 변화가 심한 패턴 (계속적으로 만들어지고 제거되는 패턴) 은 나중에 둔다. 특히 나머지 패턴들과 독립적인 경우는 나중에 둔다. 새로운 사실이 만들어지면 이것은 네트워크와 체크가 필요하고 그런 사실이 규칙의 부분적인 실행가능 (instantiation)을 발생시키면 변수의 치환이 발생된다. 변화가 심한 패턴을 아래에 둠으로서 패턴의 나머지가 이미 존재한다면 단지 변수의 치환만 필요하게 된다.

그 밖의 작업에 유의할 사항은 다중필드변수와 와일드카드의 사용이다. 비록 이런 것이 강력한 작업수행능력을 가지고 있지만 사용할 때는 신중하게 고려를 필요로 한다. 이런 변수는 0 이상의 필드를 대치시킬 수 있으므로 한 규칙의 많은 실행가능 (instantiation)을 유발시킨다. 특히 하나의 패턴에 여러개의 다중필드변수를 가지면 여러개의 실행가능(instantiation)을 발생시킨다.

마지막으로 fact-list는 데이터베이스(data base)와는 다르게 사용된다. 단순히 정보의 저장만을 위한 것이 아니라 추론에 필요한 정보를 저장하고 패턴매치를 한다. 그러므로 사용자는 패턴매치의 양을 축소시키고자 하는 노력이 필요하다. 또한 짧고 단순한 규칙은 길고 복잡한 규칙보다 작업하기가 편리하고 이해하거나 유지하기가 편리하다.

### 3.7.2 fact-list와 사실과의 비교

fact-list에 새로운 사실을 추가할 때마다 K-CLIPS는 fact-list에 새로운 사실이 이미 존재하는가를 조사한다. 새로운 사실이 존재한다면 fact-list에 추가되지 않는다. 똑같은 사실을 이중으로 실행 가능하게 하는 것은 작업의 수행이나 제어하는 데 어려움이 있기 때문에 사실이 이중으로 선언되는 것을 방지하는 것이 유리하다. 그러나 새로운 사실마다 비교하는 일은 시간을 소비한다. 이런 문제를 벗어나기 위해 시스템 개발자가 이중으로 선언되는 사실이 절대 없는 것을 보장한다면 K-CLIPS는 선언되는 사실을 비교하는 일을 피할 수 있는 기능을 가지고 있다. 다음 함수가 이런 기능을 수행한다.

[Syntax]

(check-facts <arg>)

여기서 <arg>는 on 이나 off중 하나이며 check-facts가 on이면 새로 선언한 사실은 fact-list와 비교를 수행하고 그렇지않으면 수행하지 않는다. 물론 디폴트(default)는 on 이다. 이런 함수는 규칙의 RHS에서 호출되거나 대화형의 상위레벨에서 호출된다. 규칙의 수행도중에서도 여러번 on ,off 상태가 변경될 수 있다. 그러나 check-facts on 는 이미 fact-list에 존재하는 이중적인 사실에 대해서 소급하여 제거시키지는 못한다. 그러므로 이 함수를 사용할 경우는 상당한 주의를 요한다. 최선의 사용방법으로는 check-facts가 on인 상태에서 시스템을 개발한 다음 check-facts을 off하여 여러 번 테스트한 후 정확한 결과가 얻으면 그 후에 check-facts을 off시켜 사용한다.

이런 기능은 또한 호출함수를 통해 on 또는 off 할 수 있다.

#### check\_facts

```
float set_check_facts(arg);
char *arg;
```

목적 : set\_check\_facts는 사실이 이중으로 선언되는 사실을 체크를 선택하는 함수. (on or off)

인수 : 단 하나의 인수로 "on" 이나 "off" 중 하나의 값을 가진다.  
"on"이 선언되면 선언한 사실과 fact-list와의 비교과 행해지고  
"off"가 선언되면 비교를 수행하지 않는다.

반환되는 값 : 반환되는 값이 1.0 이면 문제에 이상이 없고 0.0이면 인수에 문제가 있다.

#### 4. 사용자를 위한 인터페이스의 기능

K-CLIPS는 높은 이식성을 위한 단순하면서 대화형인 라인 타입의 인터페이스를 공급하고 있다. 기본적인 사용절차는 다음과 같다. 어떤 기본적인 에디터를 사용하여 규칙을 만들거나 에디터하고 파일로 저장한다. 그리고 에디터에서 빠져나와 K-CLIPS을 실행한 다음 K-CLIPS내에서 규칙을 포함하는 파일을 로드시킨다. K-CLIPS의 인터페이스 명령문은 현 시스템의 상태를 알수 있고 실행을 체크할 수 있다. 그리고 정보를 만들거나 제거시킨다.

#### 4.1 K-CLIPS 명령문

##### 4.1.1 환경 제어 명령문

다음 함수는 K-CLIPS 환경을 제어하는 명령문이다.

- (reset) K-CLIPS을 다시 세팅(setting)시킨다. agenda에 있는 모든 규칙과 fact-list에 있는 모든 사실을 제거시킨다. 그리나서 fact-list에 deffacts내에 정의된 사실을 만든다.
- (clear) K-CLIPS에서 모든 규칙과 deffacts을 제거한다. 그리고 agenda에 있는 모든 규칙과 fact-list에 있는 모든 사실을 제거시킨다.
- (run [<limit>]) 규칙을 실행하도록 하는 명령으로 만약 <limit>가 양수이면 <limit>만큼 규칙을 실행한 이후이거나 agenda에

속한 실행가능한 규칙이 없으면 실행이 중단된다. 만약 <limit>가 표시되지 않거나 음수일 경우는 agenda내의 모든 실행가능한 규칙이 없을 때까지 실행한다.

예) (run 3)

(save <file-name>) 현재 K-CLIPS내에 있는 모든 규칙과 deffacts를 선언된 파일에 저장한다.

예) (save "rules.txt")

(load <file-name>) 파일에 저장된 모든 규칙과 deffacts를 환경내에 로드시킨다.

예) (load "rules.txt")

(save-facts <file>) 현재 fact-list에 존재하는 사실을 지정하는 파일에 저장한다. 그러나 한 사실이 512자를 넘을 수 없다.

(load-facts <file>) 파일에 있는 정보를 K-CLIPS의 fact-list에 사실로서 추가하는 함수이다. save-facts에 의해 만든 파일이나 다른 ASCII 텍스트 파일들이 읽힐 수 있다. 각 라인은 분리된 사실로 취급되고 512자를 초과할 수 없다.

(help <path>) 온 라인 help 시스템으로 부터 도움이 되는 정보를 얻을 수 있다. 자세한 사항은 4.2에서 언급한다.

#### 4.1.2 시스템 상태 명령문

다음 명령문은 현재 K-CLIPS의 상태를 나타낸다.

- (facts) fact-list에 저장된 모든 사실을 나타낸다.
- (rules) 지식베이스에 저장된 모든 규칙의 이름을 나타낸다.
- (agenda) agenda에 있는 모든 규칙을 표시한다.
- (pprule <rule>) <rule>로 지정된 규칙의 내용을 나타낸다.  
예) (pprule bogus-name)
- (list-deffacts) 현재 K-CLIPS에 로드된 모든 deffacts블록의 이름을  
프린트한다.
- (ppdeffacts <ID>) <ID>라는 이름을 가진 deffact의 내용을 프린트한다.

#### 4.1.3 디버깅 명령문

다음 명령문들은 디버깅 작업을 도와준다.

- (watch <item>) <item>이 facts이면 모든 사실의 추가와 제거되는 사항을  
표시해 준다. 만약 <item>이 rules이면 규칙의 실행되는  
것을 표시하고 <item>이 activations 이면 모든 규칙이  
agenda에 추가되거나 제거되는 과정을 나타낸다. 그리고  
<item>이 all이면 facts, rules과 activation을 동시에  
선언한 것이다.  
예) (watch rules)

(unwatch <item>) <item>에 관한 watch명령을 제거한다.

예) (unwatch rules)

(matches <rule>) 규칙의 LHS 패턴에 매치되는 사실들의 리스트가 표시된다.

(dribble-on <file>) wkclips, wdialog, wdisplay, werror,

wtrace, wagenda와 stdout에 보내지는 정보를 지정하는

화일에 보내는 함수이다.

(dribble-off) 화일에 정보를 보내는 일을 중지한다.

#### 4.1.4 메모리 관리 명령문(memory management commands)

다음의 명령은 K-CLIPS의 메모리상태에 관한 정보를 제공한다.

(mem-used) K-CLIPS가 시작된 이후 사용한 바이트수를 표시한다.

(release-mem) K-CLIPS가 확보하고 있는 자유메모리(free memory)을 제거한다.

(conserve-mem <arg>) 정보의 간결한 형태로 저장하는 명령으로 <arg>는 on 또는 off중 하나를 가진다. 큰 시스템일 경우는 이런 함수는 상당히 메모리를 절약한다. 그리고 이 함수는 규칙이 로드되기 전에 사용해야 한다.

예) (conserve-mem on)

K-CLIPS의 메모리 관리는 다음에서 보다 상세히 언급하고자 한다.

#### a) K-CLIPS의 메모리의 사용방법

규칙, 사실과 내부의 데이터구조는 실행하는 동안에 다이내믹 메모리 (dynamic memory)로 사용된다. 또한 사실이 제거되거나 데이터구조가 없어진다면 활용할 메모리가 늘어난다. K-CLIPS의 메모리관리함수는 운영체제에서 메모리를 요구하고 한번 사용한 함수의 메모리를 저장한다. 그렇게 함으로서 운영체제에서 malloc함수의 호출횟수를 줄일수 있다. 즉, K-CLIPS함수에 의해 새로운 메모리가 필요하다면 K-CLIPS는 먼저 요구하는 형태의 구조를 지니고 활용할 수 있는 데이터버퍼를 체크한다. 만약 그런 구조가 발견된다면 그 구조가 저장된 포인터가 반환된다. 그렇지않으면 적절한 데이터 양만큼 malloc함수에 의해 확보된다. 그리고 새로운 포인터가 반환된다. 데이터 구조가 더이상 필요하지 않을 경우에는 그 메모리의 포인터가 재사용을 위해 K-CLIPS내에 저장된다. 이러한 메모리를 자유메모리(free memory)라고 한다. 실제적으로 이러한 자유 메모리가 상당히 많이 존재하면 이 메모리를 제거해야 한다. 만약 malloc함수의 반환되는 값이 NULL이면 K-CLIPS내부의 모든 자유 메모리를 모두 제거하고 다시 malloc를 호출해야 한다. 종종 규칙의 수행도중에 다음과 같은 에러메시지가

```
*** DEALLOCATING MEMORY ***
```

```
*** MEMORY DEALLOCATED ***
```

wdialog입출력 지시자에 나타나면 사용자는 자유 메모리를 제거해야 한다.

K-CLIPS에서의 메모리 할당은 malloc함수에 의해 하는데 어떤 기종의 컴퓨터는 malloc함수보다 하위레벨의 메모리할당함수와 제거함수를 사용한다. K-CLIPS의 메모리 할당함수와 제거함수는 sysdep.c화일에 있고 genalloc과 genfree로 불린다. 특수한

기종의 컴퓨터를 위해 malloc과 free를 기종에 알맞은 하위레벨의 함수로 전환해야 한다.

## B) 기본적인 메모리 관리함수

현재 K-CLIPS에서는 메모리 조절을 위한 함수를 제공하고 있다.

mem\_used

```
float mem_used();
```

목적 : K-CLIPS가 시작된 이후 사용한 바이트수를 표시한다.

인수 : 없다.

반환되는 값 : 실수; 사용된 바이트수

mem\_requests

```
float mem_requests();
```

목적 : mem\_requests함수는 운영체제에서 메모리의 할당을 원하는 횟수가 반환된다.

인수 : 없다.

반환되는 값 : 실수; K-CLIPS에게 메모리를 요구한 횟수

info. : mem\_used와 함께 사용하면 한 호출에 필요한 메모리의 양을 계산할 수 있다.

release\_mem

```
int release_mem();
```

목적 : release\_mem 함수는 K-CLIPS에 의해 가지고 있는 자유 메모리를 운영체제에 되돌려 준다.

인수 : 없다.

반환되는 값 : 없다.

info. : 이 함수는 사용자가 정의한 함수가 메모리를 요구하지만 malloc함수로 부터 어떤 값을 얻을 수 없을 경우 사용한다. 그러나 이 함수는 조심스럽게 사용해야 한다. 과도한 사용은 K-CLIPS가 많은 malloc함수의 호출을 유발한다. \*\*\* DEALLOCATING MEMORY \*\*\* 의 메시지가 출력된다면 이 함수를 사용해야 한다. 이 함수는 규칙의 RHS에서도 사용할 수 있다.

set\_conserve

```
float set_conserve(arg);  
char *arg;
```

목적 : set\_conserve함수는 사용자에게 읽기에 편한 정보의 형태로 저장할 것인지 아닌지를 결정할 선택권을 준다. 만약 pprule 함수를 사용하는데 규칙을 보기 편리한 형태로 프린트하지 않으면 상당한 양의 메모리를 절약할 수 있다.

인수 : 하나의 인수 사용. "on" 또는 "off"중 하나를 선택. "on"일 경우 보기 좋은 형태로 정보를 저장하는 것이 아니다. "off" 일 경우는 보기 좋은 형태로 정보를 저장한다.

반환되는 값 : 실수 ; 반환되는 값이 1.0이면 문제가 없고 0.0이면 인수에 문제가 있다.

info. : 이 함수는 상당히 메모리를 절약한다. 이것은 규칙의 로드전에 사용해야 하고 여러 번 호출할 수도 있다.

#### 4.1.5 그밖의 명령문

(assert <fact>) fact-list 에 사실을 정의한다.

예) (assert (this is a fact))

(retract <fact#>) fact-list로부터 언급한 사실들을 제거한다.

예) (retract 4) 또는 (retract 3 24 5)

(excise <rule>) 시스템으로부터 <rule>에 언급된 이름을 가진 규칙을 제거한다.

(undeffacts <name>) deffacts문장의 효과를 더이상 발휘하지 못하도록 한다.

<name>을 가진 deffacts구조에 있는 사실들은

reset을 하여도 fact-list에 정의되지 않는다.

(batch <file>) K-CLIPS의 대화형(interactive) 명령들의 배치작업을 가능하게 한다. 배치화일을 사용함으로써 여러개의 명령문을 한번에 사용할 수 있다.

예) (batch "rules.bat") "rules.bat"에는 여러 개의 규칙을 로드시키기 위해 여러 개의 load 명령이 있다.

(<name> [<<args>>]) 사용자가 정의한 함수이거나 K-CLIPS가 정의한 함수들은 상위레벨에서 직접 호출할 수 있다.

예) (system "dir")

(defrule <<rule>>) 새로운 규칙을 정의한다. 만약 똑같은 이름의 규칙이 존재한다면 새로운 규칙이 에러가 발생하더라도 이전의 규칙은 제거된다.

(deffacts <<facts>>) 새로운 deffacts 블록을 정의한다. 만약 똑같은 이름의 deffacts가 존재한다면 새로운 deffacts가 에러가 발생하더라도 이전의 deffacts은 제거된다.

3.2 에서 언급한 것처럼 이런 대부분의 명령문은 또한 규칙의 RHS에서 사용될 수 있다. 그러나 사용자는 이런 정보가 어떻게 표현되는가에 대한 이해가 필요하다. pprule, facts, agenda, 와 rules과 같은 함수로 부터 나오는 정보는 router system에 의해 여러 개의 출력장치로 출력을 할 수 있다. 그러나 화면상에 직접 전달되는 정보는 함수로 부터 반환되는 값을 지닌다. mem\_used나 산술함수가 이런 부유에 속한다. 후자에 속하는 함수는 규칙의 RHS에 사용할 수 있고 반환되는 값은 bind에 의해 변수

에 치환되고 변수가 프린트될 수 있다.

run, rest, excise, batch, clear 명령은 규칙의 RHS에 사용할 수 없다. 또한 defrule  
이나 deffacts 명령이 규칙의 RHS에 사용되면 에러가 발생된다.

## 4.2 HELP 시스템

### 4.2.1 온 라인 HELP 시스템

K-CLIPS는 상위레벨에서 온 라인 HELP기능을 제공한다.

#### a) HELP기능 사용방법

HELP기능은 항목들의 메뉴를 보여주고 사용자에게 항목을 선택하게 한다. 항목  
이 선택되면 그 항목에 해당하는 정보를 HELP파일에서 참조하게 된다. HELP기능은  
항목을 지정하여 사용해도 되고 지정하지 않고 사용해도 된다.

#### SYNTAX

(help [<path>])

여기서 <path>는 help tree 에서 찾고자 하는 항목에 이르는 완전한 path를 말한다.

path의 각 항목요소는 공백문자에 의해 구분되며 대,소문자를 구분하지 않는다.  
또한 항목요소의 이름은 완전한 이름으로 지정될 필요는 없다. 즉, 항목 메뉴에서 서  
로 구분될수있을 만큼의 철자만 지정되면 된다. (만일 지정된 철자가 항목사이에서 서  
로 구분되지 않으면 처음의 항목이 선택된다.) 예를 들어 (help app err run) 은  
(help appendices error\_messages run\_time errors) 를 지칭하게 된다.

HELP기능을 사용할때 몇가지의 특별한 요소이름이 있다.

- ^                   현재의 레벨에서 한 단계위로 올라 간다.
- ?                   path 의 끝에 지정되면 지정된 path의 현재 메뉴를 보여준다.
- <nil>              리턴 키(return key) 만 치는것으로 한 상위 레벨로 간다.

MAIN 에서 한 상위레벨로 가는것은 HELP기능에서 빠져 나옴을 의미한다. 디폴트상태에서는 help 기능은 MIAN 항목에서 부터 시작되며, 사용자는 메뉴에서 원하는 항목을 고르면 된다. 프롬프트라인(prompt line)은 언제나 현재 메뉴를 보여준다. HELP기능에서 상위레벨에서 빠져나가게 되므로 항상 레벨은 MAIN 항목에 위치하게 된다.

처음 HELP기능을 사용할때는 HELP화일을 내부 참조표에 올려야 하므로 처음 이후의 HELP호출때 보다는 시간이 많이 소요된다. 그러나 처음의 HELP호출이외의 호출은 상당히 빠르게 이루어 진다.

#### b) HELP화일의 조사

HELP기능은 실행중에 화일로 부터 그 정보를 얻게된다. 사용자는 시스템의 상황에 따라 HELP화일의 위치를 바꿀수도 있다. help-path 함수는 이러한 기능을 수행한다.

#### [SYNTAX]

(help-path [<help-file-name>])

아무런 이름도 주어지지않으면 현재의 HELP화일의 이름을 보여주게 된다. 만일 HELP화일의 이름이 주어지면 주어진 화일을 보여주게 된다. HELP화일의 위치 변경

을 영구화하려면 clips.h 을 변경하고 다시 컴파일을 해야만 한다. HELP기능에 사용된 tool들은 사용자외 요구에 맞는 HELP기능이나 정보제공 기능의 개발에 사용될 수 있다.

## 주.1 K-CLIPS 의 설치

K-CLIPS 팩키지(package)는 메뉴얼과 통상 MS-DOS 3.2 로 포맷된 여러장의 디스켓으로 구성되어 있다. 디스켓에는 K-CLIPS 실행 파일과 K-CLIPS 예제 그리고 K-CLIPS 소스코드가 들어있다. 사용자는 반드시 모든 디스켓을 복사하여 사용하여야 한다. 만일 K-CLIPS가 MS-DOS 기종에서 사용된다면 실행파일은 하드디스크 또는 다른 플로피 디스크에 복사되어 사용될 수 있다. K-CLIPS 의 사용은 어떠한 설치과정을 필요로 하지 않는다.

K-CLIPS을 변경하거나 다른 기종에서 사용하기를 원한다면 사용자는 소스코드를 그 기종에 옮겨서 새로운 실행파일을 만들어야 한다. K-CLIPS는 여러 기종에 쉽게 이식할 수 있게 되어있어 소스코드에 특별한 수정 없이 이식할 수 있다. K-CLIPS는 Kernighan 과 Ritchie 의 C 언어 컴파일러나 ANSI와 호환성이 있는 C언어 컴파일러를 가지고 있는 기종에서 실행되어야 한다. 아래에서 새로운 K-CLIPS 실행 파일을 만드는 과정을 설명한다.

1) 소스코드를 사용자의 시스템에 로드한다.

다음의 C 소스코드는 기본적인 K-CLIPS 시스템을 만드는데 필요하다.

clips.c	main.c	npsr.c	sysfun.c
usrint.c	parser.c	sysdep.c	analysis.c
rulecomp.c	textpro.c	math.c	structdef.h
constdef.h	clips.h	frame.c	

다음의 소스코드는 에디터를 포함한 시스템을 만드는데 추가적으로 필요하다.

edmain.c	edbasic.c	edterm.c	edmisc.c
----------	-----------	----------	----------

edsstruct.c ed.h

## 2) 모든 include문장을 변경한다.(필요할 시)

모든 ".c" 의 소스코드는 clips.h화일을 포함한다.

```
#include "clips.h"
```

clips.h화일은 다른 include화일은 포함한다.

```
#include "constdef.h"
```

```
#include "structdef.h"
```

에디터 소스코드는 ed.h화일을 포함한다.

```
#include "ed.h"
```

이러한 include문장은 사용할 컴파일러의 사양에 맞게 고쳐져야 한다.

