

설계지식 데이터베이스의 자료구조 규명과 자연어처리를 이용한 인터페이스 프로그램 개발

The Definition of Data Structure for Design Knowledge Database and Development of the Interface Program for using Natural Language Processing

이 정 재* · 이 민 호* · 윤 성 수**
Lee, Jeong Jae · Lee, Min Ho · Yoon, Seong Soo

Abstract

In this study, by using the natural language processing of the field of artificial intelligence, automated index was performed. And then, the Natural Language Processing Interface for knowledge representation(NALPI) has been developed. Furthermore, the DDesign KnOWledge DataBase(DEKODB) has been also developed, which is designed to interlock the knowledge base. The DEKODB processes both the documented design-data, like a concrete standard specification, and the design knowledge from an expert. The DEKODB is also simulates the design space of structures accordance with the production rule, and thus it is determined that DEKODB can be used as a engine to retrieve new knowledge and to implement knowledge base that is necessary to the development of automatic design system. The application field of the system, which has been developed in this study, can be expanded by supplement of the design knowledge at DEKODB and developing dictionaries for foreign languages. Furthermore, the perfect automation at the data accumulation and development of the automatic rule generator should benefit the unified design automation.

I. 서 론

구조물의 설계에는 안전성, 사용성, 경제성 등의 조건을 만족하기 위한 기준이 필요하다. 불확실한 물리적·재료적 특성을 가진 철근콘크리트 구조물 설계에서는 설계기준을 만족시키기 위해 지금까지 대부분 전문가의 경험적 지식이 중요한 역할을 담당하였다.^{2,6,7)} 그러나, 전문가의 경험적 지식은 감

각적이고, 비정형적이어서 전산화하기 어려운 반면, 콘크리트 표준시방서와 같은 문헌자료는 설계 정보를 객관적이고, 정형적으로 표현하고 있어 상대적으로 전산화가 용이하므로 문헌지식을 전산설계시스템에 효과적으로 이용하려는 연구가 있어왔다.^{5,8)}

전문가시스템은 많은 전문가를 대상으로 자료를 수집하여 지식을 획득하였기 때문에, 시스템을 구

*서울대학교 농업생명과학대학
**서울대학교 대학원

키워드 : 설계 지식, 자료구조, 자연언어처리

성하는 지식의 축적에 많은 시간이 소요되며, 획득된 지식을 전산시스템에 이용되는 규칙의 형태로 표현하기 어려운 점이 있다. 그러나, 구조물 설계 수요가 증가됨에 따라 전문가의 수가 절대적으로 부족했기 때문에, 초보 기술자에 의한 설계가 가능한 HI-RISE^{6,13)}와 SACON^{6,14)} 등의 전문가시스템의 개발이 이루어졌다.^{6,9,10,11)} 이들 전문가시스템은 설계시스템의 구성이 복잡하고 일단 시스템이 구성되면 수정이 어려우며, 처리해야 하는 지식의 양이 크게 증가하였으나, 대량의 표현된 지식을 관리하는 지식기반시스템의 기능이 미약하여 뚜렷한 발전을 거두지 못하였다.^{3,4,6,8)} 지식기반시스템에 관한 연구는 지식 표현, 추론 기법, 지식 습득 및 유지 방법 등에 중점을 두고 진행되어 왔다.^{4,8,11)} 지식기반시스템이 적용되는 응용 분야의 크기와 범위가 증가함에 따라, 지식베이스가 점차 대형화하고 복잡해지면서 기존의 지식기반시스템 만으로는 개발이 어려운 응용분야들이 발생하여서 데이터베이스 관리시스템과 지식기반시스템을 결합한 LKBS와 EDUCE 등과 같은 시스템이 개발되었다.^{4,11)} 그러나 이런 시스템은 설계지식을 표현하는 자료구조에 대한 정의가 명확하지 않고, 시방서 등 자연언어를 처리하지 못한다.

본 연구는 설계에 이용되는 콘크리트 표준시방서나 보고서 등과 같은 문헌상의 객관적인 지식뿐만 아니라, 문장형태로 표현 가능한 전문가의 경험적 지식인 설계지식을 표현할 수 있는 자료구조를 고찰하고, 설계지원 전문가시스템에서 규칙으로 사용될 수 있도록 정의된 자료구조를 가진 설계지식 데이터베이스(DEsign KnOwledge DataBase, DEKODB)를 개발하며, 설계지식 및 전문가의 경험지식 구축 환경인 자연어처리를 이용한 인터페이스 프로그램(NALPI, NATural Language Processing Interface program for knowledge representation)의 개발을 목적으로 하였다.

