

論文2001-38CI-6-6

고장진단을 위한 지식기반 시스템의 설계 및 구현

(Design and Implementation of Knowledge Base System for Fault Diagnosis)

田根 瓊*, 申盛允*, 申定薰**, 李羊遠***, 柳根鎬****

(Keun Hwan Jeon, Sung Yun Shin, Jeong Hun Shin, Yang Won Lee, and Keun Ho Ryu)

요 약

전문가 시스템은 인공지능의 한 분야로서, 인간의 사고방식을 모방함으로써 다양한 분야에서 야기되는 문제들을 해결해준다. 대부분의 전문가 시스템은 추론엔진과 지식베이스등과 같은 많은 요소들로 구성된다. 특히 전문가 시스템의 성능은 추론엔진의 효율성에 의해 좌우된다. 이러한 추론 엔진은 지식베이스가 구축될 때, 가능한 적은 제약성을 가져야 함은 물론, 다양한 추론 방법을 제공해야 한다는 특징을 갖고 있어야 한다. 이 논문에서는 도메인 지식 표현을 위한 지식스키마와, 추론을 위한 지식구현 기법, 그리고 블랙보드와 추론엔진을 혼합한 지식베이스 엔진을 제안한다. 그리고 제안한 기법들을 이용하여 구축한 중공업 장비 진단 전문가 시스템에 대해서 설명한다. 이 논문에서 연구한 고장진단 지식기반 시스템은 지식기반 시스템 연구분야의 실질적 기반이 될 수 있을 것이다.

Abstract

Expert system is one of AI area. It simulates the human's way of thinking to give solutions of problem in many applications. Most expert system consists of many components such as inference engine, knowledge base, and so on. Especially the performance of expert system depend on the control of efficiency of inference engine. Inference engine has to get features; first, if possible to minimize restrictions when it constructed the knowledge base. second, it has to serve various kinds of inferencing methods. In this paper we propose knowledge scheme for representing domain knowledge in ease, knowledge implementation technique for inferencing, and integrated knowledge-base engine with blackboard and inference engine. And we describe a expert system prototype that implemented in this paper using proposed methods, it perform diagnose about heavy industrial device. The fault diagnosis system prototype has been studied in this paper will be practical foundation in the research area of knowledge based system.

* 正會員, 群長大學 컴퓨터應用學部
(Division of Computer Application, Kunjang College, Professor)

** 正會員, 益山大學 電子計算科
(Dept. of Computer Science, Iksan College, Professor)

*** 正會員, 群山大學校 컴퓨터情報科學科

(Dept. of Computer Science Kunsan national university, Professor)

**** 正會員, 忠北大學校 컴퓨터科學科
(Dept. of Computer Science Chungbuk national university, Professor)

接受日字:2001年5月22日, 수정완료일:2001年8月3日

I. 서 론

전문가는 당면한 문제해결을 위해서 문제발생 배경 및 문제의 성격 등의 정보를 두뇌에 담긴 경험지식에 적용하여 해결방법을 제시하는 일련의 임무를 수행한다. 이를 위해 전문가는 특정 분야의 최고 경험적 지식을 소유하고 있어야 한다. 하지만 전문가가 갖는 문제점 즉, 양성의 어려움, 과도한 유지비용, 그리고 특정분야가 한사람에 의존되는 현상 등을 해결하기 위한 대안으로 조금씩 연구되기 시작한 것이 전문가 시스템의 등장배경이라 할 수 있다. 이러한 대안에 대한 활발한 연구의 시발점을 부여해준 시대적 배경은 인공지능 분야의 발전이다.

지식기반 시스템의 응용분야는 거의 무한하며, 특히 문제해결을 위하여 전문가의 지식이 필요한 분야에 적절하다. 응용영역은 법률, 화학, 생물, 공학, 제조, 항공, 군사, 금융, 기상, 지질 등 실로 다양하다. 많은 연구자들은 이러한 응용분야를 몇 가지 범주로 분류하였다. 예를 들면, Rychener^[1]는 진단, 설계, 기획의 세 범주로 분류하였고, Miller^[2]는 운영, 생산, 공학 측면으로 분류하였으며, Waterman^[3]은 해석, 예측, 진단, 설계, 기획, 모니터링, 디버깅, 수리, 명령, 제어 등으로 보다 자세하게 분류하였다. 많은 응용분야 중에서도 진단은 가장 활발하게 개발되는 분야로서, MYCIN^[4], CASNET^[5]과 같이 초기 진단 지식기반 시스템은 질병 진단을 위한 의료분야가 주축을 이루었으나, 최근에는 운영분야나 장비의 오 동작에 대한 자동진단 및 수리가 진단 지식기반 시스템의 목표가 되었다.

지식기반 시스템은 전문가의 지식을 규칙, 프레임, 객체 또는 가설, 일차 술어논리 등의 일부 또는 전부로 표현한 지식베이스를 기반으로 한다. 또한 전문가의 사고유형에 해당하는 지식베이스 엔진은 추론엔진이나 블랙보드만으로 또는 두 구조를 혼합한 형태로 구성되며, 이 지식베이스 엔진은 지식베이스의 지식들을 이용하여 진단을 수행하기 위한 새로운 지식을 생성하기도 한다.