## 3) K-CLIPS의 환경과 사양을 조정한다.

clips.h화일을 편집하고 원하는 옵션을 정한다. K-CLIPS는 각 기종에 맞는 컴파일러를 선택할 수 있다. clips.h화일의 첫번째 선택 사양은 사용될 기종에 관한 것이다. 디폴트는 어떠한 기종에서도 사용될수 있도록 하는 GENERIC으로 되어 있다. 사용자는 자신의 기종에 맞게 고쳐야 한다. 만일 사용될 기종을 모르면 GENERIC으로 조종되어야 한다. clips.h화일에는 사용용도에 맞는 여러가지 선택사양을 가지고 있는데 이의 참조는 clips.h화일에 있는 코멘트와 다음에 언급한 사항을 참조하기 바란다.

### < 참고사항 >

#### 컴파일러 지시자

K-CLIPS는 이식과 컴파일의 용이성을 위해 컴파일러 지시자를 사용한다. 컴파일

러 지시자는 K-CLIPS을 특별한 용도에 맞게 조정하는 일을 쉽게 해준다. 모든 컴파일의 옵션은 clips.h 파일에 정의되어 있는 선택사양에 의해 제어된다.

clips.h 파일의 첫번째 선택사양은 사용될 컴파일러/기종에 관한 것인데 GENERIC K-CLIPS에 맞추어져 있다. GENERIC K-CLIPS로 컴파일되면 특정한 기종에 한정된 기능들은 제외되나 어떠한 기종에서도 실행될 수 있다. 이 파일에 있는 다른 여러가지 선택사양은 사용될 컴파일러/기종의 특징을 나타내는 것으로 사용자의 용도에 맞는 사양은 ON(1)로 조정시키고 GENERIC 사양은 OFF(0)시켜야 한다. 사용될 컴파일러 기종의 타입을 제어하는 코드는 sysdep.c 파일에 존재한다.

clips.h 파일이 제공하는 다른 여러가지 선택사양은 다음과 같다.

EX\_MATH : 이 사양은 확장된 산술팩키지의 사용여부를 묻는다. 만일 이 사양이 off 되면 ( 0 으로 정하면 ), 실행파일은 약 20-30 K 정도 작아지게 되어 메모리가 제한된 기종에서도 실행될 수 있다.

CLP\_TIME : 이 사양은 K-CLIPS의 시간기능의 사용여부를 묻는다. 이 사양은 내부적인 테스트에 사용되는 데 규칙 실행의 속도에 관심이 있는 사용자는 on 시키는 것이 좋다.

TRACK\_MEMORY : 이 사양은 K-CLIPS 가 메모리 사용관리에 대한 정보를 조절한다. 만일 이 사양이 on 되면 mem\_used 와 mem\_requests 함수가 제 기능을 할 수 있으며, off 되면 men\_used 와 mem\_requests 함수는 항상 zero 값을 반환된다.

CLP\_TEXTPRO : 이 사양은 K-CLIPS의 텍스트 프로세싱(text processing) 기능을 제어한다. 사용자가 정의한 help 기능에서 fetch, toss, print\_region 의 기능을 사용하

려면 이 사양을 on 시켜야 한다. 또한 온-라인 help 기능을 이용하려면 이 사양을 on 시켜야 한다.

CLP\_HELP : 이 사양이 on 되면 K-CLIPS의 상위 레벨에서 help 기능을 사용할 수 있게 된다. 이 사양이 on 될때는 HELP\_FILE 사양에 원하는 K-CLIPS의 HELP화 일명을 지정해야 한다.

CLP\_EDIT : 이 사양은 통합된 MicorEMACS 에디터를 제어한다. 이 사양이 on 되면 에디터를 사용할 수 있으며, 만일 off 되면 에디터를 사용할 수는 없으나 대신 40K 정도의 메모리를 절약할수 있게 된다.

NOTE: 이 에디터는 모든 기종에서 사용될수 있는것은 아니므로 사용될수 있는 기종에 관해서는 clips.h 파일은 참조하기 바란다.

CLP\_RULE\_COMP : 이 사양은 규칙 컴파일 기능을 제어한다. 이 사양이 on 되면 rule은 C 코드로 컴파일되어 런타임 모듈에서 사용된다.

RUN\_TIME : 이 사양은 컴파일된 규칙을 사용할 새로운 K-CLIPS 실행화일을 만든다. 이 사양은 반드시 규칙을 컴파일한 후에 사용되어야 하며 이사양이 사용되면 약 90K 정도의 메모리가 절약된다. 이사양의 사용에 관해서는 주.4 를 참조하기 바란다.

#### 4) 모든 ".c" 파일을 목적 파일로 컴파일한다.

사용자의 기종에 맞도록 컴파일러의 옵션은 정하여 컴파일한다. ".h" 파일은 다른 파일에 의해 사용되므로 컴파일할 필요는 없다. 몇가지 옵션은 사용되는 컴파일러에 따라 지정되어야 한다. 많은 마이크로 컴퓨터는 메모리 사용에 있어서 large 또는 small 모델을 지원하는데 K-CLIPS는 항상 large모델을 사용해야 한다. 어떤 컴파일러는 변수

이름에 8자를 사용하는것을 디폴트로 되어있는 데 K-CLIPS는 변수이름에 적어도 10자를 사용하므로 이에 맞도록 옵션을 지정해야 한다.

만일 사용자가 정의한 함수를 사용하려면 이 함수가 정의된 소스코드를 컴파일하는 것은 물론이며 이 함수의 사용을 K-CLIPS가 알 수 있도록 main.c 파일의 usrfuncs의 정의를 이에 맞도록 고쳐야 한다.

5) 대화형 K-CLIPS 실행화일을 만든다.

대화형 K-CLIPS 실행화일을 만들기 위해서는 모든 목적화일을 링크한다.

K-CLIPS를 컴파일할때 몇가지 문제점

비록 K-CLIPS를 컴파일하는 것은 C 언어에 숙달된 프로그래머가 아니더라도 쉽게 할수 있으나 몇가지 문제점이 발생할 수도 있다. 이런 문제중의 하나는 시스템 라이브러리를 잘못 사용하는 것이다. 일반적으로 라이브러리의 지정은 컴파일러의 환경으로서 지정되는데 때때로 부적당한 라이브러리가 K-CLIPS 에 사용되어 문제를 야기시키게 된다. 예를 들어, 여러가지의 메모리 모델을 지원하는 컴파일러를 사용할 경우, 링커를 할 때는 반드시 컴파일 때의 메모리 모델을 사용해야 한다. 실수를 다루는 모델을 사용할 때도 마찬가지로 컴파일할 때의 모델을 링커할 때에 사용해야 한다.

## 주.2 K-CLIPS의 SYNTAX

### 주.2.1 사실의 제어(facts control)

(facts) : 모든 사실의 리스트를 프린트.  
(assert <pattern>) : 사실의 추가.  
(retract <<#>>) : 사실의 제거.  
(str\_assert "<<args>>") : 문자열 사실의 추가.  
(str\_cat <<"args">>) : 문자열의 결합.  
(save-facts "<file-name>")  
(load-facts "<file-name>")  
(deffacts <fact-name> "Optional" <<pattern>>)  
(undeffacts <fact-name>)  
(list-deffacts)  
(ppdeffact <fact-name>)

### 주.2.2 규칙의 제어(rules control)

- 추론사이클에서 매치된 사실의 변화를 최소화시키기 위해서

- 1) 특수한 패턴을 규칙의 상위패턴에 둔다.
- 2) 적게 사용되는 패턴을 상위패턴으로 둔다.
- 3) 변화가 심한 패턴은 하위패턴으로 둔다.

#### 1. 일반 사항

- 1.1 상위레벨에서 만들거나 화일을 사용하여 로드시킴.
- 1.2 응용 : pattern->agenda->priority->select->fire
- 1.3 규칙을 만든 후에 선언한 사실만 패턴매치 한다.
- 1.4 Refraction : 한번 실행한 사실은 다시 실행할 수 없다.

#### 2. 규칙의 상위레벨

- (agenda) : 실행 가능한 규칙들을 프린트.
- (rules) : 현재 로드된 모든 규칙을 프린트.
- (pprule <rule-name>)
- (load "<file-name>") : file-name에서 규칙을 로드시킴.
- (save "<file-name>") : 현재 로드된 규칙을 file-name에 저장.

### 3. 규칙의 실행

- 3.1 우선권 : 첫번째 패턴에 위치한다.  
10000에서 -10000 사이의 값을 값는다.(default=0)  
syntax : (declare (salience #))

### 3.2 RHS에서 사용하는 구조체

```
while : (while (<fun> <<args>>) (<<action>>))
if-then-else
    (if (<fun> <<arg>>)
      then <<pattern>>
      [else <<pattern>>])
```

### 4. LHS상의 패턴매치

#### 4.1 와일드카드

- ? : 단일필드
- \$? : 다중필드

#### 4.2 변수

- ?<name> : 단일필드
  - (1) 논리연산자
    - ~ : 부정           예) ~value
    - | : or             예) a|b
    - & : and            예) ?var&a|b, ?var&~a
  - (2) 조건함수  
?var&:(<fun> <<args>>)
  - (3) test 함수  
(test (<fun> [<<args>>]))

- \$?<name> : 다중필드

#### 4.3 패턴 제어

- 패턴 바인딩

- ?fname <- (pattern)    예) (retract ?fname)
- 패턴의 제한  
  and (explicit), or, not(negation)
- 패턴 확장연산자  
  =( <fun> <<args>> ) : evaluate

### 주.2.3 계층적인 지식표현

1. 클래스의 정의  
  (define\_class <class> (<subclass>))
2. 슬롯의 관리
  - 2.1 슬롯의 생성  
  (create\_slot <class> ^<slot> <value>)
  - 2.2 슬롯의 제거  
  (remove\_slot <class> ^<slot> <value>)
  - 2.3 슬롯의 추가  
  (add\_slot <class> ^<slot> <value>)
  - 2.4 슬롯의 수정  
  (modify <class> ^<slot> <value1> => <value2>)
3. 클래스의 패턴매치
  - (SUBCLASS\_OF <class 1> <class 2>)
  - (SLOT\_VALUE <class> <slot> <value>)
  - (slot\_of <class> <slot>)
  - (subclass\_of <class> <slot>)
  - (slot\_value <class> <slot>)

### 주.2.4 함수를 정의하는 기능

### 1. 함수의 정의

```
(defun <fun-name>
  (<parameter>)
  [body] )
```

### 2. 그 밖의 보조함수

(displayfun) : 새로 정의한 함수의 이름을 프린트 한다

(removefunc) : 정의한 함수를 제거

(ppfunc <fun-name>) : 선언한 함수의 내용을 프린트

## 주.2.5 시스템 제어

(clear) : 모든 사실과 규칙을 제거한다.

(reset) : initial-fact와 deffacts내의 모든 사실을 만든다.

(excise <"rule-name">) : 규칙을 제거한다.

(batch "<file-name>")

(system "<dos-command> <<args>>")

## 주.2.6 입출력 Syntax

### 1. 파일의 처리

1.1 (open <"physical-file-name"> <logical-file-ID> [<"file-access">])

- access mode : r(read only)(default), w(write only),  
r+(read and write), a(append only)

예) (open "animals.dat" animals "r")

1.2 (close [<logical-file-ID>])

- default : 열린 모든 파일을 폐쇄  
- 열린 파일을 보호

### 2. 입력 조절

2.1 (read [<log-name>]) : 단일필드 입력

logical-name default ; t(=stdin)

예) (assert (a =(read)))

2.2 (readline) : 한 라인을 입력한다

### 3. 출력 조절

- 3.1 (printout <item>) : stdout (\* old version)
- 3.2 (fprintout <log-name> "-----")
- 3.3 (format <log-name> "%-M.Nx" variable)
  - : left justification, M : field width, N[default:6] : # of digits
  - x : d(integer), f(float), e(exponential), g(general),  
o(octal), x(hexadecimal), n(carriage-return), s(string),  
%(%)

### 주.2.7 기타 기능

#### 1. 디버깅 syntax

- 1.1 (watch <arg>) : <arg> : fact, rules, activations, all
- 1.2 (unwatch <arg>) :
- 1.3 (matches <Rule-name>)

#### 2. 실행

(run [<#number>])

#### 3. 변수 생성

- 3.1 gen : (gen<#number>)
- 3.2 setgen : (setgen #number)

#### 4. 그 밖의 함수

- 4.1 bind : (bind <?variable> (+ <arg1> <args>))
- 4.2 length : 변수의 요소수 계산
- 4.3 nth : 지정하는 필드값을 계산
- 4.4 member : 요소가 있는 필드의 자리수 계산  
예) (member ?<list1> \$?<list2>)
- 4.5 subset : 부분 집합인가를 체크

### 주.2.8 계산 기능

#### 1. 기본적인 산술함수

- + : addition 예) (+ <arg1> <arg2> <<args>>)
- : subtraction 예) (- <arg1> <arg2> <<args>>)
- \* : multiplication 예) (\* <arg1> <arg2> <<args>>)

/ : division    예) (/ <arg1> <arg2> <<args>>)  
 \*\* : exponentiation    예) (\*\* <arg1> <arg2>)  
       (possible when compiled by EM package)

## 2. 확장된 산술함수

sin	asin	sinh	asinh
cos	acos	cosh	acosh
tan	atan	tanh	atanh
sec	asec	sech	asech
csc	acsc	csch	acsch
cot	acot	coth	acoth
pi	min	max	mod
log	log10	exp	sqrt
abs	trunc	deg-rad	rad-deg

## 3. 논리함수

| or not  
 && or and  
 || or or

## 4. 비교함수

eq : equal(any) functions  
 neq : not equal(any)  
 = : equal(numerical) function  
 != : not equal  
 >= : greater than and equal to  
 > : greater than  
 <= : less than and equal to  
 < : less than

## 5. 조건함수

(numberp <arg>)  
 (stringp <arg>)  
 (wordp <arg>)  
 (envep <arg>)  
 (oddp <arg>)

### 주.3 K-CLIPS의 에디터

K-CLIPS는 완전한 스크린 에디터인 MicroEMACS 에디터의 통합된 버전을 포함한다. K-CLIPS에서 에디터를 호출할 수 있으며, 에디터의 전부 또는 일부를 컴파일하거나, 일시적으로 에디터를 빠져나와 CLIPS로 돌아오거나, 완전히 빠져나올 수도 있다. 이 에디터는 완전한 스크린 에디터이므로 기종에 종속하게 된다. 현재로는 이 에디터는 VT100 또는 VT240 호환 터미널을 사용하는 VAX VMS 기종이나 TERMCAP을 지원하는 UNIX 시스템과 IBM-PC 또는 이것과 호환기종에서 사용될 수 있다.

에디터는 K-CLIPS에서 다음과 같은 코멘드로서 호출된다.

```
(edit ["<file-name>"])
```

파일명은 선택적이다. 만일 파일명이 지정되면 그 파일이 로드된다. 지정되지 않으면 파일을 로드하지않고 에디터가 호출된다. 에디터과정에서는 다음에 열거하는 모든 EMACS 의 명령어가 적용될 수 있다. 에디터를 빠져나와 버퍼를 지우기 위해서는 <Ctrl-Z> 또는 <Ctrl-X><Ctrl-C> 를 사용한다. 데이터는 버퍼에 그대로 두고 일시적으로 에디터를 빠져나오기 위해서는 <Ctrl-X> Q 를 사용한다. 버퍼의 일부를 컴파일하기 위해서는 컴파일할 부분을 표시하고 <Ctrl-X><Ctrl-T> 를 친다. 버퍼의 전부를 컴파일하기 위해서는 <Meta-T>를 사용한다. 에디터는 상당량의 메모리를 사용하게 되는데 에디터를 제거하는 선택사항은 clips.h 파일에 정의되어 있다.

#### MicroEMACS Editor

##### 1. Notation

Meta = Esc = Alt key = c - [  
Rubout = Delete = del key  
c-X-B : Type c-B when pressing c-X  
c-x-(S) : Type c-B and type c-S

## 2. Start

(edit ["<file-name>"])

## 3. Exit EMACS

c-X-C : save buffer and kill EMACS  
c-X-Z : kill EMACS  
c-C-Z : suspended EMACS

## 4. General Utilities

c-\_ : Help  
c-G : Cancel any command  
c-L : Refresh the screen  
c-C : Start a new interactive command shell (under UNIX, VMS)  
c-X-! : Execute a single external command to the OS  
c-U : Enter repeat count for next command  
c-Y : Yank(undelete) last text killed

## 5. Cursor Movement

### a. Character

c-F : Forward one command  
c-B : Backward one character

### b. Word

m-F : Forward one word  
m-B : Backward one word

### c. Line

c-A : Beginning of line  
c-E : End of line  
c-P : Previous line  
c-N : Next line  
c-X : Go to a special line number  
c-X-= : Show current cursor column and line number  
c-X-O : delete following blank line  
c-X-: : Go to a special line number  
m-! : Move current line to repeat count lines from top of window

### d. Buffer

m-< : Beginning of buffer

m-> : End of buffer

e. Screen

c-V : View the next screen

M-V : View the previous screen

6. deletion

a. Character

c-D : Character under cursor (<del>)

c-H : Previous character (<rubout>)

b. Word

m-<del> : Previous word

m-D : next word

c. Line

c-K : To end of line from cursor position

7. Blocks Manipulation

a. Mark

m-. or c- : set mark at current position

b. Manipulation

c-W : Delete region between cursor and last mark set.

c-X-X : Delete a buffer other than the current buffer.

c-X-B : Find the names of all buffers.

m-W : Copy region into kill buffer. The kill buffer is a buffer into which deleted text is put. The last text put into the kill buffer can be retrieved using the c-Y command.

8. Insertions and Searches

a. Insertions

c-I : Insert a tab

c-J : Insert CR/LF and indent the next line

c-M : Insert CR/LF

c-O : Insert line

c-Q : Quote the next character. This allows you to insert special control codes such as backspace or CR in the text. After the c-Q should be the special character to be inserted.

c-T : Transpose character. This si, switch the positions of the character at the cursor with the character on the left.

m-C : Capitalize first letter of word

m-L : make lowercase next word

m-U : Make uppercase next word

b. searches

- c-R : Reverse search
- c-S or m-J : Forward search
- c-X-R : Global reverse search and replace
- c-X-S : Global forward search and replace
- m-R : Query reverse search and replace
- m-S : Query forward search and replace

## 9. Save and Load

### a. Buffer

- c-Z : Save current buffer only and exit, quick save.
- c-X-(S) : Save current buffer into its file
- c-X-(W) : save current buffer with option to change name
- c-X-(C) : Exit without saving buffers

### b. File

- c-X-(F) : Find file. Load if not already in buffer
- c-X-(R) : Rename file. Change file name for buffer
- c-X-(V) : Visit a file. Load file in current window

## 10. Windows and Buffers

- c-X-1 : Display only current window
- c-X-2 : Split the current window
- c-X-B : Switch to a different buffer
- c-X-N : Move to next window
- c-X-P : Move to previous window
- c-X-(N) : Scroll window up by repeat count lines
- c-X-(P) : Scroll window down by repeat count lines
- c-X-Z : Enlarge current window by repeat count lines
- c-X-(Z) : Reduce current window by repeat count lines

## 11. Keyboard Macros

- c-X-( : Begin definition of keyboard macro
- c-X-) : End keyboard macro
- c-X-E : Execute a keyboard macro

## 12. Additional Capability

- c-X-T : Compile region and load it into KCLIPS
- c-T : Compile the current buffer and load it into CLIPS
- c-X-Q : Return to CLIPS, do not clear editor
- c-X-C : Return to CLIPS, clear editor
- c-X-M : File matching parenthesis for { } or []

## A.4 K-CLIPS의 런-타임(run-time) 프로그램

### A.4.1 규칙의 컴파일링

이 절에서는 K-CLIPS의 런-타임 모듈을 만드는 과정을 언급하고자 한다. 이 런-타임 프로그램 컴파일러는 모든 규칙을 하나의 실행파일로 만든다. 그리고 실행파일의 메모리 크기를 줄여 준다. 런-타임 모듈을 만드는 단계는 다음과 같다.

1) K-CLIPS을 실행하여 하나의 런-타임 모듈에 포함시킬 규칙들을 로드시킨다. 다음과 같이 함수 rule-to-c 을 호출한다.

```
(rule-to-c <file-name> <id>)
```

여기서 <file-name>은 문자열이고 <id>는 정수이다. 예를들어 만약 로드된 규칙화일이 "expert.clp" 이면 다음과 같은 변환 명령을 사용할 수 있다.

```
(rule-to-c "expert.c" 1)
```

이 명령은 규칙들을 c로 전환한 것을 출력파일 "expert.c"에 보내고 이 규칙들을 1이라는 모듈 id를 사용한다. 사용된 모듈 id에 대해서는 나중에 언급하기로 한다. 전환이 완결되면 K-CLIPS을 빠져나온다. 큰 규칙시스템(> 규칙 50)에 대해서는 이런 출력파일은 매우 크다. (> 200K)

2) 위에서 작성된 화일을 컴파일한다. 위 예의 경우는 작성된 출력파일은 "expert.c"이고 이 화일을 컴파일하여 목적화일("expert.obj")을 만든다.

3) clips.c 모듈내의 RUN\_TIME 이라는 컴파일 옵션을 1로 선언하고 K-CLIPS의 전



```
usrfuncs()
{ }
```

6) 사용자가 정의한 함수가 포함된 K-CLIPS의 전체 모듈과 함께 'C'규칙모듈을 링크시킨다.

7) 사용자의 규칙이 포함된 런-타임 모듈은 수행될 수 있다.

run-time 환경에서 사용할 수 있는 명령에는 다음과 같다. : clear, load, save facts, rules, agenda, pprule, list-deffacts, ppdeffact, watch,unwatch, matches, dribble-on, dribble-off,excise,undeffacts,batch. 그러나 다음에 언급하는 함수는 호출할 수 없다.  
: clear\_clips, load\_clips, displayfacts,printagenda,pretty\_print,pp\_deffact,set\_watch,excise,undeffacts.