II. 설계지식 데이터베이스의 자료구조 규명

1. 설계지식의 표현

설계지식은 콘크리트 표준시방서, 보고서 등과 전문가의 경험적 지식과 같이 문장으로 표현할 수 있는 지식을 대상으로 하였다. 설계지식은 설계공식, 적용범위, 제약조건, 경계치 등에 대한 정보를 담고 있으며, 표현방법에 따라, 설계공식 자체로 표시, 한글구문이 설계연산 및 공식을 의미, 공식과 한글구문이 혼용 등으로 구분할 수 있다. 설계지식의 표현형식은 선언적인 지식을 사실로, 서술을 규칙으로 구분하여 데이터베이스에 저장이 가능하다.

가. 서술형 설계지식

문장으로 표현된 서술형 설계지식은 명제(proposition)와 같이 사실로 표현할 수 있는 논리형 정보와 일련의 행동에 따른 결과로 표현할 수 있는 규칙형 정보로 구성되어 있다. 논리형 정보를 이용하여 사실을 표현할 때는 논리 베이스와 관계 모델(logic-base and relational model)을 사용하며, 절차형 규칙 표현은 문제를 해결하는 방법의 지식을 저장하기 위해 규칙 베이스 모델(rule-base model)을 사용하였다.

논리형 정보에 의한 설계지식 표현은 사실(facts)과 정의(definition)로서 수학과 논리학에서 사용되었던 논리를 이용하는 명제로서 수식으로 표현할 수 있다. 명제로 표현된 설계지식은 참과 거짓을 판별할 수 있는 문장으로서 'and, or, not, plus, rmultiple, equivalent' 등과 같은 연결자(connector)와 연산자(operator)에 의해 연결될 수 있다. 논리적이며 수학적인 설계지식으로 사실을 표현하는 것은 문장 속에 내포된 개념을 포함하고 있으며, 수학적인 근거를 바탕으로 하고 있기 때문에 구조물의 설계지식을 정형화기에 적당하다. 규칙표현은 가정(if)과 결론(then)으로 규칙표현을 위한 절차형 모델은 규칙의 가정에 해당하는 조건부가 사실과 비교하여 만족될 경우에만 결론부를

- 콘크리트의 압축연단에서 이용할 수 있는 최대 변형률인 극한변형률은 0.003으로 가정한다.
- 횡방향 상대변위가 방지되어 있는 압축부재의 경우 유효길이계수 k 는 1을 사용해야 한다. 다만, 좀 더 적은 값을 사용할 수 있다는 것을 해석에 의해 증명할 수 있는 경우는 예외로 한다.
- 사용철근의 응력이 항복강도 f_y 이하일 때 철근의 응력은 그 변형률의 E_s 배로 취한다.
- k_1 은 f_{ck} 가 280kgf/cm^2 까지의 콘크리트에서는 0.85이다.

Fig. 1 Example of the design knowledge from the practical code

수행하게 된다. 본 연구에서는 구조물 설계를 위한 대표적인 문헌지식인 콘크리트 표준시방서와 보고서 등에서 발췌한 문장으로 표현할 수 있는 전문가의 경험적 지식을 설계지식 데이터베이스 자료구조 규명의 대상으로 하였다. Fig. 1은 시방서의 설계지식의 예이다.

나. 수식과 메타지식

수식과 메타지식은 설계에 이용되는 구조물의 기하학적 구성요소, 단면 및 재료, 하중조건 등을 표현하거나 설계가 이루어진 배경, 주요제약 조건 등의 통계 및 실험자료들의 결과를 정리한 것들로 구성된다. 메타지식은 표(table)와 도표(graph and chart) 등과 같이 사용자에게 의해 해석될 수 있는 설계지식을 의미한다. 구조물을 설계할 때 이용되는 수식과 메타지식은 구조물 구성요소로 보, 슬래브, 기둥, 벽체 등 그 대상이 한정되어 있고, 각 요소들도 단면정보, 하중정보, 부재력 정보 등을 공통으로 가지고 있으며 설계에 없어서 안될 자료들이므로 일반설계 프로그램을 작성할 때 이용된다.

2. 기호 처리

개별 어휘를 중심으로 문장을 분석하는 것은 자연 언어 처리 분야의 여러 응용에서 매우 바람직한 방법이 된다. 정보 검색을 위해서는 주어진 문

장에서 중요 어휘(keyword)나 중요 문맥을 추출하는 일은 필수적인 것이다. 이때, 문장 전체에 대해 성공적인 분석 결과를 얻을 수 없는 경우라도 그 문장을 이루는 독립적인 어휘들을 인식할 수 있다면 그로부터 부분적으로나마 정보를 얻을 수 있을 것이다. 입력 문장의 내용을 요약해 주는 시스템에서도 구문 분석에서 개별 어휘의 문맥을 인식한 결과가 주어진다면, 이들을 조합하여 의미 있는 내용으로 생성하는 것이 가능하다.

본 연구에서 대상으로 하는 콘크리트 표준시방서를 자연 언어로 처리하기 위해서는 하나의 기호를 여러 가지로 표현한 표기법을 통일하는 것이 필요하다. 콘크리트 표준시방서에 자주 나오는 '철근의 항복강도, f_y , 철근이 파괴될 때의 강도' 등은 모두 '철근항복강도'를 나타낸다는 것을 설계자는 알고 있지만, 컴퓨터는 이것이 모두 같은 의미라는 것을 프로그래밍 하지 않는다면 서로 다른 의미로 인식할 것이다. 따라서 본 논문에서는 Table 1과 같이 한글로 표현된 변수와 콘크리트 표준시방서에 기 제시된 기호를 본 연구에 적합한 하나의 대표어로 통일할 것이다.

Table 1 Example of the acknowledged term for symbol processing

한글 변수	변수	대표어	한글 변수	변수	대표어
극한모멘트	Mu	Mu	압축연단길이	a	a
유효폭	b	b	단면적	A	A
유효높이	d	d	모멘트팔길이	jd	jd
높이	h	h	철근량	As	As
철근뒹개	d'	d'	유효길이계수	k1	k1

3. 설계지식의 자료구조 정의

어떤 자료의 값(value)을 알기 위해서는 관련된 여러 개의 설계지식 조합에 의해 값을 추적할 수 있다. 이러한 복잡한 관계는 '상위자료는 하나 이상의 하위 자료로 구성되고, 하위 자료는 하나 이상의 상위 자료에 속할 수 있다'는 M:N 관계로

표현할 수 있다. 이와 같은 설계지식의 작업 흐름도를 개념적으로 구성하면 Fig. 2와 같다.

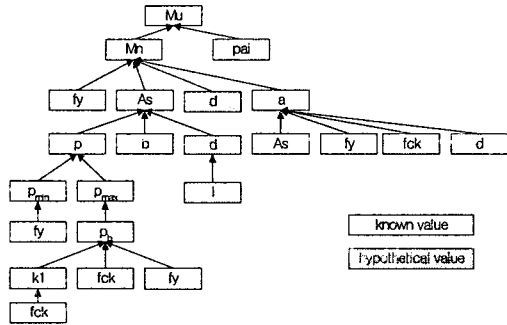


Fig. 2 Conceptual design of working flow

가. 설계지식 데이터베이스 테이블 스키마 정의
 설계지식 데이터베이스는 지식베이스 테이블, 수식 테이블, 사전 테이블 등으로 구성된다. 지식베이스는 서술형 설계지식과 메타지식을 처리하여 만들어지며, 논리형 정보와 메타지식을 저장하는 사실베이스와 규칙형 정보를 저장하는 지식베이스로 나눌 수 있다. 문장으로 표현된 설계지식을 데이터베이스에 구축하기 위해서는 두 가지 방법이 있다. 첫째, 문장으로 표현된 설계지식의 구문을 모두 표현할 수 있는 새로운 문장형식을 찾아 제시하고 이 형식에 맞게 설계지식을 새롭게 작성하는 것이다. 둘째, 문장으로 표현된 설계지식을 모두 저장할 수 있는 테이블의 기본 형태를 제시하고 모든 구문을 이 테이블에 저장하는 방법이 있다. 전자는 원형 문장을 제시된 형식에 맞게 변환하는 과정이 필요하며, 일단 변환되면 테이블에 쉽게 자료를 구축할 수 있는 장점이 있다. 그러나, 자료를 구축하는 사람의 임의 판단에 의한 침착이 수반되는 단점을 가지고 있다. 후자는 설계지식을 모두 표현할 수 있는 기본 테이블을 정의하고, 후처리를 통하여 테이블에 설계정보를 구축하기 때문에 원형 문장의 변형없이 처리할 수 있다는 장점이 있다. 하지만, 후처리를 위한 시스템의 개발에 따른 부담이 크다. 본 연구에서는 원형 문장의 변형이 없는 후자의

방법을 통해 데이터베이스를 구축하였다. 설계지식의 특징을 가장 잘 표현할 수 있는 데이터베이스 테이블을 정의할 수 있다면, 문장으로 표현할 수 있는 모든 설계지식을 자연어처리를 이용한 전처리를 통해 데이터베이스에 구축할 수 있을 것이다. 문장으로 표현된 서술형 설계지식과 메타지식은 지식베이스 테이블로 구축이 가능하지만, 그 이외의 설계지식인 수식과 기호는 문장으로 되어 있지 않기 때문에 자연어처리 시스템으로 처리할 수 없다. 수식과 기호는 자연어처리가 아닌 사용자의 별도의 입력을 통해 설계지식 데이터베이스에 구축된다. 기호는 대표어사전(Acknowledged term dictionary)에 설계변수와 함께 구축된다. Fig. 3은 정의된 테이블 스키마간의 관계를 나타낸다.