#### A.4.2 컴파일된 규칙의 호환성

rule-to-c함수가 C 코드를 만들지만 이런 화일은 다른 기종의 컴퓨터에 적용을 할 수 없다. 일반적으로 한 기종의 컴퓨터에서 만든 코드는 char,int,pointer type 등이 동등한 바이트 크기를 가지고 있지 않으면 다른 기종의 컴퓨터에 사용할 수 없다. 특히 데이터의 크기와 data alignment는 실행과 컴파일에 문제가 발생한다. 기종에 관계없이 호환성을 가지기 위해서는 rule-to-c함수를 호출할 때 다음과 같은 네개의 인자를 덧붙인다.

```
(rule-to-c <file-name> <id> <char-size> <int-size> <float-size> <pointer-size>)
```

이런 인자는 원하는 기종의 문자,정수,실수,포인터의 크기를 선언한다. 특히 데이

타의 크기가 특이한 경우 이같은 사항을 고려해야 한다.

## 주.5 K-CLIPS 에러 메시지

K-CLIPS는 두가지 종류의 에러 메시지가 있다. 하나는 규칙의 실행에 관계되는 것과 규칙을 로드시킬 때 발생하는 에러 메시지가 있다. 여기서는 일반적인 에러 메시지를 나타내고 이것이 의미하는 바를 설명한다.

### 1. 런-타임 에러 메시지

Function <name> expected <number> arguments

이 에러는 함수가 지니는 인수의 개수가 틀리다는 것을 알려준다. 함수의 이름은 <name>이고 기대하는 인수의 수는 <number>이다.

Function <name> expected at least <number> arguments

이 에러는 함수가 기대한 함수의 인수 개수가 기대한것보다 적다는 것을 나타낸다. 함수의 이름은 <name>이고 기대하는 인수의 수는 <number>이다.

Function <name> expected no more than <number> arguments.

이 에러는 기대한 함수의 인자 수보다 많은 인자가 존재한다는 것을 나타낸다. 함수의 이름은 <name>이고 기대하는 인수의 수는 <number>이다.

Function <name> expected argument #<number> to be of type <type>

rfloat received a request from function <name> for argument #<number> which is not a number

rstring received a request from function <name> for argument #<number> which is not a string

이 에러는 함수의 인수형태가 틀리다는 것을 나타낸다.

rfloat received a request from function <name> for argument #<number> which is non-existent

rstring received a request from function <name> for argument #<number> which is non-existent

runkown received a request from function <name> for argument #<number> which is non-existent

이 에러는 기대한 인자 갯수보다는 적은 값을 가지는 것을 알려 준다.

## 2. 규칙 파싱(parsing) 에러 메시지

대부분의 파싱 에러는 화일로 부터 규칙을 로드시킬 경우나 상위레벨에서 직접 규칙을 만들 경우에 발생한다. 만약 K-CLIPS의 대화형 인터페이스에서 규칙을 로드시킬 때 에러가 발생하면 화면상에 메시지를 전해준다.

Missing a rule name!

이 에러는 규칙의 구조에서 이름이 없는 것을 표시한다.

A pattern begins with '(' or ?<var>

이 에러는 규칙의 LHS상의 패턴을 작성할 때 괄호를 열거나 변수를 사용하지 않아 발생한다.

LHS of rule <name> contains no patterns

이 에러는 규칙의 LHS에 패턴이 없을 때 발생한다.

Only a single pattern may appear after a not

이 에러는 not 구조체내에 다중 패턴을 가지 때 발생한다.

Found <token> when expecting => in <rule>

이 에러는 bind, fprintout와 같은 규칙의 RHS action이 => 이전에 나타날 때에 발생된다.

Expected ')' to finish a rule

이 에러는 규칙에서 '('을 남겨 놓았을 때 나타난다.

Missing <- after ?<var> in the LHS of <rule>

이 에러는 변수에 한 패턴을 치환시킬 때 '<-' 이 없을 때 발생한다.

Missing '(' after ?<var> in the LHS of <rule>

이 에러는 변수에 한 패턴을 치환시킬 때 '(' 이 없을 때 발생한다.

Missing closing ')' for test construct in <rule>

이 에러는 test블록에 ')'가 없을 경우에 발생한다.

Missing '(' after logical pattern operator in <rule>

이 에러는 and 나 or구조체 다음에 괄호를 사용하지 않았을 때 발생한다.

Unexpected token <token> found in the LHS of <rule>

이 에러는 규칙의 LHS에 bind나 fprintout과 같은 RHS의 함수수가 사용될 경우에 발생한다.

Expected a pattern element such as a variable, word, or number  
in the pattern of <rule>

이 에러는 패턴내에 =>이나 <-와 같은 심볼이 나타날 때 일어난다.

Declaration is illegal at this point in <rule>

이 에러는 선언문을 최상위에 놓지않고 하나의 패턴이후에 나타나거나 logical operator (and , or, not)내에서 나타나면 발생된다.

Missing a '(' after declare in <rule>

이 에러는 선언문에 '('이외의 다른 문자를 사용할 경우에 발생한다.

Missing function declaration for function <function> in <rule>

이 에러는 규칙에서 사용되는 함수를 이전에 정의되지 않았을 경우에 발생된다. 이것은 타이핑에러일 수 있고 또한 usrfuncs내에 선언을 하지 않았을 경우도 있다.

Unable to find variable <name> on RHS of <rule>

이 에러는 규칙의 RHS에서 나타난 변수가 LHS에서 치환되지 않거나 이전에 bind문장에 의해 치환되지 않았을 경우에 발생한다.

Unable to find variable <name> on LHS of <rule>

이 에러는 규칙의 LHS에서 치환되지 않았을 경우 발생한다. 이런 에러는 대부분이 test문장에서 발생한다.

Unable to find binder <name> on RHS of <rule>

이 에러는 규칙의 LHS에서 패턴을 치환되지 않은 변수를 사용하여 사실을 제거할 때 발생한다.

예)

```
(defrule foo ""  
  ?f<-(a)  
  =>  
  (retract ?g))
```

Missing deffacts name

이 에러는 deffacts블록의 이름이 생략되었을 경우에 발생된다.

Variable <name> is unbound in rule <rule>

이 에러는 규칙에서 치환되지 않은 변수를 사용할 경우에 발생한다. 이런 에러는 종종 bind함수에서 발생한다.

예)

```
(defrule foo ""
```

```
(fact ?x)  
=>  
(if (> ?x 3) then (bind ?y 4)  
    (assert (new-fact ?y)))
```

## 부 록 2 : 발표논문

**발표논문 1**

**논문제목 : " A Preliminary Model of I-BUILDS : An Intelligent  
Building Design System "**

본 논문은 1987년 8월 미국의 매사추세츠주의 Cambridge에서 열렸던 제 2차 공학문제에 있어 AI의 적용에 관한 국제학술대회(2nd International Conference on Applications of Artificial in Engineering Problems)에 발표되었던 것으로 I-BUILDS의 개념적인 모델에 대한 것이다.

# Knowledge Based Expert Systems in Engineering: Planning and Design

Editors:  
D. Sriram  
R.A. Adey



Computational Mechanics Publications

## CONTENTS

Planning with Constraints in Precision Machining <i>C. Neuman, M. Donath</i>	1	Knowledge-Based Analysis of Structural Systems <i>G.M. TarApayak, S.J. Fenves</i>	273
Management of the Design Process <i>P. Derrington</i>	19	An Expert System Architecture for Construction Planning <i>C. Hendrickson, C. Zecova-Gonzalez, D. Kehak, E. Baracco-Miller and P. Lim</i>	285
Blackboard Approach to Process Planning Problems <i>A. Murphy, T. Jagannathan, S. Goodrum</i>	35	The Scheme of Posidonius - Using Taxonomic Reasoning in Design <i>C. Pittaloni</i>	299
Strategies for Micro-Architecture Design <i>F.D. Brewer, D.D. Gajski</i>	51	EXLOG: An Expert System for Logic Synthesis in Full-Custom VLSI Design <i>M. Watanabe, M. Iwanoto, T. Yamanouchi, H. Matsuda, D.J. Littleboy, N. Masuda, M. Kaitaniyoshi and M. Imahashi</i>	315
A Generic Planning Model for LARGE SCALE Integrated Engineering Design <i>W.-H. Chung, D.A. Hertzfel</i>	67	A Preliminary Model of I-BUILDS: An Intelligent Building Design System <i>C.K. Choi, E.D. Kim</i>	331
A Meta-Planning Model for Diminishing Resource Problems <i>W.P.C. Ho</i>	85	An Expert System for Nuclear Fuel Cycle Plamis Design <i>M. Caratunio, P.A. Landeyro</i>	345
Acquiring and Representing Knowledge about Material Design <i>M.L. Farinacci, J. Huthage and M.A. Przystupa</i>	99	Knowledge Representation for Mechanical Systems Design <i>J.I. Dimitrat</i>	367
Function Minimization in an Automatic Digital Circuit Design System <i>J.-G. He, W.P.C. Ho, D.Y.Y. Yau, Y.H. Hu and H.J. Yu</i>	115	An Aid to Preliminary Design <i>N. Herly</i>	377
Hierarchical Abstraction for Process Planning <i>D.S. Nau</i>	129	LADY: A Logic-based Architectural Design Program System <i>Zs. Markusz, J. Rakosty</i>	393
COSSACK: A Constraint-Based Expert System for Configuration Tasks <i>F. Frayman, S. Mittal</i>	143	Expert System for Optimal Design of Structural Steel Members <i>S. Rajee, C.S. Kirisunmoorthy</i>	405
ROADLAB - A Constraint Based Laboratory for Road Design <i>S.M. Erwin, M.D. Gross</i>	167	Artificial Intelligence Based Computer Aided Design of Shell Structures <i>H.S. Satch</i>	421
Expert Systems and Prolog in Multiple Project Management <i>E. Guertin</i>	185	EXIST - An Expert System for the Innerplant Planning of Locations <i>M. Kleib, J. Herrmann</i>	429
Constraint Management in Conceptual Design <i>D. Soriano, D. Gosand</i>	211	A Knowledge Based Process Planning System <i>P. Hubbard, F. Mill and S. Harrington</i>	439
Explanation for Routing Design Problem Solving <i>A. Kasabali, D.C. Bracka</i>	225	An Expert System for Objects Plating in Three-Dimensional Space <i>R. Chantani, M. Tei, Inanc, P.M. Chentira and Y. Descale</i>	449
SPIKE: A Generic Design Standards Processing Expert System <i>W.J. Rasajski, P.E., T.E. Wang</i>	241		
A Knowledge-Based Computer-Aided Process Planning System <i>H.P. Hong</i>	259		

a task which can be completed by iterating analysis-design cycle beginning with the preliminary design. The main goal of this study is to construct an automated design system for building structure (I-BULLDS; an Intelligent BUILDing Design System) by synthesizing the conventional analysis/design CAD programs (BULLDS; an Integrated BUILDing Design System) 10, 11 and the knowledge-based system that is established by accumulating experts' experiences.

#### REVIEW OF BUILDING DESIGN PROCESS

It may be helpful to discuss a very general model for building design process prior to the discussion on any building design expert system. The building design process, which is one of most frequently arising engineering problems, begins with examining and defining the problem at hand. 12, 13. At the stage of problem definition, the conditions for innovative solutions are created and a systematic analysis of the objectives and the requirements to be met by any solutions should follow. The candidate solutions are then generated by some creative processes. Each candidate solution is evaluated with the requirements and compared with each other to find the best or optimum solution. If no solutions are possible, or if they cost too much, the requirements may have to be eased and/or the problem definition reevaluated. Since it is practically impossible to enumerate all the requirements or generate all the possible solutions, this process can be hardly mathematical. Therefore, the designer's judgment during the iterative optimum seeking process to decide which factors are more important and which candidate solutions are more plausible is of significant importance. Although this model of design process is more or less subject to personal style, the building design process can be summarized in four modules as shown in Figure 1.

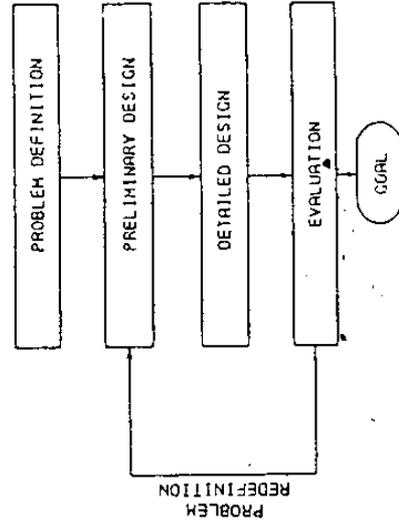


Figure 1. Representation of Building Design Process

#### A Preliminary Model of I-BULLDS: An Intelligent Building Design System

C.-K. Choi, E.-D. Kim  
*Department of Civil Engineering, Korea Advanced Institute of Science and Technology, Seoul 131, Korea*

#### SUMMARY

The purpose of this paper is to outline the conceptual development of I-BULLDS, which is an acronym for an Intelligent BUILDing Design System. In I-BULLDS the human knowledge acquired from experiences of experts in structural engineering is incorporated into the conventional building design system (BULLDS) to form a knowledge-based building design system. Some object networks of this knowledge are presented to illustrate the organizations of informations on building structures. The programming environment called GEPSE is used for computer implementation in which a mixed language programming is utilized. The state of work and the direction for further research are also discussed.

#### INTRODUCTION

Due to the rapid advancement of Artificial Intelligence techniques, knowledge based expert system has emerged as a new technology on this subject. 1-3 The expert system has become the very appropriate tool for the structural design problems that rely much on designer's expertise and experience. Improving the current pattern of computer usage in structural engineering, i.e., mainly numerical computation, to include symbolic processing has received increasing attention in recent years. 4-9

In general, the structural design problem requires more experiences and intuition at the preliminary design stage than at the detailed analysis/design stage. Especially for building structures, one may face at first with the problem of selecting optimum structural system, i.e., a system obtained implicitly through the designer's expertise, preference and existing documents etc. Accumulated knowledge of experts that is obtained through long years of experience in this area can be transformed into the knowledge base and used as a tool or an assistant for the building design problems in case that the established theories, handbooks or experts are not readily found. Thus, it is possible that with an expert system structural engineers can obtain the optimal building structure by selecting one from more alternatives than with the conventional method.

The HI-RISE<sup>9</sup> may be the earliest effort for the preliminary design using knowledge-based system. The building design is

Problem definition module

A successful building design work is usually done by combined efforts of those people from different expertises, i.e., client/owner, architect, structural engineer and cost estimator, etc. Due to the large number of experts involved in the planning stage, this preliminary design process is a very large scale job. The concept of structural design is to transfer the applied force safely to the foundation satisfying the design constraints such as cost, geometry, performance, and serviceability, etc. The approach which many experts take in their building design is a Top-Down method, in which determination of the overall concept precedes to the detailed determination, e.g., member properties.

Preliminary design module

At this stage, all structural data are synthesized, feasible configurations are generated, and the best candidate is selected based on the designer's intuition and judgment.

Detailed design module

This stage involves the determination of structural response of the selected configuration with the assumed initial properties of members to the applied excitations. Then the detailed design of the structures is performed based on the results of the analysis.

Evaluation module

When the process in the above modules is completed, the given configuration is evaluated with respect to the effectiveness and optimality of total system. If it is not satisfactory, the task is sent back to the preliminary design module through problem redefinition.

**BUILDS: A CONVENTIONAL CAD PROGRAM**

BUILDS, which is being developed at the Korea Advanced Institute of Science and Technology, is an integrated design system (from analysis to drafting) for building structures which is designed to meet the needs of the computer aids in structural engineering practice. The system, which is now in the advanced stage of development, includes the following sub-systems each of which is assigned to have various computational capabilities as follows;

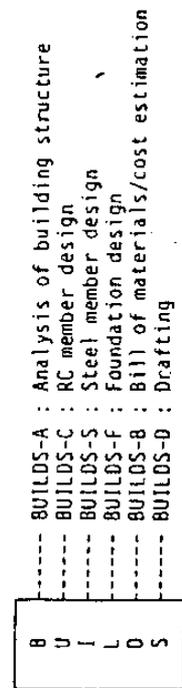


Figure 2 Sub-systems of BUILDS

The structural analysis/design by BUILDS is initiated with the selection of an initial configuration of the building structure that is modeled as an assembly of planar frames in arbitrary directions interconnected with rigid slabs at each floor level. The simplified three-dimensional analysis of a building structure with the forementioned model is performed to get the response of the structure for given excitations. In the structural analysis by BUILDS-A the sequential nature of gravity load application is considered.<sup>14</sup>

Based on the response above, BUILDS performs the structural member design and the analysis-design cycle is repeated until optimal solution (design) is obtained. The design sub-systems of BUILDS (BUILDS-C/S/F) emphasize the application of optimal design theory to the practical design of building structures with elements of discrete sections. The more sophisticated optimization model requires so many variables and complicated descriptive functions that only simple structures may be treated at a reasonable expense. In order to avoid this difficulty, the optimization of total structure is carried out with two separate stages splitting the subjective constraints into two categories, i.e., 1) the constraints to be considered in the first stage optimization (element design) such as stress constraints and local member deflections and 2) the constraints to be considered in the second stage optimization (whole structure design) such as structural displacements and natural frequencies.

Then BUILDS proceeds with cost estimation (BUILDS-B) and if the cost is satisfactory, drafting (BUILDS-D) is made to complete the entire process. If not, BUILDS reiterates the analysis-design analysis-design-cost estimation cycle.

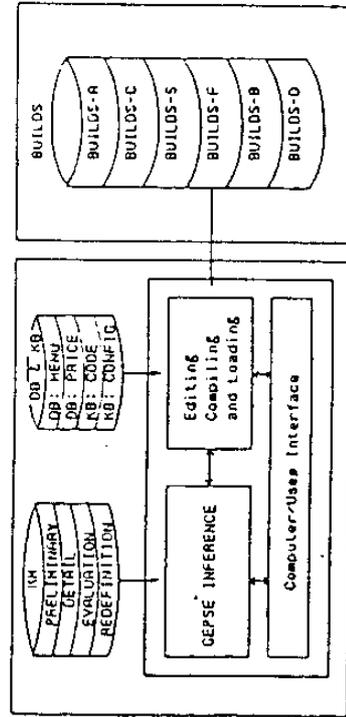


Figure 3 I-BUILDS Architecture



Table 1 Example of Data Generation of Floor Plan

Initial Data	Generated Data
<ul style="list-style-type: none"> <li>- Building Material.</li> <li>- Height, Width, and Length.</li> <li>- Number and Size of Span.</li> <li>- Usage, etc.</li> </ul>	<ul style="list-style-type: none"> <li>- Frame ID, Location.</li> <li>- Line ID, Location of Column, Beam and Slab.</li> <li>- Wall/Truss ID, Location</li> </ul>

Detailed design module

This phase involves the picking up the properties of members of the selected configuration determined in the preliminary design process. This phase also involves the processes of preparation/checking of input data file, analysis/design, checking results and feedback control, etc.

Stage 1 For a chosen candidate configuration of building, it is required to introduce initial values of various design parameters such as beam depth and slab thickness. IDEALIZED-MODEL KU generates them heuristically from general span/depth ratio or some approximate analytic algorithms such as portal method. Then the input data is prepared and checked automatically by INPUT-FILE KU for the later use in structural analysis. To prepare the input file, job control data is input directly by user or by default values. Then the input file is checked using FORTRAN statements implemented in BUILDS program. Figure 6 and Table 2 show some information generated in this stage.

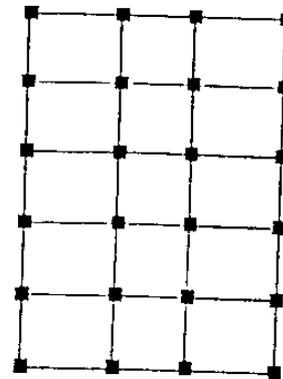


Figure 6 Specification of Member Properties

Table 2 Data Generation

- No. of Section Prop. of each component.
- Section Properties of member, i.e., Inertia of Moment, Area, Depth, Thickness, etc.
- Optional Control Data
For an example, a variable IPLOT determines whether the output displayed is copied to a hard copier or not.

Stage 2 With the input file established in the stage 1, the structural analysis is performed by BUILDS-A and its response is determined for the applied load.

Stage 3 Based on the results obtained by the structural analysis, improved member design or determination of member properties in accordance with the current regulations and codes of practice is performed and the initial members are updated (optimized). In this process carried out by subsystems BUILDS-C/S/F, emphasis is placed on the optimization of individual members of the structural system.

Stage 4 With the individual members optimized, the acceptability of design is checked with constraints such as structural drift, overturning moment and member deflection/stress at this stage by CODE DB. If acceptable, the detailed design proceeds to the next stage. Otherwise, sensitivity vectors, which was obtained in the precedent stage, is retrieved to identify the most sensitive member(s) to correct the violated constraints. Then goes to stage 1 to update the input data with newly modified section properties using REFINE-MODEL KU. The modification of solution is continued until the optimum solution is found.

Evaluation module

The effectiveness of the selected configuration is evaluated in this stage. If it is appropriate, drafting is performed. If not, Preliminary design module is taken up to modify the configuration and the process is reiterated.

Stage 1 Bill of materials and cost estimation of the selected configuration are made by BUILDS-8 using PRICE DB.

Stage 2 At this stage, SUGGEST-ALTERNATE-DESIGN KU evaluates the design effectiveness based on the cost, structural performance, etc. and a decision has to be made whether to modify the current design or not, both at the total system level and at subsystem levels. If design is acceptable, drafting is performed using BUILDS-D. If a modification is needed, it is performed in redefinition module.

Redefinition module

In modifying the building configuration, some information is needed which tells in which direction a move should be made. To decide the direction of the move, decision should be made by REW-CONFIGURATION KU whether the selected configuration should be modified or an entirely new configuration should be introduced. This also requires much knowledge of experts. In this process, the results of a certain structural configuration is saved in a knowledge base named CONFIG and then efficiently used to retrieve the information with which the selection and the evaluation of alternatives are made. This also contributes to nourishing the knowledge base of I-BUILDS.

KNOWLEDGE ACQUISITION

The knowledge base in this paper is first established with knowledge in printed text, regulations and codes of practice. Then to get more information and/or qualify this knowledge, the experts' experience and knowledge are solicited. This paper classifies the knowledge characteristics of the building structural system into three main categories as shown in Figure 7.

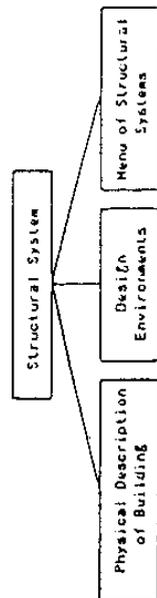


Figure 7 Knowledge Classification

Physical description of building  
 The design goal of structural system is to develop a system that minimizes the cost required to resist all kinds of loads applied to the building and satisfies the building functional requirements such as architectural, mechanical and serviceability requirements.

The simple way to model the building structural system is to assume that the system consists of mainly framing systems (2-D) which resist both gravity and lateral loads and are interconnected by slab systems which transfer the gravity load (floor load)

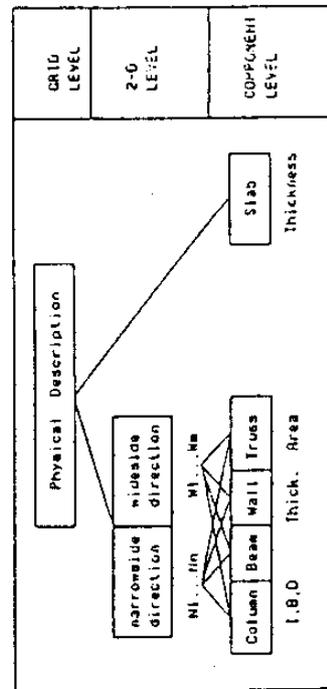


Figure 8 Physical Description of Building

to the framing systems. These 2-D subsystems are physically composed of primitive structural components such as columns, beams, walls and trusses. The physical components which construct the structural system are presented in Figure 8.

Design environments

This knowledge is mainly related to the design constraints as shown in figure 9 with which the properties of structural components are determined and the factors which effect the determination of overall structural systems. Regulations or codes affect the determination of the loading conditions, constraints of member deformation/stress. Related subsystem includes other building functional system such as mechanical/service, electric, construction and foundation, which play an important role in determining the structural system. Owner's need is also another factor which contains the requirements such as building material, usage, etc.

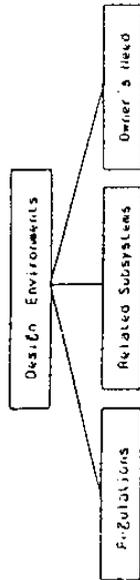


Figure 9 Design Environments

Menu of structural systems

This increases the structural design capability of I-BUILDS in selecting initial 2-D frames by providing the user with the editing mode in which he can select and modify the initial frames from MENU in accordance with his preferences. This process becomes possible by storing in the database called MENU the merits and faults of 2-D frames and their components which can be retrieved as needed. The items of this kind are shown in Figure 10.

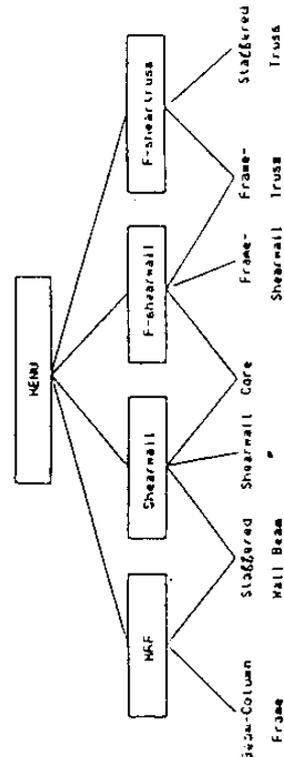


Figure 10 Menu of Structural Systems

## COMPUTER IMPLEMENTATION

The core of I-BUILDS shown in Figure 3 mainly consists of GEPSE-INFERENCE, DB, KB and BUILDS program. The executable programs under I-BUILDS are the execution files created by the meta-rules which edit, compile and/or load KB, DB and BUILDS as shown in the following example meta-rule.

```
IF (Input file is completed and Analysis is required)
THEN(Load input file and BUILDS-A, and Run BUILDS-A)
```

After the meta-rule like above example is applied, the information which is needed for next stage is extracted. It is required to transfer the structural displacement acquired from analysis to CODE KU in order to check if the results satisfy the code requirements. Because the programs or systems in I-BUILDS are written in different languages, this process requires mixed language programming. While the basic language used in GEPSE is C, KB is written in OML (Object Network Language), DB in C, and BUILDS in FORTRAN. To achieve mixed language programming, I-BUILDS utilizes Microsoft C (version 3.0 or later) and FORTRAN (version 3.3 or later) which are designed to be linked with each other. 17, 18 Therefore, it is possible for GEPSE in C to control, compile and run BUILDS in FORTRAN, and communicate with each other systems.

The key elements of GEPSE are an Object Network Language that simplifies the construction of objects and rule bases, function libraries, interface packages and meta-rule capability. Since C programs are known to be more portable and generally more efficient when there is a significant amount of numerical processing, GEPSE written in C is effectively utilized in building I-BUILDS. GEPSE solves the problems by problem reduction approach which uses meta-rule, sub-goal and sub-rule bases. Thus, the solution process is represented by a sequence of tasks which are defined in terms of sub-goals, that is, control is passed from task to task until the final design is achieved. The selection of sub-sets of knowledge requires the division of knowledge into knowledge packets, i.e., KUs, DBs, and KBs. Grouping knowledge according to its function results in a modular structures and allows parallel development of the various knowledge units. The knowledge base of I-BUILDS is constructed by four modules each of which represents the design stage as discussed in I-BUILDS ORGANIZATION.

GEPSE uses an associative triple, that is, OBJECT-ATTRIBUTE-VALUE to represent their static knowledge (facts). As an example, the syntax of assigning height ATTRIBUTE of building OBJECT a VALUE is as following statement, i.e., {set height building (read "How tall is it?")}. And the syntax representing active knowledge (rule) in GEPSE is shown as a following example. The example rule selecting 2-D frames is as follows.

```
(make-rule I-system)
  (and (about (get nst owner's need) 40)
        (is? material owner's need concrete)
        (is? usage owner's need commercial)
        (known? gravity structure)
        (unknown? lateral structure)
      )
  (do (set lateral structure frame-shearwall)
      (explain lateral)
    )
  )
```

This means,

IF the story of the building is about 40 and material of the building is concrete and usage of the building is commercial and the properties of the vertical system is known and those of the lateral system is unknown THEN get frame-shearwall system for lateral system and explain the reason.

## CONCLUSION

The conceptual model of constructing an Intelligent Building Design System (I-BUILDS) by combining GEPSE and BUILDS is presented in this paper. The mixed language programming is utilized to link these two systems which were written in different languages. The knowledge modules and units of the system are under development in parallel. Although the development of I-BUILDS is still in its infant stage and requires a lot of efforts to be completed, it seems to have a good prospect of success.

## REFERENCES

1. P. Harmon and D. King (1985), Artificial Intelligence in Business: Expert System, John Wiley and Sons, New York.
2. D. Waterman (1985), A Guide to Expert Systems, Addison Wesley Publishing Company.
3. S. Weiss and C. Kulikowski (1984), A Practical Guide to Designing Expert Systems, Rowman and Allanheld, Totowa, NJ.
4. M. F. Rooney and S. E. Smith (1983), Artificial Intelligence in Engineering Design, Computers and Structures, Vol.16, No. 1-4, pp.279-288.
5. D. Sriaram, M. L. Maher and S. J. Fenves (1985), Knowledge-Based Expert Systems in Structural Design, Computers and Structures, Vol.20, No.1-3, pp.1-9.
6. W. J. Rasdorf and G. C. Salley (1985), Generative Engineering-Database-Toward Expert Systems, Computers and Structures, Vol.20, No.1-3, pp.11-15.
7. D. Sriaram (1986), DESTINY: A Model for Integrated Structural Design, Artificial Intelligence, Vol.1, No.2, pp.109-116.
8. S. J. Fenves and J. H. Garrett (1986), Knowledge Based Standard Processing, Artificial Intelligence, Vol.1, No.1, pp. 3-14.

9. M. L. Maher (1985), HI-RISE and Beyond: Directions for Expert Systems in Design, Computer Aided Design, Vol.17, No.9, pp. 420-427.
10. C. K. Choi and E. D. Kim (1985), BUILDOS-3 : A Three Dimensional Analysis Program for Building Structures, Proceedings of the Second International Conference on Civil and Structural Engineering Computing: CIVIL COMP0 85, London, England.
11. C. K. Choi and H. W. Lee (1986), Three-Dimensional Analysis and Optimal Design of Building Structures, Proceedings of 10th Triennial CIB Congress, Washington D.C.
12. ASCE, Monograph on Planning and Design of Tall Buildings, Vol.PC(1981), Vol.SC(1980), Vol.CL(1980). Vol.SB(1979), Vol.CB(1978), New York.
13. W. Schueller (1977), High-rise Building Structures, John Wiley and Sons, New York.
14. C. K. Choi and E. D. Kim (1985), Multistory Frames under Sequential Gravity Loads, Journal of Structural Engineering, ASCE, Vol.113, No.11, pp.2373-2384.
15. F. Chehayeb and J. Connor (1985), GEPSE-A Computer Environment for Engineering Problem Solving, Research Report R86-11, Dept. of Civil Engineering, MIT.
16. ACI Committee 442 (1971), Response of Buildings to Lateral Forces, ACI Journal.
17. Microsoft Corporation (1985), Microsoft FORTRAN Compiler for the MS-DOS Operating System, User's Guide.
18. Microsoft Corporation (1985), Microsoft C Compiler for the MS-DOS Operating System, User's Guide.

## 발표논문 2

논문제목 : " An Expert System for Analysis/Design of Building  
Structure "

본 논문은 1988년 4월 홍콩에서 열렸던 제 4차 고층건축 국제학술대회  
(Fourth International Conference on Tall Buildings)에 발표되었던 것으로  
I-BUILDS의 개념적인 모델에 관한 것이다.

# Proceedings of the Fourth International Conference On Tall Buildings

Volume I

第四屆高層建築國際會議論文集

第一冊

Hong Kong & Shanghai  
April/May 1988

*Edited by*

*Y. K. Cheung P. K. K. Lee*

SESSION 5D

- A LIGHTWEIGHT FLOOR SYSTEM FOR HIGH-RISE CONSTRUCTION 502  
J.L. MEEK & D. STONE
- TWO-WAY SLAB-BEAM FLOOR SYSTEM FOR TALL BUILDINGS:  
EXPERIMENTAL MOTIVATION OF ITS ULTIMATE LIMIT STATE  
DESIGN 505  
M. KWIECINSKI
- ANALYSIS OF PARTIALLY PRESTRESSED BEAM-COLUMN  
ASSEMBLIES IN PLANAR FRAME SYSTEM 511  
D. CAPECCHI, D. GALEOTA & M.M. GIAMMATTEO
- BANK OF CHINA BUILDING - HONG KONG, TEMPORARY WORKS FOR  
BASEMENT CONSTRUCTION 518  
D. SUFF, M. PEARSON & K.Y. CHOI

SESSION 6A

- EFFECT OF AXIAL DEFORMATION ON BEHAVIOUR OF FRAME-CORE  
WALL BUILDINGS 526  
S.L. LEE, S. SWADDIWUDHIPONG & S. PIRIYAKOONTORN
- COUPLED SHEAR WALLS WITH STIFF ZONES 533  
H.C. CHAN & J.S. KUANG
- ASSESSMENT OF THE ACCURACY OF SOME APPROXIMATE  
METHODS OF ANALYSIS OF TALL BUILDINGS 538  
P. BHATT & Y.K. TO
- ANALYSIS OF FREE VIBRATION OF WALL-FRAMES USING STRIP  
ELEMENTS 542  
Z.W. ZHAO & Z.X. CAI

SESSION 6B

- THE USE OF EXPERT SYSTEMS TO CONSERVE ENGINEERING  
EXPERTISE 546  
A. DUNN
- REALIZATION OF EXPERT SYSTEM OF TALL BUILDING DESIGN 552  
A. MIYAMURA & Y. KOHAMA
- AN EXPERT SYSTEM FOR ANALYSIS/DESIGN OF BUILDING  
STRUCTURES 557  
C.K. CHOI & E.D. KIM

SESSION 6C

- LIQUEFIED PETROLEUM GAS INSTALLATION FOR TALL BUILDINGS 563  
S.Y. WONG
- LATEST TECHNIQUES IN PREVENTING VIBRATION AND NOISE  
TRANSMISSION FROM HVAC EQUIPMENT IN TALL BUILDINGS 570  
N.J. MASON
- A CO-ORDINATED APPROACH TO EARTHING AND BONDING IN TALL  
BUILDINGS 577  
J. BURNETT

# AN EXPERT SYSTEM FOR ANALYSIS/DESIGN OF BUILDING STRUCTURES

C.K. CHOI  
Dept. of Civil Engineering  
Korea Advanced Institute  
of Science and Technology  
Seoul 131, Korea

E.D. KIM  
Dept. of Civil Engineering  
Korea Advanced Institute  
of Science and Technology  
Seoul 131, Korea

## SUMMARY

The purpose of this paper is to outline the conceptual development of I-BUILDS (an Intelligent BUILDing Design System). Incorporating the human knowledge acquired from experts' experiences in structural engineering into the conventional building design system (BUILDS) to form a knowledge-based building design system, I-BUILDS automates all phases of the building design process. Some object networks are presented to illustrate the organizations of classified information on building structures. I-BUILDS utilizes the mixed language programming technique to combine an existing programming environment (GEPSÉ) written in C with BUILDS in FORTRAN.

## INTRODUCTION

The main phases of the engineering design process consist of 1) Specification of the problem domain, 2) Preliminary design, 3) Analysis, 4) Detailed design, and 5) Evaluation. In general, the designer arrives at the best appropriate design after carrying out a number of trials. While computers have been used extensively for the advanced analysis and detailed design, accumulated knowledge of experts obtained through long years of experiences has been utilized for the preliminary design and evaluation process. By conventional algorithmic programming technique, it is difficult to represent heuristic knowledge which is mostly obtained implicitly through the designer's expertise, preference and existing documents etc. But the rapid advancement of Artificial Intelligence have made it possible to formally represent the so called ill-structured knowledge [Ref.1-3]. The knowledge based expert system has become the very appropriate tool for the structural problems that much rely on designer's expertise and experiences [Ref.4-9]. The HI-RISE [Ref.9] may be the earliest effort for the preliminary design utilizing Artificial Intelligence technique.

In developing the integrated knowledge based design system, the existing conventional CAD programs are combined with experts' knowledge. The main goal

of this study is to construct an automated design system for building structures (I-BUILDS; an Intelligent BUILDing Design System) by synthesizing the conventional analysis/design CAD programs (BUILDS; an integrated BUILDing Design System) [Ref.10,11] and the knowledge-based system that is established by accumulating experts' experiences.

## BUILDS: A CONVENTIONAL CAD PROGRAM

BUILDS, which is being developed at the Korea Advanced Institute of Science and Technology, is an integrated design system (from analysis to drafting) for building structures which is designed to meet the needs of the computer aids in structural engineering practice. The system, which is now in the advanced stage of development, includes the following sub-systems each of which is assigned to have various computational capabilities as follows;

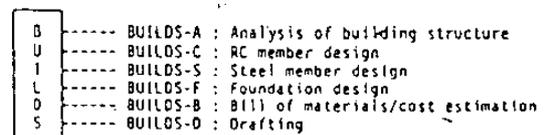


Figure 1 Sub-systems of BUILDS

The structural analysis/design by BUILDS is initiated with the selection of a candidate configuration of the building structure that is modeled as an assemblage of planar frames in arbitrary direction interconnected with rigid slabs at each floor level. The linear three dimensional analysis of the building structure with the forementioned model is performed to get its responses for given excitations. In the structural analysis by BUILDS-A, the sequential nature of gravity load application is considered [Ref.12].

Based on the responses obtained by BUILDS-A, other sub-systems of BUILDS perform the structural member design and the analysis-design cycle is repeated until optimal solution (design) is obtained. The sub-systems of BUILDS for detailing (BUILDS-C/S/F) emphasize the application of optimal design theory to the practical

design of building structures with elements of discrete sections. The more sophisticated optimization model requires so many variables and complicated descriptive functions that only simple structures may be treated at a reasonable expenses. In order to avoid this difficulty, the optimization of total structure is carried out with two separate stages splitting the subjective constraints into two categories, i.e., 1) the constraints to be considered in the first stage optimization (element design) such as stress constraints and local member deflections and 2) the constraints to be considered in the second stage optimization (whole structure design) such as structural displacements and natural frequencies.

Then BUILDS proceeds with cost estimation process (BUILDS-B) and if the cost is satisfactory, drafting (BUILDS-D) is followed to complete the entire process. If not, BUILDS reiterates the analysis-design-cost estimation cycle with the modified structural model.

#### MODEL OF BUILDING DESIGN PROCESS[Ref.13,14]

It may be helpful to discuss a very general design methodology for building structures prior to the discussion on constructing the design expert system. The building design process, which is one of most frequently arising engineering problems, begins with examining and defining the problem at hand. At the stage of problem definition, the conditions for innovative solutions are created and a systematic analysis of the objectives and the requirements to be met by any solutions should follow. The feasible solutions are then generated by some creative processes. Each candidate solution is evaluated with the requirements and compared with each other to find the best or optimum solution. If no solutions are possible, or if they cost too much, the requirements may have to be eased and/or the problem definition reevaluated. The approach which many experts take in their building design is a hierarchical approach, in which determination of the overall concept (e.g. building plan) precedes to the detailed determination(e.g. member properties). Although this design process model is more or less subject to personal preference, the building design process can be summarized in four or five phases as following discussions.

#### Problem Definition and/or Redefinition Phases

A successful building design work is usually done by combined efforts of those people from different expertises, i.e., client/owner, architects, structural engineers and cost estimators etc. Due to the large number of experts involved in the planning stage, this design

process is a very large scale job. Therefore, the problem domain will be confined to designing the structural systems.

#### Preliminary Design Phase

At this stage, all the conditions for building design are synthesized and all the feasible solutions are generated. Then the best candidate is selected based on the designer's intuition and judgment.

#### Detailed Design Phase

In order to analyze the chosen candidate, it is required to expand it and to introduce more detail into various design parameters. This stage involves the determination of structural response of the selected configuration with the assumed initial properties of members to the applied excitations. Then the detailed member design of the structures is performed based on the results of the analysis.

#### Evaluation Phase

When the process in the aforementioned phases is completed, the selected candidate is evaluated for its effectiveness and optimality in comparison with requirements. If it is not satisfactory, the control is sent back to the Problem Definition or the Preliminary Design Phase through problem redefinition. Otherwise the design ends with satisfying the goal.

#### I-BUILDS ORGANIZATION

In order to provide the BUILDS users with consultations and advices at the different phases of building design process, the development of an knowledge based expert system is desirable. Renovating the existing BUILDS program in accordance with the aforementioned building design model, a totally automated building design system utilizing knowledge-based expert system can be constructed. BUILDS, which is a conventional CAD system written in FORTRAN, manipulates mainly numerical computations while the knowledge-based expert system does the symbolic processing.

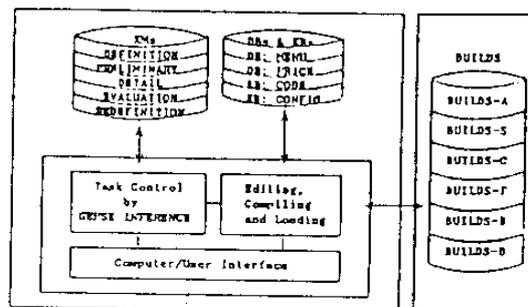


Figure 2 I-BUILDS Architecture

I-BUILDS consists of BUILDS and the knowledge based systems which is built under GEPSE shell [Ref.15]. The knowledge base is divided into five knowledge Modules (KMs) that communicate through working memory controlled by meta-rules. KMs are also composed of several Knowledge Units (KUs) in accordance with design steps while the Data Bases (DBs) and BUILDS subsystems in accordance with their functions. Figure 2 represents the architecture of an automated building design system named as I-BUILDS (Intelligent BUILDS) which is constructed by incorporating BUILDS and the knowledge based system into one framework.

I-BUILDS TASK CONTROL

The task control of I-BUILDS is accomplished by using meta-rules and discussed in detail in the following subsections according to job sequence.

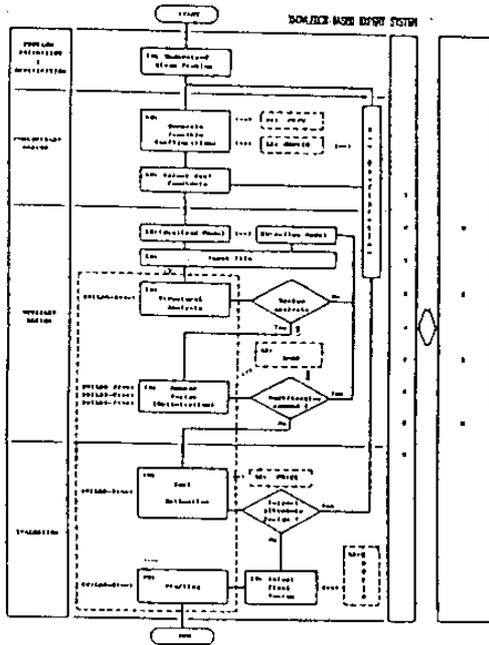


Figure 3 I-BUILDS Task Control

Problem Definition Module

The current development capability of I-BUILDS is delimited on the structural designer's problem so that the problem domain is reduced at the planning stage.