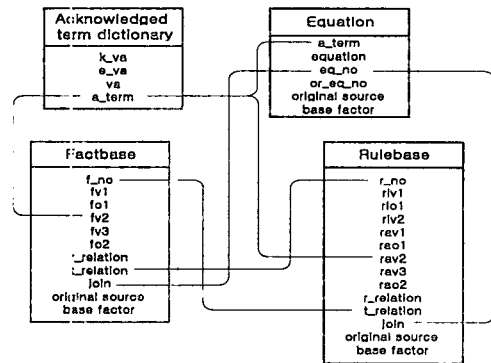


Fig. 3 Relation chart of the defined tables

나. 설계지식 자료의 관리

구축할 지식의 양이 많아지고, 다양한 문헌자료로부터 설계지식을 추출할 경우 서로 다른 이론을 수용한 문헌자료에 의해 발생하고, 새로운 연구결과에 의해 지식의 내용이 변하여 그 변경 내용을 새롭게 설계지식 데이터베이스에 구축할 경우 지식의 비일관성이 발생한다. 비일관성의 종류에는 중복규칙(redundant rule), 모순규칙(contradiction rule), 충돌규칙(conflict rule), 포함규칙(subsumption rule), 불필요한 전제부(unnecessary condition) 등이 있으며, 지식의 비일관성은 구축된 설계지식

Table 2 Comparison between knowledge-base system and database management system

	Knowledge-Base System	Database System
Data Expression	Complex Explicit Easy update	Simple Presentation internal meta data Difficult update
Data Form	Few instance Don't Guarantee faultless Consist of fact and rule	Many instance Guarantee faultless Consist of fact
Solve	General Use heuristic	Use Matching Simple
Environment	Alone user Virtual memory Central type	Multi user Use Assistance Memory Distributed type
Performance	Slow	Fast

데이터베이스를 이용할 경우 예측치 못한 결과를 발생시킬 수 있기 때문에 제거되어야 한다.

그리고, 이러한 지식의 비밀관성은 지식기반시스템과 데이터베이스 관리시스템간에 이용되는 자료형의 차이에서 발생할 수 있다. Table 2는 지식기반 시스템과 데이터베이스 관리시스템 사이에 발생할 수 있는 차이점을 비교하였다. 지식기반시스템과 데이터베이스 관리시스템의 결합에서 가장 중요한 문제는 자료형의 차이에 의한 결합 불일치(unified mismatch)를 해결하는 것이다. 결합 불일치 문제는 표현력(expressability) 문제, 기능(functionality) 문제, 성능(performance) 문제로 구분되며, 본 연구에서는 결합 불일치를 해결할 수 있도록 자료구조를 정의하여 자료를 관리하였다.

Ⅲ. 자연어처리를 이용한 인터페이스 프로그램 개발

제시된 설계지식 데이터베이스의 자료구조를 이용하여 문장으로 표현된 설계지식으로부터 지식을 획득하기 위한 자연어 처리를 이용한 인터페이스 프로그램인 NALPI(NATURAL Language Processing Interface for knowledge representation)를 개발하였다. NALPI는 크게 자연 언어를 처리하기 위해 대표어로 변환하는 사전부와 자연어처리 전

처리부, HAM에 의해 문장을 분석하는 언어처리부, HAM에 의해 분석된 결과를 제시하는 결과제시부로 나눌 수 있다.

1. 대표어 변환

가. 대표어사전 작성

자연 언어로 표현된 설계지식을 처리하기 위해서는 전술한 것과 같이 하나의 기호를 여러 가지로 표현하고 있는 표기법을 통일하는 것이 중요하다. 컴퓨터는 프로그램된 그대로 작업을 처리하기 때문에, 동일한 기호에 대한 표기법이 상이하다면 작업 결과도 역시 상이할 것이다.