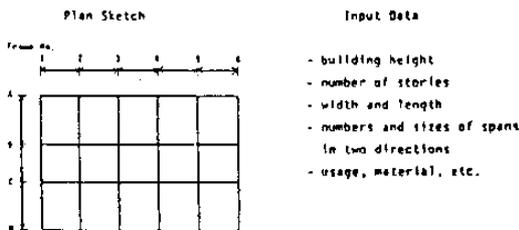


Figure 4 Initial Building Data

So some initial primitive data as shown in Figure 4 are requested for a space allocation in UNDERSTAND-GIVEN-PROBLEM KU from a user or a space planner, who characterizes preliminary building configuration such as height, width, length, number and size of span in two main directions, etc.

Preliminary Design Module

This module generates all the possible structural configurations from the given space allocation and makes preliminary selection of best candidate. Since the building is assumed as an assemblage of 2-D frames, the selection of 2-D framing system based on the previous design experience must be made first. To select the possible 2-D frames, the knowledge is extracted from the various sources such as recommendations of F. Khan and ACI Committee 442 [Ref.16], textbooks on the structural systems, Monograph on Planning and Design of Tall Buildings [Ref.13] and so on. These selected 2-D frames in two major axis directions are combined by GENERATE-FEASIBLE-CONFIGURATIONS KU referencing CONFIG KU to generate 3-D candidate configurations that satisfy design requirements. SELECT-BEST-CANDIDATE KU makes quantitative and qualitative comparisons of current candidate with other structural configurations, utilizing heuristic functions. Any configurations whose performance and acceptability fall well below those of others are abandoned and the others are saved to be retrieved in the later cycle. The information dealt in this process is shown in Figure 5.

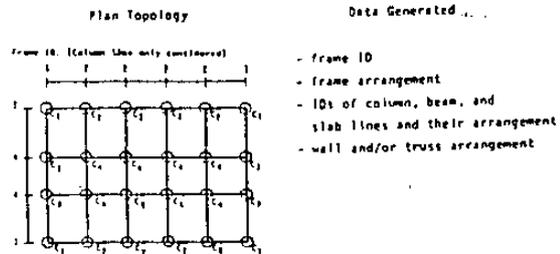


Figure 5 Generation of structural Configuration

Detailed Design Module

By this module, the properties of members of the selected candidate in the preliminary design process are picked up. This phase also involves the processes of preparation/checking of input data file, structural analysis/design, checking results and feedback control, etc.

Stage 1: For a chosen candidate configuration of building, it is required to introduce initial values of various design parameters such as beam depth and slab thickness. IDEALIZED-MODEL KU generates them from heu-

ristic rules (e.g. span/depth ratio) or some approximate analytic algorithms such as portal method. Then the input file is prepared automatically by INPUT-FILE KU for the later use in structural analysis. To complete the input file, job control data is given directly by user or by default values. Then the input file is checked in terms of FORTRAN format compatibility implemented in BUILDS and the completeness of each item. Figure 6 shows some information generated in this stage.

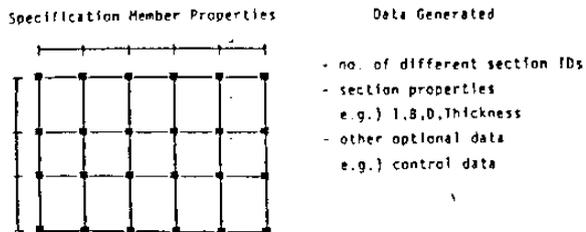


Figure 6 Specification of Member Properties

Stage 2: With the input file established in the stage 1, the structural analysis is performed by BUILDS-A and its responses are determined for the applied loads.

Stage 3: Based on the results obtained from the structural analysis, determination of member properties in accordance with the current regulations and codes of practice is performed and the improved members are updated (optimized). In the design process carried out by BUILDS-C/S/F, emphasis is placed on the optimization of individual members of the structural system.

Stage 4: With the individual members optimized, the acceptability of design is checked at this stage with constraints such as structural drift, overturning moment and member deflection/stress by CODE DB. If acceptable, the detailed design proceeds to the next stage. Otherwise, sensitivity vectors, which was obtained in the precedent stage, is retrieved to identify the most sensitive member(s) to correct the violated constraints. Then goes to stage 1 to update the input data with newly modified section properties using REFINE-MODEL KU. The modification of solution is continued until the optimum solution is found.

#### Evaluation module

The effectiveness of the selected configuration is evaluated in this stage. If it is appropriate, drafting is performed. If not, preliminary design module is taken up to modify the configuration and the process is reiterated.

Stage 1: Bill of materials and cost estimation of the selected candidate are made by BUILDS-B with PRICE DB.

Stage 2: At this stage, SUGGEST-ALTERNATE-DESIGN KU evaluates the design effectiveness based on the cost, structural performance, etc. and a decision has to be made whether to modify the current design both at the total system level and at sub-systems' levels. If modification is needed, it is performed in the Problem Redefinition Module.

Stage 3: If the current design is acceptable, drafting is performed by BUILDS-D. Before this design process, the results of given structural configurations are saved in a knowledge base named CONFIG and then efficiently used to retrieve the information with which the selection and the evaluation of alternatives are made. This makes it possible to nourish the knowledge bases of I-BUILDS.

#### Problem Redefinition Module

In modifying the building configuration, some information is needed which tells in which direction a move should be made. To decide the direction of next move, decision should be made by NEW-CONFIGURATION KU referencing CONFIG KB whether the selected candidate should be modified or an entirely new configuration should be introduced.

#### KNOWLEDGE CLASSIFICATION

Knowledge base is the kernel of expert system. In this paper, knowledge bases are first established with knowledge from printed texts, regulations and codes of practice. Then to get more information and/or qualify this knowledge, the experts' experiences and knowledge are solicited. This paper classifies the knowledge characteristics of the building structural system into three main categories as the following.

#### Physical Description of Building

The goal of building design is to determine the physical properties of a structural system that minimizes the cost required to resist all kinds of loads applied to the building and satisfies the building functional requirements such as architectural, mechanical and serviceability, etc. The simple model of the building structural system is to assume that it consists of mainly 2-D framing systems each of which resist both gravity and lateral loads and are interconnected by a framing system called slab which transfers the gravity loads (floor loads) to the 2-D frames. These 2-D sub-systems are physically composed of primitive structural components such as columns, beams, walls and trusses.

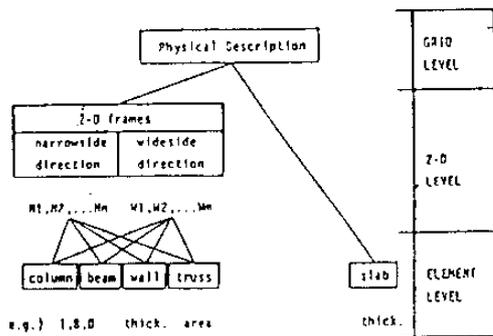


Figure 7 Physical Description of Building

The physical network of components which construct the structural system is presented in Figure 7.

#### Design environments

As shown in Figure 8, this knowledge is mainly related to the design constraints with which the properties of structural components are determined and the factors which affect the determination of overall structural systems. Regulations or codes present the loading conditions and constraints of member deformation/stress. The other functional subsystems including such as mechanical/service, electronic, construction and foundation play an important roles in determining the structural physical system. Owner's needs are also another constraints, among which are building material, usage, etc.

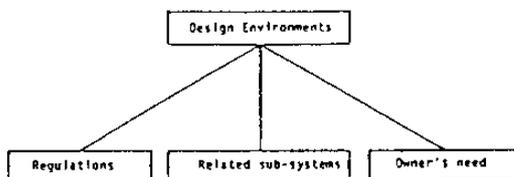


Figure 8 Design Environments

#### Menu of structural systems

I-BUILDS can select initial 2-D frames, which are needed at the preliminary design stage, from the data base called MENU. In the editing mode, a user could modify the previously selected 2-D frames at his disposal. This data base is constructed by storing the merits and faults of existing 2-D frames and their components in it. They can be retrieved later as needed. This kind of items is shown in Figure 9.

#### COMPUTER IMPLEMENTATION

The core of I-BUILDS shown in Figure 2 mainly consists of GEPSE INFERENCE, DBs, XBs and BUILDS subsystems. The executable programs under I-BUILDS are the execution

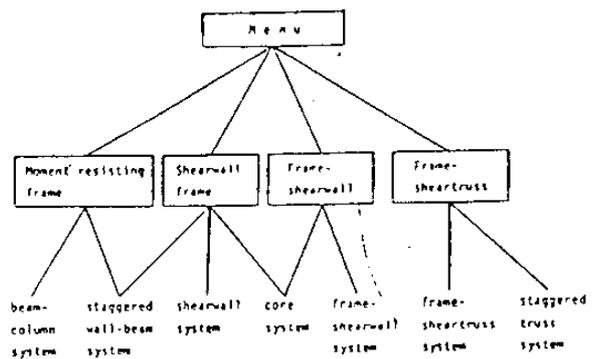


Figure 9 Menu of Structural Systems

files created by the meta-rules which edit, compile and /or load KBs, DBs and BUILDS as shown in the following example meta-rule.

```
IF (Input file is completed and Analysis is required)
THEN (Load input file and BUILDS-A, and Run BUILDS-A)
```

After the meta-rule like above example is applied, the information which is needed for next stage is extracted. It is required to transfer the structural displacement acquired from analysis to CODE KU in order to check if the results satisfy the code requirements. Because the programs or systems in I-BUILDS are written in different languages, this process requires mixed language programming. While the basic language used in GEPSE is C, KB is written in ONL (Object Network Language), DB in C, and BUILDS in FORTRAN. To achieve mixed language programming, the implemented knowledge and subsystems in I-BUILDS are linked with each other [Ref.17,18] so that it may be possible for the knowledge in C to control, compile and run BUILDS in FORTRAN, and communicate with each other systems.

The system solves the problems by problem reduction approach which uses meta-rule, sub-goal and sub-rule bases. Thus, the solution process is represented by a sequence of tasks which are defined in terms of sub-goals, that is, control is passed from task to task until the final design is achieved. The selection of sub-sets of knowledge requires the division of knowledge into knowledge packets, i.e., KUs, DBs, and XBs. Grouping knowledge according to its functions in a modular structures and allows parallel development of the various knowledge units. the knowledge base of I-BUILDS is constructed by five modules each of which represents the design stage as discussed in I-BUILDS ORGANIZATION.

GEPSE uses an associative triple, that is, OBJECT-ATTRIBUTE-VALUE to represent their static knowledge (facts). As an example, the syntax of assigning height ATTRIBUTE of building OBJECT a VALUE is as following statement, i.e., (set height building (read "How tall is it?")). And the syntax representing active knowledge (rule) is shown as a following example, which selects vertical 2-D frames.

```
(make-rule L-system1
  (and (about (get nst owner's_need) 40)
        (is? material owner's_need concrete)
        (is? usage owner's_need commercial)
        (known? gravity structure)
        (unknown? lateral structure)
  )
  (do (set lateral structure frame-shearwall)
      (explain lateral)
  )
)
```

This means,

```
IF   the story of the building is about 40
     and material of the building is concrete
     and usage of the building is commercial
     and the properties of the vertical system is known
     and those of the lateral system is unknown
THEN get frame-shearwall system for lateral system
     and explain the reason.
```

#### CONCLUSION

The conceptual model of constructing an Intelligent BUILDing Design System (I-BUILDS) by combining GEPSE and BUILDS is presented in this paper. The mixed language programming is utilized to link these two systems which were written in different languages. The knowledge modules and units of the system are under development in parallel. Although the development of I-BUILDS is still at its infant stage and requires a lot of efforts to be completed, it seems to have a good prospect of success.

#### REFERENCES

1. P. Harmon and D. King (1985), Artificial Intelligence in Business : Expert System, John Wiley and Sons, New York.
2. D. Waterman (1985), A Guide to Expert Systems, Addison Wesley Publishing Company.
3. S. Weiss and C. Kulikowski (1984), A Practical Guide to Designing Expert Systems, Rowman and Allanheld, Totowa, NJ.
4. M. F. Rooney and S. E. Smith (1983), Artificial Intelligence in Engineering Design, Computers and Structures, Vol.16, No. 1-4, pp.279-288.
5. D. Srinam, M. L. Maher and S. J. Fenves (1985), Knowledge-Based Expert Systems in Structural Design, Computers and Structures, Vol.20, No.1-3, pp.1-9.
6. W. J. Rasdorf and G. C. Salley (1985), Generative Engineering Database-Toward Expert Systems, Computers and structures, Vol.20, No.1-3, pp.11-15.
7. D. Srinam (1986), DESTINY:A Model for Integrated Structural Design, Artificial Intelligence, Vol.1, No.2, pp.109-116.
8. S. J. Fenves and J. H. Garrett (1986), Knowledge Based Standard Processing, Artificial Intelligence, Vol.1, No.1, pp.3-14.
9. M. L. Maher (1985), HI-RISE and Beyond: Directions for Expert Systems in Design, Computer Aided Design, Vol.17, No.9, pp.420-427.
10. C. K. Choi and E. D. Kim (1985), BUILDS-A : A Three Dimensional Analysis Program for Building Structures, Proceedings of the Second International Conference on Civil and Structural Engineering Computing: CIVIL COMPO 85, London, England.
11. C. K. Choi and H. W. Lee (1986), Three-Dimensional Analysis and Optimal Design of Building Structures, Proceedings of 10th Triennial CIB Congress, Washington D.C.
12. C. K. Choi and E. D. Kim (1985), Multistory Frames under Sequential Gravity Loads, Journal of Structural Engineering, ASCE, Vol.113, No.11, pp.2373-2384.
13. ASCE, Monograph on Planning and Design of Tall Buildings, Vol.PC(1981), Vol.SC(1980), Vol.CL(1980), Vol.SB(1979), Vol.CB(1978), New York.
14. W. Schueller (1977), High-rise Building Structures, John Wiley and Sons, New York.
15. F. Chehayeb and J. Connor (1985), GEPSE-A Computer Environment for Engineering Problem Solving, Research Report R86-11, Dept. of Civil Engineering, MIT.
16. ACI Committee 442 (1971), Response of Buildings to Lateral Forces, ACI Journal.
17. Microsoft Corporation (1987), Microsoft FORTRAN Optimizing Compiler for the MS-DOS Operating System, User's Guide.
18. Microsoft Corporation (1986), Microsoft C Compiler for the MS-DOS Operating System, User's Guide.

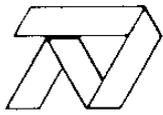


발표논문 3

논문제목 : " Input Data File Generation in I-BUILDS by An  
Expert System "

본 논문은 1988년 9월 한국전산구조공학회 창간호에 기재한 논문으로서  
I-BUILDS의 입력데이터 파일의 작성에 관한 것이다.

□ 발행처 : (주)대한민원 한국전산구조공학회 □ 주소 : / 130-4650 서울특별시 동대문구 박원대로 207-43  
 전화 960-2391 □ 이메일 : 012229-31-2836930 □ 발행인/최창근, 편집인/이광수, 인쇄인/박문식 □ 개간  
 □ 등록일자 1988년 8월 3일 □ 등록번호/자-1142 □ 인쇄 1988년 8월 25일 □ 발행 1988년 9월 1일



# 전산 구조공학

한국전산구조공학회지 제 1 권 제 1 호 (창간호)

## 목 차

창간사	.....	회장 최창근 (i)
격려사	.....	과학기술처장관 이 관 (ii)
<b>□ 창립총회 특집</b>		
취임사	.....	회장 최창근 3 ~ 4
한국전산구조공학회 창립배경 및 경과보고	.....	이동근 5 ~ 6
한국전산구조공학회 창립취지문 및 발기인명단	.....	발기인 이동 7 ~ 8
창립총회	.....	총무부 9 ~ 11
학회 Logo 해설	.....	12
사단법인 한국전산구조공학회 정관	.....	총무부 13 ~ 17
사단법인 한국전산구조공학회 사업계획	.....	총무부 18
<b>□ 학술기사</b>		
내진 설계시 고려할 사항 및 해석방법	.....	구민세 19 ~ 20
<b>□ 기술정보</b>		
컴퓨터 아키텍처의 새로운 바람, RISC란?	.....	삼성휴렛팩커드 21 ~ 22
Engineering Workstation이란?	.....	아프로컴퓨터 23 ~ 24
80년대 전산화의 추세와 구조설계용 CAE/CADD의 발전	.....	조효남 25 ~ 30
<b>□ 특별회원 소개</b>		
Apolo Computer - Korea	.....	섭외부 31 ~ 32
<b>□ 학회기사</b>		
.....	.....	편집부 33 ~ 47
<b>□ 안내</b>		
국제학술회의 안내	.....	편집부 48 ~ 50
해외주요학술지 목차	.....	편집부 51 ~ 63
전산기자재 공동구매 사업안내	.....	섭외부 63
학회행사안내	.....	학술부 64
<b>□ 논문</b>		
새로운 요소분해 방법에 의한 쉘 유한요소의 개발	.....	이재영 67 ~ 78
제 2 계 2 차모멘트/신뢰성조건 조합방법에 의한 부분안전계수	.....	이종현, 신현묵, 손승요 79 ~ 85
Application of Linearization Method for Large-Scale Structure Optimizations	.....	이희각 87 ~ 94
Input Data File Generation in I-BUILDS by an Expert System	.....	최창근, 김이두 95 ~ 102
기초의 Uplift를 고려한 건축구조물의 동적해석	.....	이동근, 송윤환 103 ~ 112
강도로교의 피로신뢰성 해석을 위한 실용적 모델	.....	신재철, 이승재, 장동일, 조효남 113 ~ 122
단순 및 연속 강도로교의 충격계수 변화에 관한 연구	.....	장동일, 이희현 123 ~ 133

# INPUT DATA FILE GENERATION IN I-BUILDS BY AN EXPERT SYSTEM

전문가 시스템에 의한 I-BUILDS의 입력데이터 파일 작성

최	창	근*
Choi,	Chang-	Koon
김	이	두**
Kim,	E-	Doo

### 요 약

I-BUILDS는 기존의 CAD 프로그램인 BUILDS와 지식베이스를 일관된 시스템으로 통합한 건물설계용 전문가 시스템이다. 본 시스템을 구축하기 위한 노력의 일환으로 본 논문에서는 추론에 의해 발생된 데이터들을 다양한 계산능력을 가지면서 특별한 업무도 수행할 수 있게 설계된 건물의 일괄구조설계 시스템인 BUILDS의 부시스템(BUILDS-A)에 이용되도록 입력파일화하기 위한 기본적인 개념을 기술한다.

### ABSTRACT

I-BUILDS is the building design expert system conceptually modeled for synthesizing the conventional CAD programs (BUILDS: an integrated BUILDing Design System) and the knowledge base into an unified system. As a part of the efforts to build this system, this paper presents the conceptual model by which the data items generated by inference are transformed into a formatted input file to be used for several BUILDS subsystems, each of which is assigned to have various computational capabilities and to perform specialized task.

### INTRODUCTION

The increased availability of expert system tools that simplify the construction of expert systems made it easier to build special purpose automating systems. These expert systems are the very appropriate tools for solving the ill-structured

problems that much rely on the domain expertise and experiences[Ref.1-4]. A lot of efforts to develop these systems have been made by the structural engineers to utilize accumulated heuristic knowledge obtained implicitly through long years of experiences in their computer aided design process.[Ref. 5-9] Because computers have been

\* 성희원, 회장, 한국과학기술원 교수, 공바  
\*\* 성희원, 울산대학교 전임강사

□ 이 논문에 대한 토론을 1988년 12월 31일까지 본학  
회에 보내주시면, 그 결과를 1989년 6월호에 게재하  
겠습니다.

extensively used for numerical computations these days, it is desirable that any emerging expert system would integrate existing algorithmic codes to construct a new automated design system.

When the knowledge-based expert system needs to be combined with the procedural code to maximize its capability and vice versa, the problem occurs that much information must be communicated among them. Doing this may contribute to data consistency and memory savings of the systems and avoid the time-consuming works otherwise the procedural programs must be translated into the language provided by its environments. Although hybrid tools provide this ability, most of the tools which run on microcomputers have such limited capabilities that they solve the problem by re-coding the procedural algorithms into LISP or PROLOG, or utilizing the tools based on the conventional languages such as C, FORTRAN or PASCAL etc. Most of the systems solve this problem by communicating each other through passing data values or sharing structured working memory with procedural code. The latter

is recommendable for its simplicity.

### I-BUILDS : AN INTELLIGENT BUILDS

The building design expert system, named I-BUILDS[Ref. 10], which is an acronym of an Intelligent BUILding Design System, is conceptually designed to provide users with consultations and manage BUILDS as the design phase proceeds. Figure 1 shows the architecture of this system, where the knowledge of design process, identified by Knowledge Modules(KMs) control BUILDS sub-systems, making use of the domain experiences implemented into Knowledge Bases and Data Bases.

Since I-BUILDS expert system handles BUILDS sub systems, the interface problem naturally occurs. In order to solve this problem, I-BUILDS communicates with BUILDS partly through sharing working memory and also partly through transmitting data by utilizing external files. In spite of input / output time consumed, sometimes the latter is particularly required.

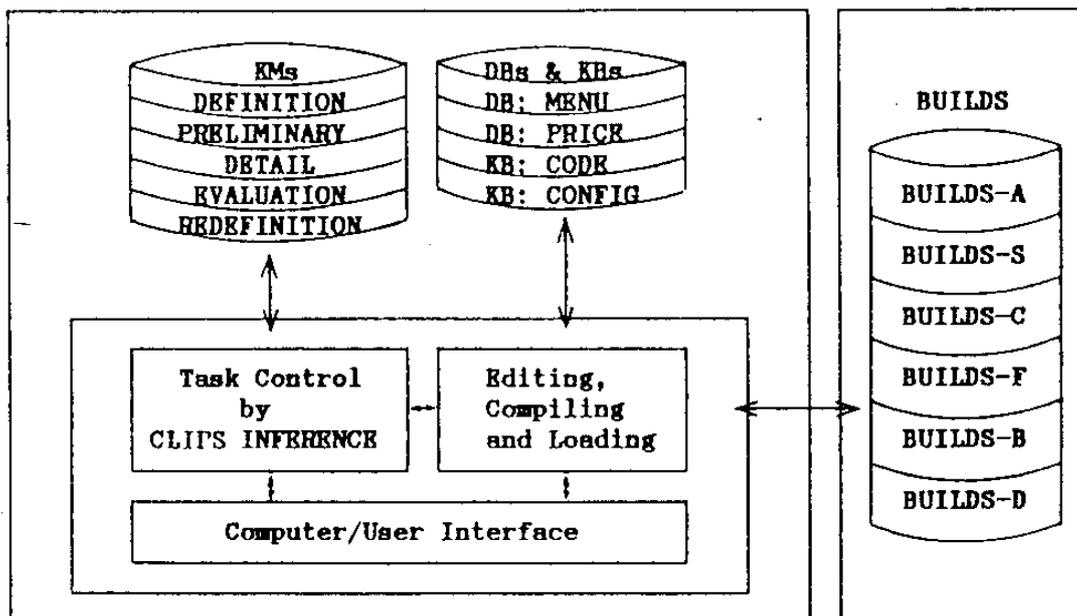


Figure 1 Architecture of I-BUILDS

### THE NEED TO DEVELOP AN INPUT DATA FILE GENERATOR

The commercially available structural analysis softwares which were sequentially programmed by systematic algorithms are widely spread over the worldwide engineering community. But with these codes, the fact that the more capabilities they perform the more knowledge the user must have to use effectively makes users occasionally aside from them. Due to the cost for purchasing commercial software, and its bulkiness of user's manuals, structural engineers can no longer be experts in using every special purpose package.

Therefore, a user of commercial packages may face with many difficulties in being skilled with the preparation of input data file for the program unless an expert system is utilized.

Making effective use of data structures provided by an expert system tool, the user can systematically handle all items of data. For an example, it is possible that the data about physical description of structural topology is represented into context network through using frame / object knowledge representation. When there are absence and incompleteness of required data, inference mechanism could verify or generate them utilizing experiences implemented. This paper presents an input data file generator model that may be

thought as a semi manual expert system which substitutes for text manuals for the conveniences of users. This model also can play a consultant's role in modeling the structure.

### INPUT DATA FILE GENERATOR MODEL

Tasks done before the structural analysis are sequentially shown in Figure 2. At the first stage, user prepares all the data such as structural configuration, loading condition and design constraints from design specifications, codes and standards, etc. Then, referencing user's manuals, a user synthesizes all the data needed until their completeness is confirmed and establishes input data file. Using this prepared data file a user tries to run and if any, pre-processor is utilized to check out errors. If runtime errors occur, user searches clues relative to this phenomena through the data file, referencing user's manual, and retries until he succeeds in running.

To improve the defaults described above which the conventional software accompanies, and for user's convenience, I-BUILDS affords users facilities for converting current data into input file, which is immediately applicable to BUILDS subsystems. The inference system automatically diagnoses errors utilizing the knowledge extracted from experiences and guidelines of the manual, and

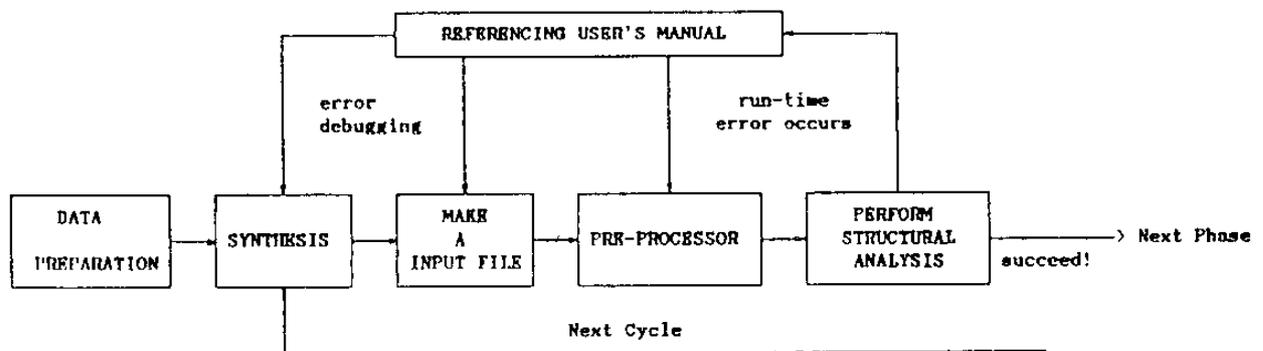


Figure 2 Conventional Pre-processor Model

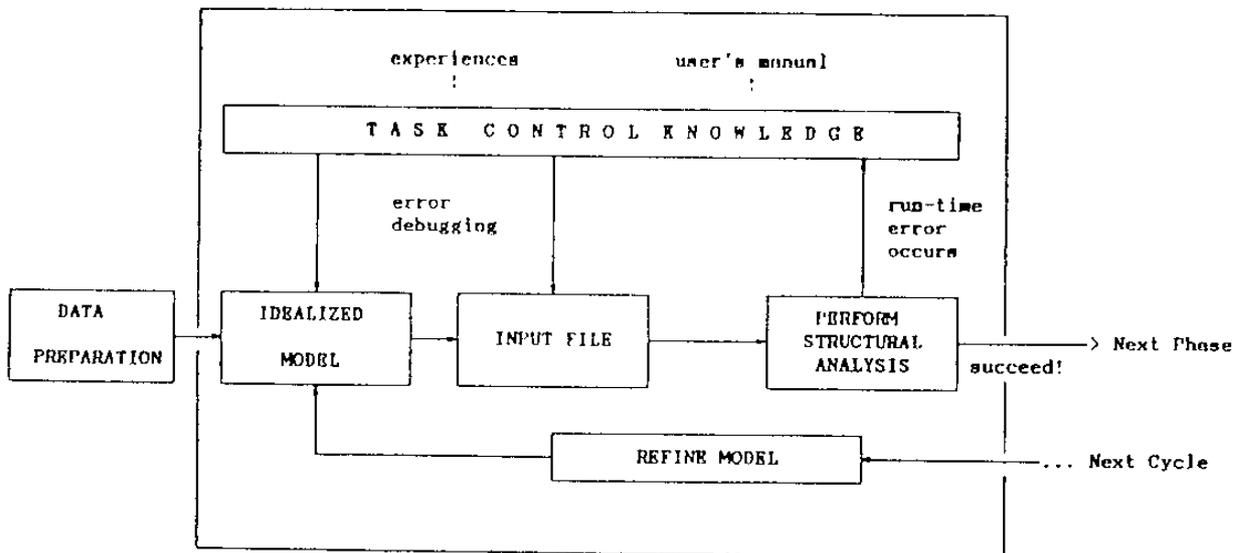


Figure 3 Input Data File Generator Model

gives prescription to users if errors occurred. In addition, this model can improve the member properties in the re-analysis cycle later, referencing the analysis results. This model is a part of building design model of I-BUILDS shown in Figure 4. Each knowledge units shown in Figure 3.functions as described below.

**IDEALIZED-MODEL:** For a chosen candidate configuration of structure, it is required to initialize various design parameters such as the sectional properties of the members. It generates initial values from heuristic rules (e.g., span/depth ratio) or some approximate analytic algorithms such as portal method. And all items of data are synthesized to make an input file.

**REFINE-MODEL:** This unit plays the same role as IDEALIZED-MODEL does in generating the required data except that it partly updates the input data with newly modified design parameters such as geometric properties.

**INPUT-FILE :** This Unit prepares the complete input file for the later use in structural analysis. To do this, job control data is directly read in or set by default values. Then the input file is

diagnosed with knowledge implemented into expert system in terms of FORTRAN format statements and their data structures.

#### BUILDS-A : A CONVENTIONAL STRUCTURAL ANALYSIS PROGRAM

BUILDS-A is one of sub-systems in BUILDS, which is an integrated building design system capable of performing entire processes of building design, i.e., from analysis to drafting.

**Capability :** BUILDS-A is a special purpose computer program for the linear three dimensional analysis of building structures for gravity, lateral and earthquake loads. The building structure is assumed as an assemblage of a series of rectangular frames, frame-shearwalls and frame-shear trusses interconnected through a rigid floor diaphragm. The analysis by BUILDS-A includes non-symmetric, non-rectangular building structures which have various kinds of frames located arbitrarily in the plan. The static loads may be combined with lateral earthquake input which is specified as a time dependent ground acceleration or as an

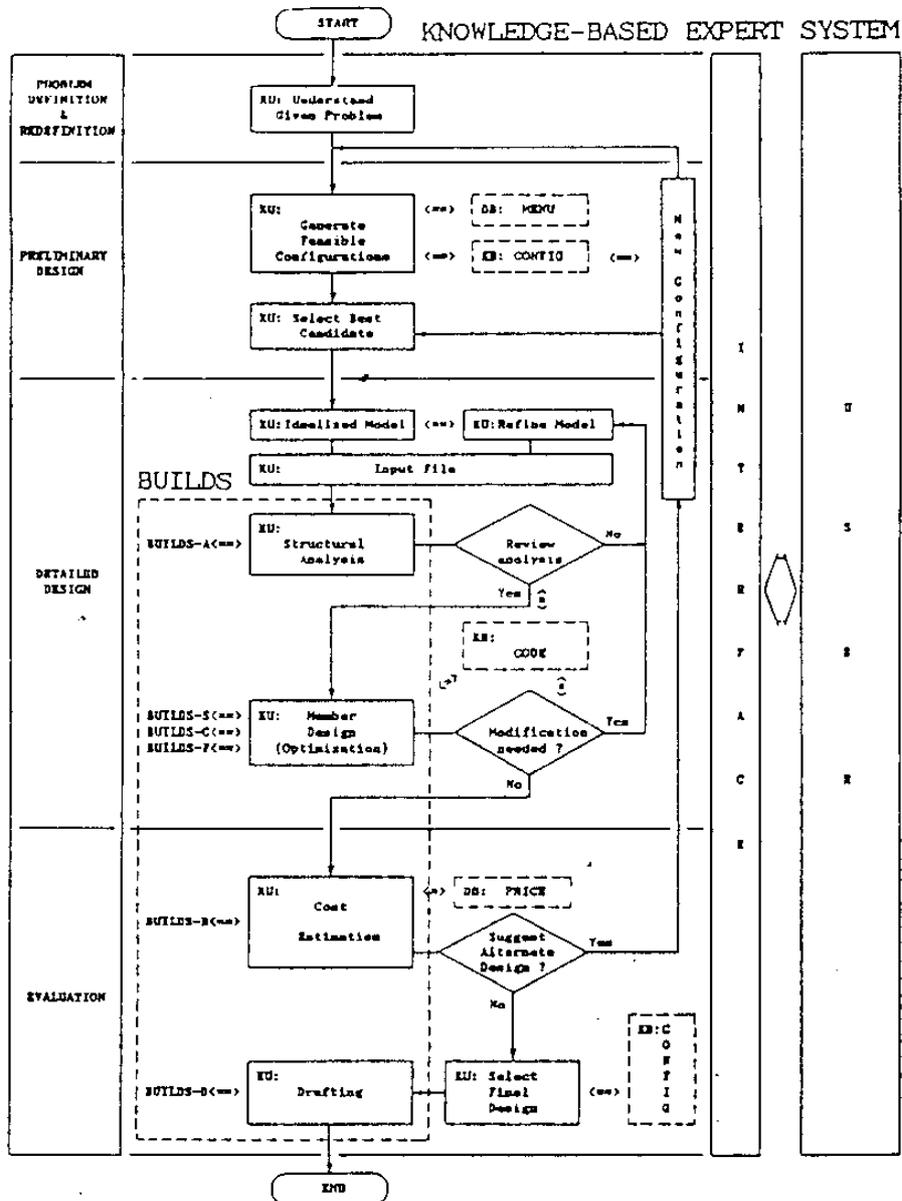


Figure 4 I-BUILDERS Task Control

acceleration spectrum response.

This program has two unique features, i.e. 1) the use of newly developed rectangular "plate element with cut-outs" for the analysis of perforated shear wall and 2) the structural analysis model for the gravity loads which take the construction sequence into account. In addition to these two features, the program also utilizes well-known characteristics of the building structures such

as the fact that building consists of mostly vertical and horizontal members of identical dimensions. Therefore, input data preparation can be minimized and a significant reduction in computational efforts can be resulted. This program also provides computer graphics for checking the input data and more efficient interpretation of the results.

Data Classification : Input data contents to BUILDERS-A are classified into following four

categories. Interested readers are referred to the references [Ref. 10-13] for the detailed informations.

- Execution Control Data
- Structural Configuration Data
- Element and Material Properties
- Applied Loads

Execution control data is mostly related to the task control knowledge implemented, therefore it can be input directly by a user or set by defaults. But the rest of data is dealt by the inference mechanism of expert system. To represent data items effectively, the hierarchical physical description is made as a context network.

#### DATA CONTROL STRATEGY

**Design Process :** Design is considered to be the iterative process of constructing artefacts which satisfy a number of constraints. The design process is usually iterated until the solution is optimized. The design process can be divided into the qualitative design, which includes the preliminary design and evaluation phase, and the quantitative design, which includes detailed design. While the detailed design which is called routine design mainly relies on so much numerical computation that it can be characterized by quantitative solution, the preliminary design which is called creative design deals with symbolic processing so that it is appropriate for qualitative non-numerical problems. The evaluation phase is also included in the latter domain because of its illstructured characteristics, which require much of designer's experiences.

The application of I-BUILDS is limited to the structural design problems, so that its domain is reduced to starting with planning stage. From given space allocation determined by space planners, I-BUILDS assumes the building structure as an

assemblage of 2-D frames. These selected 2-D frames in two directions combined to generated 3-D structural configuration and the best among the possible candidates is firstly chosen based on quantitative and qualitative comparisons using heuristic functions in I-BUILDS. Any configurations whose performance and acceptability fall well below those of others are abandoned and the other selected are saved to be retrieved in the later design. Then the properties of members of the selected candidate in the preliminary design process are determined by the iterative conventional technique. This detailed design phase involves the processes of preparation/checking of input file, structural analysis/design, reviewing the results and feedback control, etc. If this process is completed, the effectiveness of the current candidate is evaluated. If it is acceptable, it proceeds with the next process. But if not, preliminary design process is reiterated. This concept is represented in detail as shown in Figure 4.

**Hierarchical Approach :** The approach which many experts take in their design process is a hierarchical approach, in which determination of the overall concepts (e.g. building plan) precedes to the detailed determination (e.g. member properties). I-BUILDS also deals with abstract to detailed structural data as design proceeds as shown in Figure 5.

**Generation and Test :** During design process, I-BUILDS generates all possible alternatives and evaluates them to select the best candidate. As an example this approach is used in the process of selecting the best 2-D frames and 3-D configuration. It contributes to innovative design or creative design.

**Forward / Backward Changing :** While engineering design problem which makes artefacts from given data tends to be solved by a forward changing, a backward changing is more useful

DATA CLASS	INFORMATION FLOW	DESIGN PHASE	SOURCE
Execution	-	Task Control	read in
Structural Configuration	TOP	Planning	read in
	BUILDING SYSTEMS		
	3-D	Prelimi. Des.	heuristics
2-D			
Elements	-	Detailed Des.	heuristics & procedures (BUILDS)
Material	ELEMENT		

Figure 5 Data Management Strategy

for the issue in this paper, which has goals like diagnosis / prescription of making completeness of input data file. I-BUILDS utilizes these two inference strategy to make it possible that inference mechanism is selectively used in the special sub-problem domain, and it includes sub-goal, sub-rulebase and meta-rule to solve problems by using problem reduction[Ref.14].

DATA STRUCTURE OF INPUT FILE

Data structure adopted in input file usually consists of large blocks, each of them can be hierarchically divided into principal and auxiliary cards. Principal cards usually control the condition of subsequent input cards. The auxiliary cards are so length variable that it is difficult to deal with fixed-size memory such as an array. Since the variable data structures can not be predicated before run time, the use of pointers rather than arrays to achieve an efficient and reliable implementation is desirable. Utilizing their dynamic nature, it is possible for nodes to be added to or removed from a list at any time. Then ordering

sequence of needed cards can be made through handling the nodes of linked list[Ref.15, 16]. I-BUILDS inference mechanism, which utilizes this characteristics of knowledge representation, converts the fact existing in the working memory into a formatted input file in accordance with requirements of BUILDS-A. Quite a lot of rules, of course, check the compatibility of all the data items before this function is used.

CONCLUSIONS

Integrating all the design phases of building structures with conventional procedural language is very difficult work because of having to deal with large amounts of data items. To overcome this difficulty, I-BUILDS combines algorithmic BUILDS and knowledge based expert system. The inference mechanism can be used to make abstract decision, e.g., preparation of the input file for BUILDS in FORTRAN, and current procedural software can be also embedded into the expert system to perform most of the numerical computation without any conversion. In this paper the prototype

of input data file generator for the procedural code is presented. Although this work is in the infant stage of development, this system will make the access to the conventional software easier and more effective.

REFERENCES

1. P. Harmon and D. King(1985), Artificial Intelligence in Business :Expert System, John Wiley and Sons, New York.
2. D. Waterman (1985), A Guide to Expert Systems, Addison Wesley Publishing Company.
3. S. Weiss and C. Kulikowski (1984), A Practical Guide to Designing Expert Systems, Rowman and Allanheld, Totowa, NJ.
4. M.F. Rooney and S.E. Smith(1983), Artificial Intelligence in Engineering Design, Computers and Structures, Vol.16, No. 14, pp.279-288.
5. D. Sriram, M. L. Maher and S. J. Fenves (1985), Knowledge-Based Expert Systems in Structural Design, Computers and Structures, Vol.20, No.1-3, pp.1-9.
6. W. J. Rasdorf and G. C. Salley (1985), Generative Engineering Database-Toward Expert Systems, Computers and structures, Vol. 20, No.1-3, pp.11-15.
7. D. Sriram (1986), DESTINY: A Model for Integrated Structural Design, Artificial Intelligence, Vol.1, No.2, pp.109-116.
8. S.J. Fenves and J.H. Garrett (1986), Knowledge Based Standard Processing, Artificial Intelligence, Vol.1, No.1, pp.3-14.
9. M.L. Maher (1985), HI-RISE and Beyond: Directions for Expert Systems in Design, Computer Aided Design, Vol.17, No.9, pp. 420-427.
10. C.K. Choi and E.D. Kim (1987), A preliminary Model of I-BUILDS :An Intelligent Building Design System, Proceedings of Second International Conference on Application of Artificial Intelligence in Engineering, Knowledge Based Expert System in Engineering: Planning and Design, Edited by D. Sriram and R.A. Adey, Computational Mechanics Publications, Boston, USA.
11. C.K. Choi and E.D. Kim (1985), BUILDS-A : A Three Dimensional Analysis Program for Building Structures, Proceedings of the Second International Conference on Civil and Structural Engineering Computing: CIVIL COMP 85, London, England.
12. E.D. Kim (1986), Linear 3-D Analysis of Building Structures, Thesis of M.S. Degree, Department of Civil Engineering, KAIST, Seoul Korea.
13. C.K. Choi and H. W. Lee (1986), Three-Dimensional Analysis and Optimal Design of Building Structures, Proceedings of 10th Triennial CIB Congress, Washington D.C.
14. F. Chehayeb and J. Connor (1985), GEPSE-A Computer Environment for Engineering Problem Solving, Research Report R86-11, Dept. of Civil Engineering, Problem Solving, Research Report R86-11, Dept. of Civil Engineering, MIT.
15. A. Hansen (1987), Proficient C for the Microsoft C (verison 4), Microsoft Press, Washington.
16. G.E. Sobelman and D.E. Krekelberg (1985), Advanced C: Techniques and Applications, Que Corporation, Indianapolis, Indiana.

(1988년 6월 10일 접수)

**발표논문 4**

**논문제목 : " 전문가시스템도구인 K-CLIPS에서의 지식표현 "**

본 논문은 1989년 4월 한국전산구조공학회 학술대회에 발표되었던 것으로 I-BUILDS에 사용된 K-CLIPS의 지식표현에 관한 것이다.

한국전산구조공학회

# 학술 발표회 논문집

Proceedings of COSEIK Symposium - Spring 1989

강철도교의 정-동적 응답에 관한 연구 ..... (61)

A Study on Static and Dynamic Responses of  
Steel Railway Bridges

장 동 일      최 강 희      이 희 현

구조물의 동특성 추정방법에 관한 연구 ..... (66)

Identification of Structural Dynamic Systems

윤 정 방      Shinozuka, Masanobu

전문가시스템도구인 K-CLIPS에서의 지식표현 ..... (71)

Representation of Knowledge in K-CLIPS

-Expert System Development Tool

최 창 근      백 종 균

## 전문가시스템도구인 K-CLIPS에서의 지식표현

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

최 장 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, 팜플렛사, 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 일부만을 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 유효요소해석 프로그램과 컴퓨터그래피 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.

\* 정회원 한국과학기술원 토목공학과 교수  
\*\* 정회원 한국과학기술원 토목공학과 박사과정

이러한 지식을 표현하기 위하여 대부분의 규칙기반 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 조직하기 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 판제자로서 객체와 값, 객체와 객체간, 그리고 객체와 함수간의 관계를 묘사하는 역할을 한다.