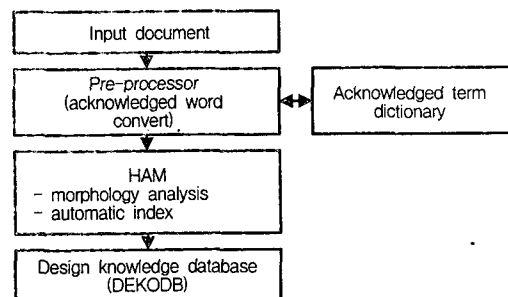


Fig. 4 Conceptual design of NALPI

Table 3과 같이 다양한 표현에 대한 자연어처리

방법에는 모든 가능성 있는 표현을 미리 별도의 사전으로 제시해 두는 방법과 분석된 결과를 가지고 후처리를 하는 두 가지 방법이 있을 수 있다. 전자의 경우는 가장 확실하고 자동화된 처리를 할 수 있는 방법이라는 장점에도 불구하고, 제시되지 않은 표현이 출현할 가능성과 작성된 별도의 사건의 활용에 있어서 시스템의 성능을 떨어뜨릴 수 있다는 단점이 있다. 후자는 별도의 사전을 작성할 필요가 없으며 사용자와의 인터페이스를 통하여 빠른 시간에 정확한 분석 결과를 찾아낼 수는 있으나, 시스템의 자동화라는 측면에서 단점으로 지적될 수 있을 것이다.

본 연구에서 사용된 대표어사전은 자연어처리 시스템의 효율성을 높이며 신뢰성 있는 분석 결과를 얻기 위하여, 구조물 설계에 주로 이용되는 콘크리트 표준시방서의 기호와 설계변수를 중심으로 Table 4와 같이 작성하였다.

나. 한글 자연어처리의 전처리

문장의 형태로 입력된 설계지식은 콘크리트 표준시방서에서 일상적으로 사용되는 각종 부호들과, 영어, 한자, 특수기호 등의 다양한 표기들이

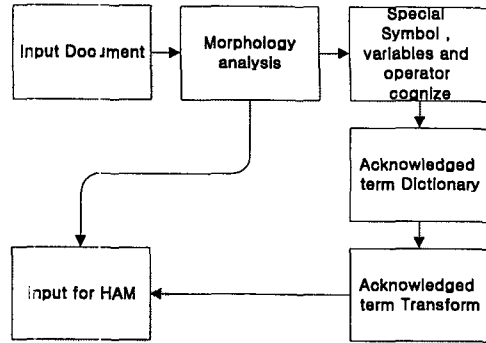


Fig. 5 Flow chart of natural language pre-processor

포함되어 있기 때문에, 먼저 이들에 대한 처리가 필요하다. 본 연구에서는 다음과 같이 용어 및 기호의 표기를 통일하기 위해 자연어 전처리 알고리즘을 Fig. 5와 같이 개발하였다.

2. HAM을 이용한 자동 색인 및 서술형 설계 지식 전처리기 개발

본 연구에서는 HAM(Hangul Analysis Module)¹⁾ 중이서 형태소 분석을 기반으로 하는 자동 색인(automatic indexing) 기법을 사용하였다. 형

Table 3 Various expression of the concrete specification

required expression	expression I in document	expression II in document
철근항복강도	철근의 항복강도...	철근이 항복상태에 도달할 때의 강도...
철근응력	철근의 응력...	사용철근의 응력...
콘크리트 극한변형률	콘크리트의 압축연단에서... 극한변형률...	콘크리트의 가정 극한변형률...

Table 4 Example of acknowledged term dictionary

한글 변수	변수	대표어	한글 변수	변수	대표어
극한모멘트	M_u	Mu	사하중계수	ϕ_d	paid
사하중모멘트	M_d	Md	활하중계수	ϕ_l	pail
활하중모멘트	M_l	Ml	균형철근비	ρ_b	pb
콘크리트 압축강도	f_{ck}	fck	최소철근비	ρ_{min}	pmin
철근항복강도	f_y	fy	인장철근비	ρ	p
유효폭	b	b	압축철근비	ρ'	pc
유효높이	d	d	최대철근비	ρ_{max}	pmax

태소 분석은 하나의 말마디를 여러 형태소로 분리한 후, 형태적 변형을 처리하는 것이다. 형태소 분석을 한 후, 입력된 문장으로부터 주요어(key word)를 추출해내기 위해서는 자동 색인 기법이 이용된다. 형태소 분석을 통한 자동 색인 기법에서는 문헌을 이루고 있는 각 문장에 대한 형태소 분석결과로부터 주제를 나타내는 단어나 구를 식별해 내는 것이 일반적이며, 한국어와 같이 조사가 발달된 경우에는 한 어절에서 조사를 제외하 나머지를 색인으로 채택한다. 자동 색인시 발생할 수 있는 오류와 필요 없는 색인 결과를 제거하기 위하여 서술형 설계지식의 pre-processor를 개발하였다.