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

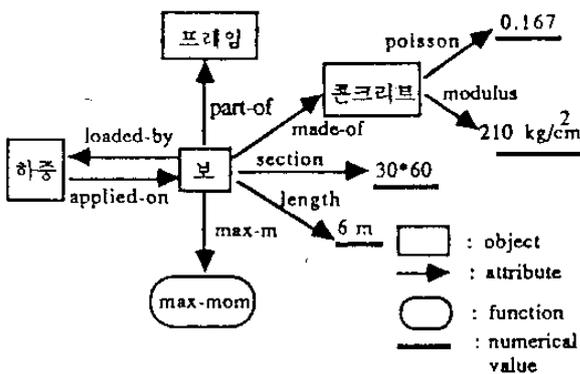


그림 1 O-A-V 양식에 의한 보 지식의 표현의 예

## 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 알고리즘 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 선정할 후, 이중에서 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 입시목표로 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 사용자 하여금 지식베이스를 조작하도록 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 있으며 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

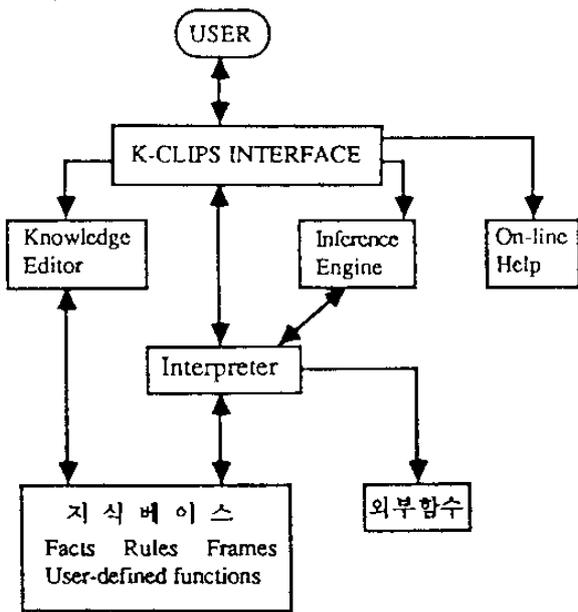


그림 2 K-CLIPS의 구조

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-execute와 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 이루어 진

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 플라스라고 부른다. 그림 4와 같은 계층적인 지식은 그림 5와 같이 표현된다.