가. 자연어처리를 이용한 자동색인

정보를 탐색할 때 색인은 방대한 양의 정보로부터 사용자가 원하는 정보만을 걸러 주는 여과기와 같은 구실을 한다. 만약 설계지식을 내포한 문장을 색인한다면, 사용자 자신의 수작업에 의한 방법과 컴퓨터를 이용하여 자동적으로 색인하는 방법으로 나눌 수 있다. HAM의 자동색인은 다음과 같이 이루어진다. 먼저 문장에 포함된 색인어를 추출하는 과정은 Fig. 6과 같이 형태소 분석에 의하여 정보자료를 분석하여, 색인어 후보를 생성하는 과정과 생성된 후보 중에서 불용어(stopword) 처리와 특수 색인어 추출 과정에 의하여 색인어를 선택하는 과정으로 이루어진다. 그리고 자동 색인시 가장 문제가 되는 복합명사 인식 기능, 미등록어 추정 기능, 불용어 처리 기능이 사용자 사전, 전문용어사전, 불용어사전 등에 의해 보강되었다. 그러나 형태소 분석을 통한 자동색인은 후보 색인어 생성시 '... 가장 많다.'의 부사 '가장'을 '집안의 어른'이라는 의미로 추출하는 오류가 발생한다. 이 문제를 극복하기 위해 Fig. 7과 같은 이상적인 자동 색인 시스템의 구조가 필요하다. 그리고, 본 연구에서 이용한 HAM은 이상적인 자동 색인 시스템의 구조 중에서 후처리기를 제외하고 다른 기능적인 면에서는 신뢰성 있게 구현되었다. 본 연구에서는 현재까지 개발된 HAM의 기능을 사용자가

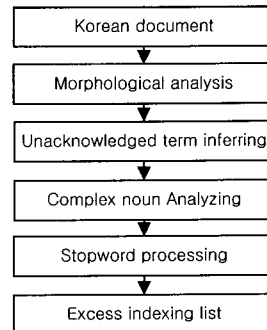


Fig. 6 Automatic index for morphological analyzing

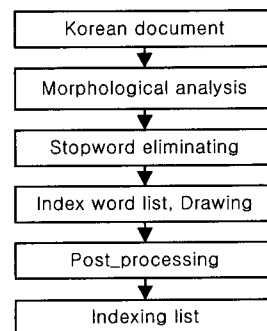


Fig. 7 Idealistic system structure of automatic index

쉽게 이용할 수 있도록 자동색인 결과를 제시할 수 있으며, 설계지식 전산화된 처리를 위한 시도로서 NALPI를 개발하였다. NALPI는 Java™를 사용하여 운영체계에 관계없이 사용할 수 있도록 하였다. 또한 설계지식데이터베이스인 DEKODB는 DBMS인 Postgres™를 이용하여 구성하였다.

3. NALPI의 운용

본 연구에서는 관계형 데이터베이스 관리시스템인 MS Access를 이용하여 설계지식 데이터베이스를 구축하였다. Fig. 8, Fig. 9는 NALPI를 이용하여 설계지식 데이터베이스를 구축하는 모습이다.

4. 적용 및 고찰

설계지식 데이터베이스에 구축되어 관리되는 설계지식의 적용성과 적합성을 평가하기 위해서는

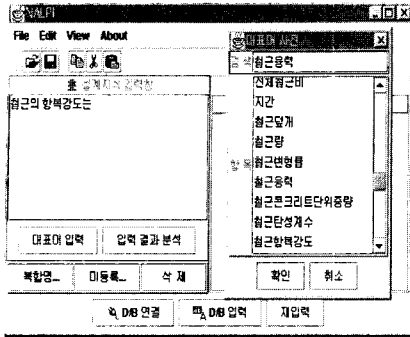


Fig. 8 Display of connecting the acknowledged term dictionary for input the design knowledge document

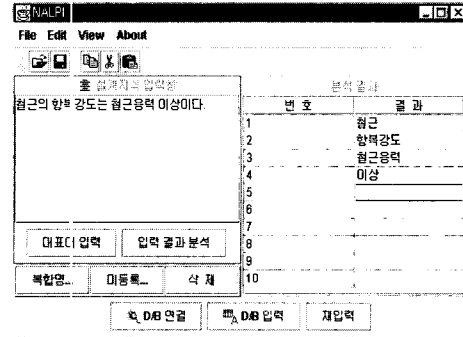


Fig. 9 Display of printing the indexing result

전문가시스템에서 추론을 위해 필요한 데이터를 정확하게 공급할 수 있어야 한다. DEKODB에 구

축한 자료가 실제로 전문가시스템에 이용될 수 있다는 것을 검증하기 위하여 Fig. 10의 단순보를 대상으로 적용성을 평가하였다. Table 5는 10명의

Table 5 Design procedure of expert

Order	Detail design	Expert
cross-section assumption	assumption of 'd' assumption of 'b'	$d/b = 1.5 \sim 2.5$
load computation	computation of dead load combination of load	table 3.1 equation 3.9
steel ratio assumption	computation of p_{max} & p_{min}	table 4.3.1 equation 4.3
normal strength computation	computation of 'As' computation of 'a'	$p = A_s/bd$ $a = A_s \cdot f_y / (0.85 \cdot f_{ck} \cdot b)$
trial method	strength reduction coefficient(ϕ) strength compare	$M_u \leq \phi \cdot M_n$

Table 6 Working step of manually query

Order	Constraint	Query	Searching value	Next query
1	Boundary condition Input condition(strength)	Mu	compare & trial method	p_{ai} , M_n
1.1	consider base factor	p_{ai}	constant	
1.2	hypothetical value	M_n	d	A_s , d, a
1.2.1	hypothetical value	A_s	b, d	p
1.2.1.1	hypothetical value	p	p	p_{min} , p_{max}
1.2.1.1.1	consider base factor	p_{min}	operation	constant
1.2.1.1.2	consider base factor	p_{max}	operation	p_b
1.2.1.1.2.1	consider base factor	p_b	operation	k1
1.2.1.1.2.1.1	Input condition	k1	operation	
1.2.2	hypothetical value	d	operation	l
1.2.2.1	Input condition	l	constant	
1.2.3	operation	a	operation	

전문가들의 실제 설계과정을 정리한 것이다. Fig. 11은 실제 전문가시스템에서 후향 추론을 할 때, 추론에 필요한 데이터의 흐름을 나타내고 있으며, Table 7은 DEKODB로부터 추출한 자료를 정리한 것이다. 또한, Fig. 12는 문장으로 표현된 설계지식의 원문을 NALPI에 의해 전처리를 하고 DEKODB에 자료로서 구축한 것을 수작업을 통해 EMYCIN의 규칙으로 전환하는 것을 순차적으로 제시하고 있다. 따라서, DEKODB로부터 추출한

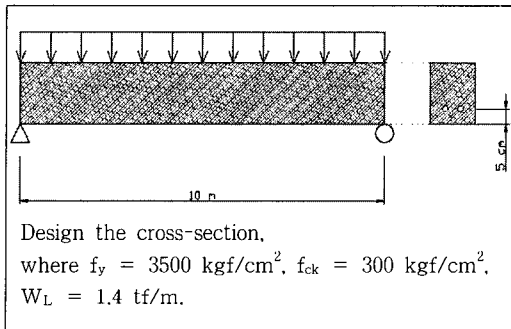


Fig. 10 Design example

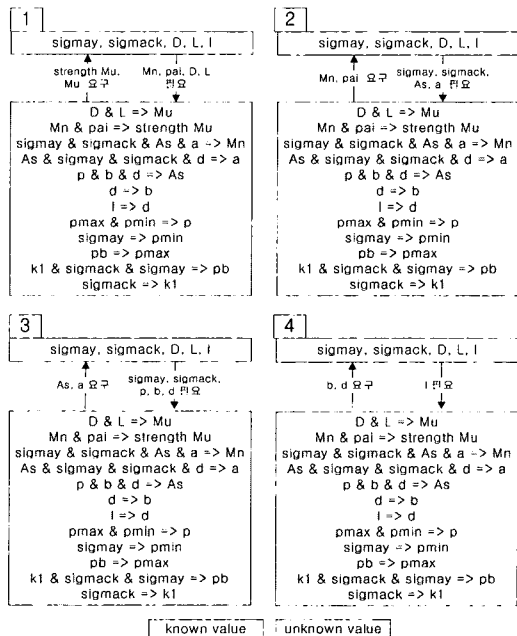


Fig. 11 Example of Working data flow for reasoning

Table 7 Searching data from the DEKODB

Search	Knowledge base output	Equation output
Mu	f29	e12
Mn	f9	e1
pai	f11	
As		e9
a		e3
d	f26, f27, f30	
p		e8
Pmin	f15	e7
Pmax	f27	
Pb		e1
k1	f13, r6, r7	e10

* f-: knowledge of the fact-base table
 * r-: knowledge of the rule-base table
 * e-: knowledge of the equation table

- 1) Design knowledge
 - ① 휨 보강 철근항복강도는 5000kgf/cm²을 초과할 수 없다.
 - ② k1은 fck가 280kgf/cm²이하이면 0.85이다.
- 2) Preprocessing using NALPI
 - ① 휨, 보강, 철근항복강도, 5000, 이하
 - ② k1, fck, 280, 이하, 0.85
- 3) Constructured Data in DEKODB
 - ① 철근항복강도, 5000, 이하, 휨, 보강
 - ② fck, 280, 이하, k1, 0.85
- 4) Transformed Rule Expression
 - ① (\$AND (SAME CNTXT AT (LISTOF FY)) (SMALLERP* CNTXT 5000))
 - ② (\$AND (SAME CNTXT AT (LISTOF FCK)) (SMALLERP* CNTXT 280)) (CONCLUDE CNTXT K1 0.85)

Fig. 12 Converting the rule to the DEKODB's data

자료는 전문가시스템의 지식베이스로 확장성이 있음을 알 수 있다.