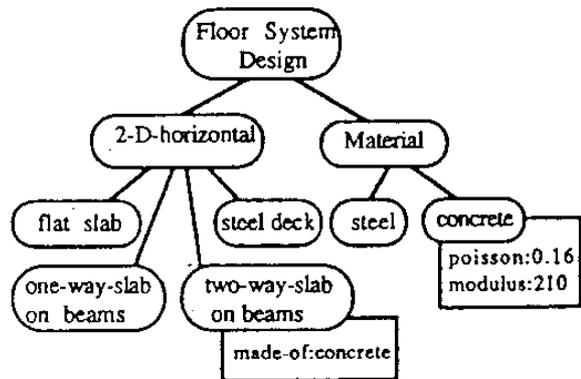


그림 4 계층적인 지식구조

```

(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 데, 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. 결 론

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 데이터틀 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.Chelayeb, 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.



발표논문 5

논문제목 : " 인공지능을 이용한 교량의 예비설계용 전문가시스템의 개발 "

본 논문은 1989년 9월 한국전산구조공학회 학술대회에 발표되었던 것으로 교량의 예비설계용 전문가시스템에 관한 것이다.

한국전산구조공학회

# 학술발표회 논문집

Proceedings of COSEIK Symposium - Fall 1989

## 목 차

- 구조설계의 CAD화를 위한 설계지향 유한요소 프로그램의 개발에 관한연구 ..... ( 1 )  
Development of Design Oriented Finite Element Program for CAD Integration  
이성우
- 인공지능을 이용한 교량의 예비설계용 전문가시스템의 개발 ..... ( 7 )  
Development of Expert System for Preliminary Bridge Design with Artificial  
Intelligence  
최창근, 최인혁
- 콘크리트 포장구조의 평면응력 해석을 위한 컴퓨터 모델 ..... ( 15 )  
A Computer Model for the Planar Effects of Concrete Pavements with Skewed  
Joints  
조병환
- 원자력 차폐 구조물의 설계하중 조합 기준 ..... ( 19 )  
Load Combination Criteria for Design of NPP Conintainment Structures  
한봉구, 조효남
- 사무실 적재하중의 통계적 조사 연구 ..... ( 25 )  
Study on Live Loads in Office Buildings  
김상효, 조형근, 배규웅, 박홍석
- 잔류응력하의 강구조물의 피로신뢰도 ..... ( 29 )  
Fatigue Reliability of Steel Structures in the Presence of Residual Stresses  
조효남, 김두한, 허상구
- 평면뼈대구조의 신뢰성해석에 관한연구 ..... ( 34 )  
A study on Reliability Analysis for Plane Frame Structure  
이중빈, 신형우
- 평면 FRAME 구조물의 확률 유한요소 해석 ..... ( 40 )  
Probabilistic Finite Element Analysis of Plane Frame  
양영순, 김지호

# 인공지능을 이용한 교량의 예비설계용 전문가시스템의 개발

Development of Expert System for Preliminary Bridge Design with Artificial Intelligence

최창근\*      최인혁\*\*  
Choi, Chang-Koon      Choi, In-Hyuk

## ABSTRACT

This paper presents an expert system approach to solve preliminary bridge design problems. The system employs a forward chaining inference strategy to 1) choose the appropriate superstructure types and construction methods and 2) use the solutions chosen in 1) to determine a list of ranked alternatives.

The basic information used in the selection is collected from various sources. Due to the uncertainties presented in the information collected, Fuzzy sets are used to handle these uncertainties in the system.

Finally to approve this system some applications are made to select superstructure types and construction methods of them.

## 1. 서론

일반적으로 토목공학에 있어서의 구조해석이나 제작작업의 면에서 컴퓨터의 이용은 거의 확립되었다고 말해도 과언은 아니다. 현재에는 단순한 기술적인 측면 뿐만 아니라 보다 넓은 범위에 있어서 컴퓨터를 이용하려는 노력이 행해지고 있다. 1960년대 후반에 미국에 있어서 그의 개념이 생겨난 CAD(Computer Aided Design) System을 포함한 자동설계의 보급은 그 현상의 하나이고, 현재 토목공학의 각 분야에서 설계지원 시스템으로서 이용되고 있다.

그러나 보다 상위의 단계인 예비설계에 있어서 적절한 설계를 위해서는 교종, 경간, 교각의 위치등은 법규, 지형조건, 환경조건, 경제조건등의 여러조건을 주시해 가면서 설계자의 상당한 숙련 및 경험이 필요하고 또한 고도의 지식도 함께 요구된다. 그렇지만 오늘날과 같이 기술이 급속하게 발달해 전문성이 분화되어 1인의 토목기술자가 여러분야에 충분한 경험을 쌓아 그것을 기초로 하여 교량등의 특정구조물의 형식선정을 수행하는등 보다 상위의 단계에 있어서의 컴퓨터의 이용은 현재로는 실행되고 있지않다. 이같은 문제를 해결하는 수단으로서 인공지능(Artificial Intelligence)분야 중에서 특히 최근에 주목을 받고 있는 전문가 시스템(Expert System)이 있다.

전문가 시스템이란 해결해야할 문제에 속하는 영역의 전문지식(전문적 사실과 경험적 지식)을 데이터베이스(Data Base)화 하여 비전문가가 이것을 이용해서 매우 충분한 문제물 전문가와 동등한 정도의 수준으로 해결하는 것을 목표로 하는 시스템이다. 최근에 비교적 성공적인 결실을 거둔 전문가 시스템을 보면 PROSPECTOR 및 컴퓨터의 설치계획을 위한 R1등이 있으며, 구조공학 분야에서도 수치계산을 주로 행하는 기존의 컴퓨터 이용방법을 기호처리(symbolic processing)에 바탕을 둔 새로운 기법을 도입하여 컴퓨터의 이용효율을 향상시키려는 노력이 이루어지고

있다. 이에 대한 전문가 시스템으로는 중층 건물의 개략설계 지원을 위한 HI-RISE, 구조해석법의 선정을 위한 SACON, 내진손상도 해석을 위한 SAGE 및 구조안정성 평가를 위한 HOWSAFE, DAPS등이 있다.

본 논문에서는 이 전문가시스템을 이용하여 교량의 예비설계 부분인 상부구조의 형식 및 가설공법을 선정하고, 설계자 및 법규등에서 얻은 지식에 대한 불확실성(uncertainty)을 해결하기 위하여 Fuzzy 이론을 도입하였다.

## 2. 시스템의 배경

### 2.1 전문가 시스템

전문가 시스템은 앞에서 설명된 바와 같이 전문적인 지식을 지식베이스(knowledge base)로서 저장하고 그 저장된 것을 이용하여 문제해결을 하는 시스템이다. 여기서 전문적인 지식은 사실(fact)에 관한 지식과 전문가의 경험으로부터 얻은 지식(heuristic knowledge)을 가리킨다. 일반적으로 전문가 시스템의 구축에 있어서 문제가 되는 것은 지식의 획득, 지식의 표현, 지식의 이용 및 사용자와 기계와의 인터페이스(interface)의 4가지를 들 수 있다. 그중에서 가장 중요한 문제는 지식의 획득이다. 지식중에서 전문가의 경험적 지식을 얻어내는 일은 매우 어렵고 그와 같은 지식은 아무리 전문가라 하더라도 표현상의 어려움이 있기 때문이다. 그러므로 이 경험적 지식을 어떻게 알기쉽게 체계화하여 정립할 것인가 하는 문제가 지식베이스 구축의 성공여부를 결정한다.

전문가 시스템의 대표적인 표현으로서 production system이 있는데 이 시스템에는 IF(조건부), THEN(실행부)의 형태를 가지는 production rule을 모은 지식베이스가 있다. 이 시스템의 특징은 지식을 이해하기 쉽고 정의, 변경 및 확장이 용이하다는 것이다.

본 논문에서는 production system 방식에 적합한 전문가 시스템 개발용 도구(tool)인 K-CLIPS(KAIST-C Language Integrated Production System)를 이용하여 본 시스템을 구축하였다.

\* 정회원 한국과학기술원 교수  
\*\* 정회원 한국과학기술원 박사과정

## 2.2 시스템 구축용 도구

본 시스템에서는 전문가 시스템의 구축을 위하여 기존의 도구인 K-CLIPS를 이용함으로써 우선 지식베이스를 구축하는 노력을 감소할 수 있고, 또한 제한된 시간내에 획득, 입력되는 지식의 양을 쉽게 확장할 수 있다. K-CLIPS는 NASA에서 개발한 CLIPS를 바탕으로 하여 본원 토목공학과에서 그 기능을 확장한 전진추론(forward chaining)방식의 전문가 시스템 개발용 도구이다. 이 도구의 주요 구조는 그림 1에서 보는 바와 같이 지식베이스, 지능형 에디터(knowledge editor), 추론기관(inference engine), 인터프리터(interpreter) 및 사용자 인터페이스(user interface)로 되어 있다.

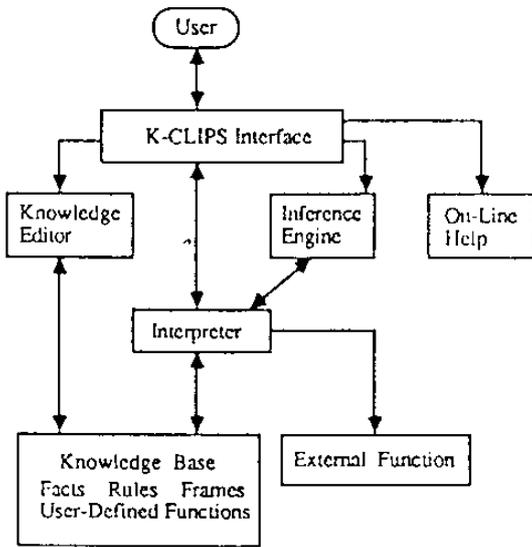


Fig.1 The Structure of K-CLIPS

## 3. 시스템의 구축 및 개요

### 3.1 시스템의 선정과정

교량계획에 있어서 기본적으로 고려되어야 할 사항은 가교위치와 노선선명에 있다. 모든 교량은 노선의 선명에 준하여 설계가 이루어져야 하며, 일단 가교위치가 정해지면 교장, 지간, 교대, 교각의 위치 및 방향, 형상과 깊이등이 문제가 된다. 이것들은 현장의 지형, 지질상태 이외에도 교차하는 하천과 도로등에 관리자의 의견이 중요한 요소가 되므로 기본조사를 충분히 하고 관리자와의 충분한 협의를 거쳐야 한다.

다음 단계로 하부구조, 상부구조 및 가설공법의 선정이 있는데 여기서 더욱 중요한 요인으로 되는 것이 경제성의 문제이다. 가령 지간분할을 계획하면 기초의 수가 적게 끝나 상부구조의 중량이 크게 되어 기초를 견고하게 하여야 하고, 반면에 지간분할을 많게 하면 기초의 수가 많게 끝나 상부구조의 중량이 작게 되어 기초는 안정하게 된다. 이와 같은 경제성의 문제는 하부, 상부구조의 일체로서 다루어지도록 하여 지간분할, 기초형식을 결정하지 않으면 안된다. 또한 경제성 이외에도 시공성, 유지관리성, 주행의 안정성 및 경관성에 대해서도 충분한 고려를 하여야 한다.

이상과 같은 배경 및 관점에서 본 논문에서는 교량의 예비설계를 대상으로 각 전문가의 지식, 경험을 토대로 한 부시스템(sub system)을 설정하고 또한 그것들을 합성한 전체 시스템(total system)의 구축을 목표로 하는 것이다. 물론 각 부시스템간에는 동일 지표로 알 수 없는 것, 결정론적으로 설정할 수 없는 것등의 문제점이 있다. 이상과 같은 사항을 고려한 예비설계의 순서는 그림 2와 같다.

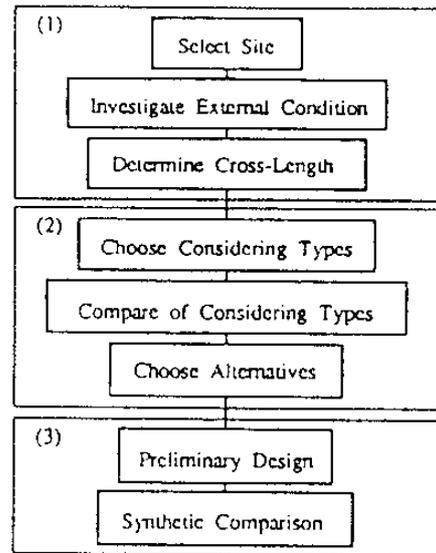


Fig.2 A Flow of Preliminary Bridge Design

그림 2에서 (1) 교량 가설지점의 지형, 지질, 인접노선계획의 상황, 하천의 개수계획 및 하천의 관리조건등을 토대로 교장을 결정하는 작업이다. (2) 교종, 지간분할, 구조등에 대해서 충분한 검토를 행하여, 적당하다고 생각되는 교종에서 하부구조, 상부구조 및 가설공법에 대해서 2-3 종류를 선정하는 작업이다. (3) (2)에서 선정된 종류에 대해서 각각설계를 행하여 경제성, 구조특성에 대해서 종합비교를 행한다.

본 논문에서는 이같은 예비설계 업무내에서 (2)에서의 사용을 전체로 한 prototype의 시스템 구축을 한다. 본 시스템에서 고려하고 있는 상부구조의 형식 및 가설공법은 표 1, 표 2와 같고, 이에 대한 지식베이스의 구축은 시방서, 실시설계 보고서, 교량전문가의 경험을 토대로한 의견을 참고로 하여 지식을 체계화하였다. 지식의 표현형식은 IF-THEN 형식의 production rule로 하였고, 지식의 예애한 정도를 표현하는 척도로서 확신도를 사용하였는데, 본 연구에서는 Fuzzy이론으로부터 이에 대한 해결을 했다. 이에 대한 본 시스템의 선정순서는 그림 3과 같다.

### 3.2 상부구조형식 및 가설공법 선정시스템

본 시스템에서 고려된 상부구조의 형식은 표 1에서 나타난 바와 같이 강교와 콘크리트교로 분리할 수 있다. 상부구조의 형식을 선정시 고려할 조건으로는 아주 많이 있지만 본 시스템에서는 그중에서 중요하다고 생각되는 bridge geometry, main span length, girder depth, truss height, arch rise, bridge site, bridge elevation, ease of maintenance, ease of

Table 1 Lists of Superstructures

Steel superstructures	Concrete superstructures
Simple plate girder bridge	Precast slab bridge
Conti. plate girder bridge	Simple p.c. girder bridge
Simple box girder bridge	Conti. p.c. girder bridge
Conti. box girder bridge	Simple p.c. box girder bridge
Orthotropic deck bridge	Conti. p.c. box girder bridge
Simple truss bridge	Segmental bridge
Conti. truss bridge	Arch bridge
Langer bridge	
- upper deck	
- through deck	
Lohse bridge	
- upper deck	
- half-through deck	
- through deck	
Arch bridge	

Table 2 Lists of Construction Methods

Steel superstructures	Concrete superstructures
Staging method	Cast-in-place method
- with truck crane	Precast method
- with cable crane	- with truck crane
- with stiffleg derrick	- with floating crane
- with floating crane	- with erection truss
Cable crane erection	- with gantry crane
Cable erection method	Movable scaffolding method
Erection truss method	Incremental launching method
Pushing out method	Free cantilever method
- with erection truss	- cast-in-place method
- with erection nose	- precast method
- with barge	Progressive placing method
- with movable bent	
Cantilever method	
- with truck crane	
- with cable crane	
- with stiffleg derrick	
- with floating crane	
Large block method	
- with truck crane	
- with floating crane	
- with barge	

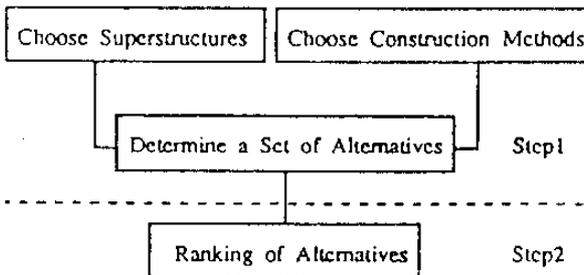


Fig.3 A Flow of This System

fabrication 및 appearance등을 들 수있다. bridge system은 단순교와 연속교로 분리할 수있고, bridge

geometry는 직선교와 곡선교로 분리할 수있고, main span length는 각 교종에 따라 각각 이상적인 지간이 있는데 실제로는 하부구조와의 관계, 교장등의 제한 때문에 생각한 것처럼 쉽게 정해지지 않는다. 그래서 본 시스템에서는 문헌 및 전문가의 의견을 참고로 각 교종별로 적용지간의 범위를 정했다. 또한 bridge site는 하천부, 산간부 및 기타지역으로 분류하고, maintenance는 콘크리트교에서는 보통 도장은 불필요하므로 유지비용은 적은 것으로 고려하여 강교보다는 좋다고 생각되고, 강교는 도장이 정기적으로 행해져야 하므로 유지비용이 많이 드는 것으로 고려한다. 또한 fabrication에 있어서는 안전성, 난이도, 공기, 편리성 및 가설비용이 환경요소로 들 수있을 것이다. 마지막으로 appearance는 선정요소를 고정화 하는 것은 어렵기 때문에 교종과 가설장소로부터 판정한다.

다음으로 가설공법에 대해서는 표 2에 나타난 바와 같이 강교와 콘크리트교에 대해서 각각 가설공법이 분류되었다. 현장의 여러조건을 고려한 가설공법의 선정에 대한 흐름도는 그림 4와 같다. 이상과 같은 과정으로 상부구조의 형식과 가설공법이 얻어지면 본 시스템에서는 표 3으로부터 각 상부구조의 형식에 적합한 가설공법들이 몇가지 나오게 된다. 이렇게 얻어진 비교안들은 공학적인 판단에 의하여 A, B, C, D, E의 5단계로 등급을 부여한다. 가령 상부구조에

대해서는 표 4에서와 같이 main span length, girder depth, appearance 및 maintenance등에 의하여 부여되며, 또한 가설공법에 대해서는 표 5에서와 같이 site geometry, site environment 및 superstructure type등에 의해 등급이 부여된다. 이러한 공학적 판단에 의해 평가된 비교안들은 Fuzzy 이론에 의하여 최종적으로 상부구조의 형식 및 가설공법이 선정되게 된다.

Table 4 Rating Table of Superstructures (in Case of Concrete Superstructures)

Sup-struct. type	Concrete Superstructures						
	p.c. slab	p.c. girder		p.c. box girder			arch
		simp	cont	simp	cont	seg.	
Judgement factor							
Main span length:							
20 - 50 m	A	A	A	A	A	D	D
51 - 75 m	D	D	C	C	B	A	B
101 - 150 m	D	D	D	D	D	A	A
Girder depth:							
1.6 - 2.5 m	B	A	B	B	A	C	E
2.6 - 3.5 m	C	B	B	D	C	B	E
Height of arch, truss:							
11 - 15 m	E	E	E	E	E	E	B
16 - 20 m	E	E	E	E	E	E	A
Ease of fabrication	A	A	A	A	A	B	B
Ease of maintenance	A	A	A	A	A	A	A
Appearance:							
river area	E	A	A	A	A	A	E
valley or the like	E	E	E	E	B	B	A
metropolitan area	E	B	B	B	B	B	E

where A: very good, B: good, C: fair, D: poor, E: unknown  
simp: simple span, cont: continuous span  
seg: segmental bridge

Table 3 Rating Table of Superstructures and Construction Methods (for Steel Superstructures)

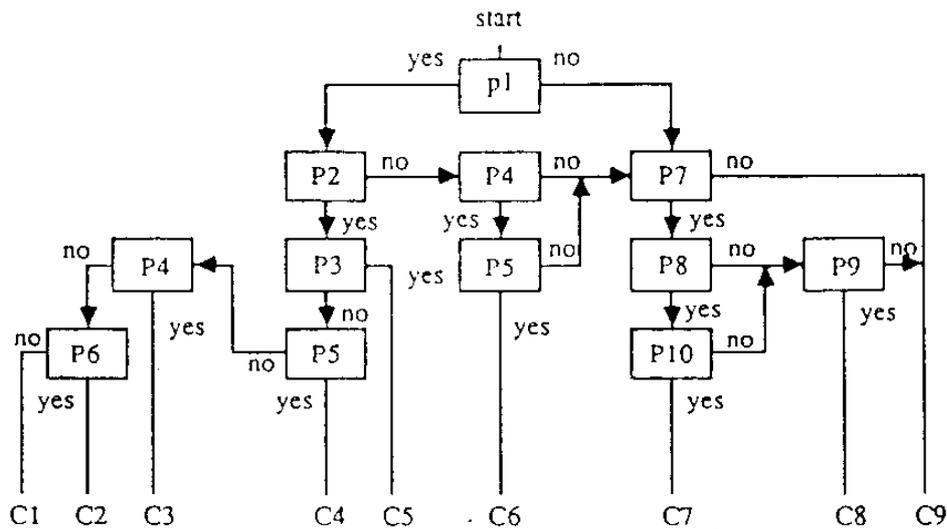
Superstructure type Construction method	Steel Superstructures												
	girder		box girder		orth deck	ruess		langer		lohse			larch
	simp	cont	simp	cont		simp	cont	up.d	th.d	up.d	ht.d	th.d	
Staging method - with cable crane	B	B	B	B	B	B	C	E	B	E	E	B	B
- with stiffleg derrick	E	B	C	B	B	B	B	E	E	E	E	E	E
- with floating crane	E	E	B	B	B	B	B	E	E	E	E	E	B
Cable crane method	C	C	E	E	E	A	C	A	A	E	E	A	A
Cable erection method	E	E	E	E	E	E	E	A	A	A	A	C	E
Erection truss method	B	B	B	B	B	E	E	E	E	E	E	E	E
Pushing out method - with erection truss	E	B	E	B	B	B	E	E	B	E	E	C	D
- with erection nose	A	A	A	A	A	E	E	E	E	E	E	E	E
- with movable bent	B	E	E	E	E	E	E	E	B	E	E	B	B
Can'tilever method - with cable crane	E	B	E	B	B	E	B	E	E	E	E	E	E
- with floating crane	E	B	E	B	B	E	B	E	E	E	E	E	E
Large block method - with truck crane	A	C	A	C	E	E	E	E	E	E	E	E	E
- with floating crane	E	E	B	B	B	B	B	E	C	E	B	C	C

where A: very good, B: good, C: fair, D: poor, E: not consider  
 simp: simple span, cont: continuous span, orth deck: orthotropic deck bridge  
 up.d: upper deck, th.d: through deck, ht.d: half through deck

Table 5 Rating Table of Construction Methods (in Case of Steel Construction Methods)

Construction method Judgement factor	For Steel Superstructures									
	staging method		cable crane method	erection truss method	pushing out method		can'tilever method		large block method	
	with t.c	with c.c			with e.n	with m.b	with c.c	with f.c	with t.c	with f.c
Superstructure geometry: curved bridge	A	B	D	A	D	D	D	D	C	D
high-level bridge (h > 29 m)	D	B	A	A	A	A	A	A	C	A
Possible work in narrow working area	D	B	C	A	A	D	A	C	C	C
Require rapid construction	A	C	C	C	B	A	C	B	A	A
Multi-span construction	A	C	C	B	A	C	B	B	A	B
Minimal disturbance to the existing traffic	D	C	A	A	A	C	B	B	A	B
Economic	B	B	C	B	B	C	B	B	B	B
Construction in valley area	D	C	A	B	B	D	A	E	D	E
Construction on soft ground	D	C	A	A	A	B	A	E	D	E
Ease of construction	A	C	D	C	C	C	C	B	A	B

where A: very good, B: good, C: fair, D: poor, E: not consider  
 t.c: truck crane, f.c: floating crane, c.c: cable crane  
 e.n: erection nose, m.b: movable bent



- P1 -> The working area under the bridge is available.
  - P2 -> It is possible to place temporary bent for erection.
  - P3 -> The truck crane or trailer or the like can be entered into the working area.
  - P4 -> There is a yard for fabrication near the site.
  - P5 -> The floating crane or barge can be entered into the working area.
  - P6 -> A cable crane equipment can be installed for erection.
  - P7 -> There is a working space at the one or both end of the bridge.
  - P8 -> Some part of some span can place temporary bent for erection.
  - P9 -> The cable erection equipment can be installed for erection.
  - P10-> The trucker(or floating) crane or trailer or the like can be entered into some part of working area.
- 
- C1 -> (1) Staging method with stiffleg derrick  
(2) Cantilever method with stiffleg derrick  
(3) Precast method with erection truss
  - C2 -> (1) Staging method with cable crane  
(2) Cantilever method with cable crane
  - ...
  - ...
  - ...
  - C9 -> (1) Erection truss method  
(2) Progressive placing method  
(3) Incremental launching method  
(4) Movable scaffolding method

Fig.4 A Flow for Choosing Construction Methods

### 3.3 Fuzzy 이론에 의한 비교안의 평가시스템

본 시스템에서는 평가항목의 애매한 요소를 표현하는 데 Fuzzy 이론을 이용하였다. 평가항목에 있어서의 경판성등은 설계자의 성격이나 학습과정(생활배경이나 평가자로서의 경험)등의 인간 본래의 주관에 기인하는 애매한 요소가 많이 포함된다. 이 같은 주관적이고 애매한 요소를 수학적으로 취급하기 위한 한 수법으로서 Fuzzy 이론이 있고, 확신도등으로부터 보

다 유연하게 애매한 요소가 표현가능한 것이다. 전문가 시스템은 인간에 의한 판단을 컴퓨터에 대항 또는 저원시키는 것이지만 기계에 걸기 위한 지식을 수치화하는 것은 피할 수가 없다. 예를 들면 IF(지간이 길다), THEN(크레인 가설공법이 적합하지 않다.)인 것을 IF(지간>60M), THEN(크레인 가설이 불가)라는 것처럼이다. 이런 사항들을 보완하는 것이 Fuzzy 이론이고 본 시스템에 있어서도 유효성이 있는 것으로 생각된다.

예를 들면 표 6에서 보는 바와 같이 두 평가항

Table 6 Rating Table for Example

Judgement factors	Weight	Ratings for A1	Ratings for A2
C1	very important	good	fair
C2	rather unimportant	fair	good

RULE 1  
 IF CONSIDER C1  
 THEN THE SUITABILITY OF USING A1 SHOULD BE GOOD  
 and THE SUITABILITY OF USING A2 SHOULD BE FAIR

RULE 2  
 IF CONSIDER C2  
 THEN THE SUITABILITY OF USING A1 SHOULD BE FAIR  
 and THE SUITABILITY OF USING A2 SHOULD BE GOOD

FACT: CONSIDER C1 (VERY IMPORTANT)  
 CONSIDER C2 (RATHER UNIMPORTANT)

Fig.5 Typical Rules for Example

목(judgement factor)으로서 두 후보(alternative)에 대한 평가를 원한다면, 단 여기서 C1, C2는 평가항목이고, A1, A2는 후보이며, good, fair는 등급(rating)을 나타낸다. 이를 rule의 형태로 표현하면 그림 5와 같다. 이와 같은 문제를 해결하기 위해 다음과 같은 방법을 이용한다. 만일  $A_1, A_2, \dots, A_m$  을 후보로 하고,  $a_1, a_2, \dots, a_n$  을 평가항목으로 할 때, 주어진 후보  $A_i$  에 관계된 평가항목  $a_j$  에 대한 등급은  $r_{ij}$  로 한다. 또한 평가항목  $a_j$  에 대한 중요도(weight)를  $w_j$  라 할 때, 후보  $A_i$  는 다음 식과 같은 weighted average rating을 갖는다.

$$r_i = \frac{\sum_{j=1}^n w_j r_{ij}}{\sum_{j=1}^n w_j} \quad (1)$$

이 weighted average rating은  $A_1, A_2, \dots, A_m$  의 후보순서로 유도된다. 이러한 접근방법은 실제문제에서 rating과 weight의 정확한 수치적 표현을 할 수 있다. 그러나 만일 rating과 weight가 fuzzy set으로서 표현된다면 후보  $A_i$  의 평가항목  $a_j$  에 대한 fuzzy rating은 membership function  $\mu_{r_{ij}}(r_{ij})$  로 나타내어진다(여기서  $r_{ij} \in R$ ). 이와 비슷하게 평가항목  $a_j$  에 대한 상대적인 중요도는  $\mu_{w_j}(w_j)$  로 나타내어진다(여기서  $w_j \in R$ ). 또한 모든 membership function은 [0,1] 사이의 값이다. 이때 rating과 weighted membership function 그래프는 그림 6과 같이 삼각형으로 나타내어진다. Fuzzy set을 이용해서 후보  $A_i$  의 ranking을 위해서 Bass와 Kwakernaak에 의해 발표된 방법을 고려하면 함수  $g_i(z): R^{2n} \rightarrow R$  은 다음 식과 같다.

$$g_i(z) = \frac{\sum_{j=1}^n w_j r_{ij}}{\sum_{j=1}^n w_j} = \bar{r}_i, \quad i = 1, \dots, m \quad (2)$$

$$z = \begin{pmatrix} w_1, \dots, w_n \\ r_{i1}, \dots, r_{in} \end{pmatrix}$$

또한 membership function  $\mu_z(z)$  은 다음 식과 같이 표현된다.

$$\mu_z(z) = \left[ \bigwedge_{j=1}^n \mu_{w_j}(w_j) \right] \wedge \left[ \bigwedge_{j=1}^n \mu_{r_{ij}}(r_{ij}) \right] \quad (3)$$

여기서  $g_i: R^{2n} \rightarrow R$ 를 Fuzzy set  $z_i$ 의 mapping을 통하여 Fuzzy set  $R_i$ 는 membership function과 함께 다음과 같이 유도된다.

$$\mu_{R_i}(\bar{r}) = \sup_{z: g_i(z) = \bar{r}} \mu_z(z), \quad \bar{r} \in R \quad (4)$$

이러한 membership function은 후보  $A_i$  의 마지막 rating을 부여한다. 결국 각 후보에 대한 최종적인 ranking 그래프가 얻어지고, 이 그래프에 의해 Fuzzy set을 배열한다. 최종적인 ranking 그래프의 해결을 위한 가장 직접적인 방법은 그래프의 좌편상의 비교이지만 이는 컴퓨터 실행에 있어서 적당하지 않으므로 여기서는 ranking 그래프 아래의 면적의 중심을 각 후보의 ranking 순서로 한다. 또한 최종적인 ranking 그래프의 계산시 continuous membership function 대신에 discrete membership function을 이용한다. 앞의 예에서의 각 후보에 대한 최종적인 ranking 그래프는 그림 7과 같고 이에 대한 면적의 중심은 표 7과 같다. 표 7에서 보는 바와 같이 continuous 그래프와 discrete 그래프에서 얻어진 중심의 값은 약간의 차이가 있는 것을 알 수 있는데 본 시스템에서는 discrete 그래프에 의한 중심의 값을 이용한다.

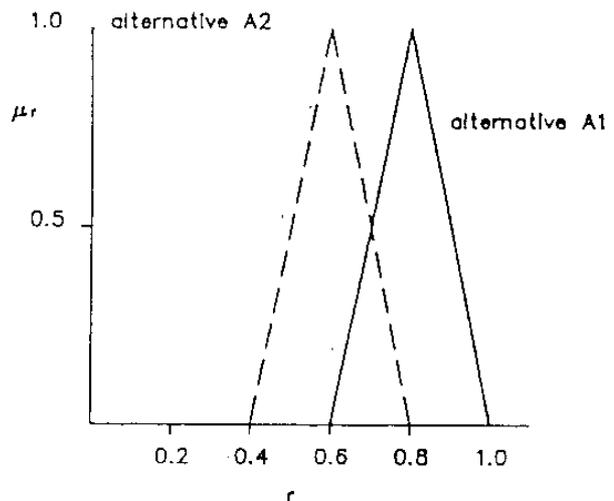


Fig. 7 The Final Ranking Graph for Example

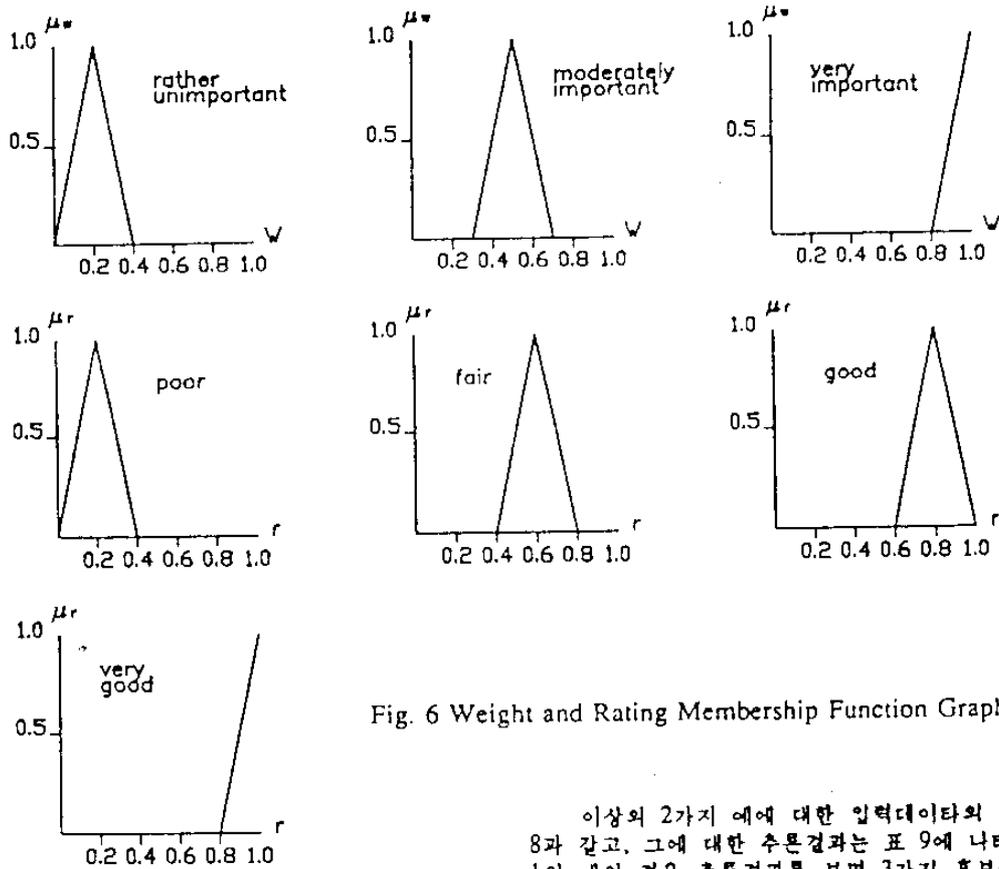


Fig. 6 Weight and Rating Membership Function Graph

Table 7 Centroid of the Final Ranking Graphs

Alternatives	Centroid	
	continuous graph (Fig.8)	discrete graph (obtained from the system)
A1	0.8	0.76
A2	0.6	0.63

#### 4. 본 시스템의 적용 예

본 연구에서 개발한 교량의 예비설계 시스템을 실제에 가설된 교량에 따라 적용하고, 그 적용에 따른 문제점을 검토한다.

(예 1) 본 장작교는 경기도 광주군 중부면 일천의 43번 국도와 광지천을 통과하며 도시계획 구역 및 학교 주변을 통과하는 상, 하행선 분리된 교량으로 교장은 305M이고, 7경간 연속교이며, 교각의 높이가 약 20M 정도로서 ILM(Incremental launching method)에 의한 pc box girder 형식으로 된 교량이다.

(예 2) 본 광지천교는 상, 하행선이 분리된 국도정단 교로서 교장은 200M이고, 5경간 연속교로 되어 있으며, 곡선교량형태로 되어 있고, 교량 아래에서 작업이 거의 불가능하게 되어 있는 steel box girder 형식으로 된 교량이다.

이상의 2가지 예에 대한 입력데이터의 형태는 표 8과 같고, 그에 대한 추천결과는 표 9에 나타나 있다. 1의 예의 경우 추천결과를 보면 3가지 후보중에서 제 1안인 precast cantilever 방법에 의한 pc box girder 형식의 후보가 채택됐다. 또한 2의 예의 경우 추천결과를 보면 3가지 후보중에서 floating crane의 가설장비를 이용한 cantilever 공법의 steel box girder 형식의 후보가 채택됐다. 이상의 두 예에서 보면 실제의 교량과는 상부구조의 형식은 일치하지만 가설공법상의 차이가 있는 데, 이는 본 시스템의 평가항목들이 실제 건설현장에서 설계자가 고려하는 평가항목에 비해서 불완전 하고, 또한 설계자의 의함에 따라 차이가 난다고 볼 수도 있을 것이다. 그렇지만 본 시스템에서 제공된 prototype은 어느정도 만족된 결과를 보인다고 할 수 있다.

#### 5. 결 론

본 논문에서는 중, 소 교량의 예비설계 단계에 있어서 상부구조의 형식 및 가설공법의 선정에 있어 기존의 전문적인 지식을 컴퓨터에 내장하여 전문가 시스템을 구축하였고, 또한 기존의 전문가 시스템 개발용 도구인 K-CLIPS를 이용함으로써 그 유효성에 대해서도 검토를 하였다. 그리고 추천결과와 불확실성을 Fuzzy set로 표현함으로써 서로 다른 전문가로부터 얻은 지식이나 데이터를 어느정도 효율적으로 처리하게 하였다.

앞으로 본 시스템이 완전한 시스템으로 이용되려면 모든 현장조건에 적합한 평가항목의 증가가 필요하고, 또한 중, 소 교량의 한계에서 벗어난 모든 교량에 적용할 수 있도록 본 시스템을 확장하여 구축하여야 하며, 마지막으로 예비설계 단계의 일부인 기초형식의 선정시스템을 추가로 구축하여야 하겠다.

Table 8 Input Data used in Example

Judgement factors	Data for Example 1	Data for Example 2
- bridge system	continuous span	continuous span
- bridge geometry	straight bridge	curved bridge
- main span length	45 M	50 M
- girder depth	2.65 M	0.5 - 1.5 M
- truss height	unknown	unknown
- arch rise	unknown	unknown
- bridge geometry (in elevation)	high level bridge	low level bridge
- bridge site	river area	river area
- consider ease of maintenance	yes (very important)	yes (very important)
- consider ease of fabrication	yes (very important)	yes (very important)
- consider appearance	yes (very important)	yes (very important)
- working area under the bridge available	yes	no
- it is possible to place temporary bent	no	no
- truck crane can be entered into working area	-	-
- there is a fabrication yard near site	yes	yes
- floating crane can be entered into working area	yes	no
- a cable crane can be installed	-	-
- there is a working space at the one or both end of the bridge	-	-
- some part of span can place temporary bent	-	-
- truck (or floating) crane can be entered into that part	-	-
- construction in valley area or like	no	no
- construction on soft ground	no	no
- consider method that possible work in narrow working area	yes (rather unimportant)	yes (moderately important)
- consider rapid construction	yes (moderately important)	yes (moderately important)
- consider multi-span construction	no	no
- consider method that minimal disturbance over existing traffic	yes (very important)	yes (very important)
- consider ease of construction	yes (very important)	yes (very important)
- consider economic method	yes (very important)	yes (very important)

Table 9 Results of the Expert System Selection (Given in Descending Order)

	TYPE
E X A M P L E 1	PC BOX GIRDER BRIDGE (.90) by precast cantilever method (.78)
	ORTHOTROPIC DECK BRIDGE (.82) by large block method with floating crane (.82)
	CONTINUOUS TRUSS BRIDGE (.45) by large block method with floating crane (.82)
E X A M P L E 2	STEEL BOX GIRDER BRIDGE (.70) by cantilever method with floating crane (.74)
	THROUGH LANGER BRIDGE (.70) by large block method with floating crane (.74)
	THROUGH LOSHE BRIDGE (.62) by large block method with floating crane (.70)

Note: The number in ( ) is the centroid of the area under a final rating graph

참 고 문 헌

1. M.L. Maher, Expert System for Civil Engineering: Technology and Application, A.S.C.E., 1987.
2. M. Biswas and J.G. Welch, BDES: A Bridge Design Expert System, Journal of Engineering with Computers 2, 1987, pp.125-136.
3. N. Harty, An Aid to Preliminary Design, Proceedings of second International Conference on Application of A.I. in Engineering Problems, Cambridge Massachusetts, 1987.
4. 대한토목학회, 도로교 표준시방서 해설, 사단법인 대한토목학회, 1982.
5. S.M. Bass and H. Kwakernaak, Rating and Ranking of Multiple Aspect Alternatives Using Fuzzy Sets Automatic, Vol. 13, 1977.

## 주 의

1. 이 보고서는 과학기술처에서 시행한 특정연구개발사업의 연구보고서이다.
2. 이 연구개발내용을 대외적으로 발표할 때는 반드시 과학기술처에서 시행한 특정연구개발사업의 연구결과임을 밝혀야 한다.