IV. 결 론

문장으로 표현할 수 있는 설계지식을 데이터의 유지·보수가 편리하며 설계지원 시스템에 쉽게 이용될 수 있는 설계지식 데이터베이스에 축적하

기 위하여 자료구조를 정의하고, 인공지능 분야의 하나인 자연어처리를 이용한 인터페이스 프로그램인 NALPI를 개발하였다. NALPI에 의해 처리된 자료는 정의된 자료구조에 의해 지식베이스로 확장이 가능한 DEKODB에 구축하였으며, 그 결과는 다음과 같다.

1. 문장으로 표현할 수 있는 모든 설계지식을 데이터베이스에 구축할 수 있는 자료구조를 정의하였다.

2. 자연어처리를 이용하여 콘크리트 표준시방서와 문장으로 표현된 전문가의 경험 지식으로부터 DEKODB의 자료를 생성할 수 있었다.

3. DEKODB는 전문가시스템 등의 설계지원 시스템에 이용되는 지식베이스로의 확장이 가능함을 확인하였으며, 자료의 관리가 쉽다.

4. DEKODB로부터 추출한 자료는 전문가시스템을 이용한 추론에 필요한 데이터를 모두 공급할 수 있으며 전문가에 의한 설계결과와 비교하여 설계지식의 높은 재현성이 있었다.

앞으로 DEKODB로 구축할 설계지식의 범위를 확대하고, 영어와 같은 타 언어에 대한 대표어사전을 구축하고, HAM의 출력 결과를 이용한 자동화된 후처리부가 개발된다면, 본 연구에서 개발한 알고리즘의 적용 영역이 확대될 것으로 기대된다. 자료의 자동생성을 위한 자료축적 프로그램에 관한 연구가 이루어지고, DEKODB로부터 추출된 자료를 자동으로 다양한 형태의 설계지원 시스템에 적용할 수 있도록 하는 자료변환기(data converter)가 개발된다면 통합설계자동화에 기여할 수 있을 것으로 판단된다.

참 고 문 헌

1. 강승식, 1996, HAM : 한국어 형태소 분석 라이브러리, 한글공학연구소
2. 고일두, 김문현, 최창근, 1983, 컴퓨터를 이용한 RC 부재 자동설계에 관한 연구, 건설기술고도화 시스템의 개발에 관한 연구(I), 과학기술원
3. 류은상, 1995, 데이터베이스를 이용한 전문가

시스템의 효율적인 지식베이스 구축에 관한 연구, 국방대학원 석사학위 논문

4. 민민경, 1993, 대규모 지식베이스의 운용 : 지식 표현, 매칭, 저장 관리기법, 서울대학교 박사학위 논문
5. 박익수, 정성구, 신태양, 신남수, 유영동, 1998, 하이퍼미디어 지원 지능형 건축설계 학습 시스템에 관한 기초적 연구, 대한건축학회논문집 14 권 1호, p61-70
6. 장주흠, 윤성수, 김한중, 이정재, 1999, 구조물 설계에 있어서 데이터베이스 관리 시스템의 응용에 관한 연구, 한국농공학회지 3권 8호
7. 정종현, 이재철, 김치경, 홍성목, 1995, 바닥판 시스템 설계자동화를 위한 데이터베이스, 대한건축학회 춘추학술발표대회 논문집, p45-50
8. 조문상, 1996, 건축 설계 지식베이스 개발을 위한 Knowledge Acquisition 방법에 관한 연구, 대한건축학회논문집 12권 6호
9. 홍성목, 허명제, 고일두, 김치경, 1989, 인공지능을 이용한 구조설계 전문가시스템에 관한 연구(I), 대한건축학회논문집 5권 3호
10. 홍성목, 허명제, 고일두, 김치경, 1989, 인공지능을 이용한 구조설계 전문가시스템에 관한 연구(II), 대한건축학회논문집 5권 4호
11. Brodie M. L., 1988, Future Intelligent Information Systems: AI and Database Technologies Working Together, Readings in Artificial Intelligence and Databases p623-641
12. Habibullah A., 1988, SAFE(Slab Analysis by the Finite Element Method), Computers & Structures Inc.
13. King D., 1990, Intelligent Decision Support: Strategies for Integrating Decision Support, Database Management, and Expert System Technologies, Expert Systems with Applications vol1 no1 p23-38
14. Sellis T., 1990, A homogeneous Approach to Bringing AI and DBMS Together, Proceedings of the CompSac p32-33