이 연구에서는 신뢰성과 적응력을 갖춘 전문가 시스템을 구축하는 과정에서 시스템의 각 구성요소들 중 지식구현과 추론엔진 요소를 중심으로 연구내용을 제안하고 전반적인 전문가 시스템 구축에 대해 설명한다. 이를 위해, 지식베이스 엔진으로는 추론엔진과 블랙보

드 두 구조를 결합한 형태로 구현하고, 지식베이스는 실세계의 여러 자료 유형들을 잘 혼용하여 필요에 따라 활용될 수 있는 스키마를 구성하여 구축한다. 지식 표현은 규칙을 기반으로 가설을 이용하여 이들 규칙을 클러스터링 하였고, 생성규칙의 구현과 추론을 위해 기존의 절-비트 매트릭스(Clause-Bit Matrix : CBM)^[6]를 연산자까지 표현할 수 있는 형태로 확장한 ECBM(Extended Clause Bit Matrix)을 제안한다^[7]. 끝으로, 사용자가 실세계의 지식을 용이하게 획득하고 지식베이스를 효율적으로 유지관리하기 위하여 인터페이스 측면에서 규칙 편집기와 객체 및 가설 편집기를 개발하여 지식기반 시스템에 연결하였다.

이 논문의 제 2장에서는 관련연구로 지식베이스 구축에 필요한 지식표현과 구현 방법들을 살펴보고 기존에 개발된 지식기반 시스템들을 살펴본 후, 지식베이스 엔진을 검토한다. 제 3장에서는 지식베이스 스키마를 제시하고, ECBM을 이용한 지식의 구현과 지식베이스 구축방안을 제시한다. 제 4장에서는 블랙보드와 추론엔진으로 구성된 지식베이스 엔진을 제안하고, 제 5장에서 프로토타입을 선정하여 고장진단 지식기반 시스템 구현과 함께 시스템의 성능을 평가 분석한다. 제 6장에서 결론 및 향후 연구내용의 제시로 이 논문을 마무리한다.

II. 관련연구

추론을 위한 지식 표현 방법은 대표적으로 규칙, 객체, 가설 등이 있고, 지식베이스 엔진은 추론을 시행하는 과정에서 필요한 프로시저들의 논리적인 묶음으로서 추론엔진기반, 블랙보드 기반, 그리고 이 둘을 혼합한 혼합구조로 나눌 수 있다^[8].

이 장에서는 본 연구에서 제안하여 채택한 생성규칙 기반의 지식의 구현방법과 추론엔진 구성방법의 타당성 제시를 위해 기존에 연구된 내용들을 지식구현 기법과 추론엔진을 중심으로 살펴본다.

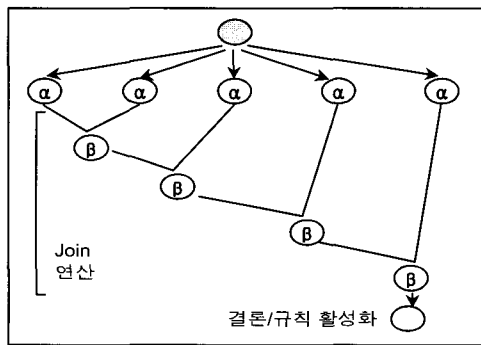
2.1 지식구현

지식기반 시스템의 주요 구성요소로는 추론엔진과 지식베이스를 제기한다.

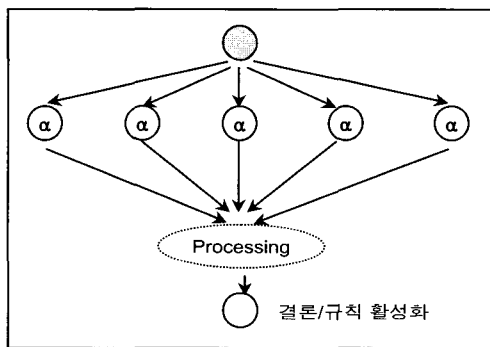
기존 구현된 시스템들에서 지식표현 형태를 살펴보면 대부분의 시스템에서 표현의 다양성 획득을 위해 몇 가지 표현기법을 혼합하여 표현하였다. P-Shell^[9]에

서는 일차 술어논리를 기반으로 프레임, 프로시저, 규칙, 의미망 외에 테이블, 트리 표현까지 수용하였고, Prolog/Rex^[10]에서는 프레임과 규칙, 프로시저의 세 가지 표현법을 이용하였으며, MANTRA^[11]에서는 P-Shell과 비슷하게 일차 술어논리를 기반으로, 프레임, 의미망 그리고 규칙으로 지식을 표현하였다^[12,13,14,15,16]. 이렇게 표현된 지식은 패턴매칭을 위해 트리 구조나 구현이 쉬운 매트릭스 구조 또는 관계리스트 구조 등으로 구현된다.

트리 구조의 대표적 형태인 Rete^[17]와 TREAT^[18,19]는 다수의 객체에 대한 다수의 패턴매칭 기법으로, 규칙들에 대한 패턴매칭 속도는 빠르지만, 알파와 베타메모리의 보관에 따른 저장공간의 비효율성이 단점으로 지적된다. Rete의 저장공간 비 효율성 해결을 위해 변형된 알고리즘 중의 하나가 TREAT^[19]이다. 하지만 TREAT는 구조적인 면에서 탐색속도가 Rete보다 느려, Rete의 단점에 대한 대안의 역할보다는 변형된 새로운 패턴매칭 기법의 역할을 수행한다^[18]. 다음 그림 1은 Rete(a)



(a) Rete의 예



(b) TREAT의 예

그림 1. 트리구조를 이용한 지식 구현
Fig. 1. Knowledge implementation using tree structure.

와 TREAT(b)에 대한 논리적 구조이다. (a)는 알파메모리에 대한 반복적인 조인 연산으로 발생하는 베타메모리가 (b)TREAT에서는 나타나지 않음을 보이고 있다.

매트릭스 표현은 규칙의 전제부와 결론부를 각각 X, Y축에 위치시킨 2차원 행렬이다. 이 방법은 추적이 매우 빠르며 구현이 매우 쉽다는 장점을 가진 다중연결을 표현할 수 있는 자료구조이다^[6,7]. 매트릭스 표현에는 규칙을 하나의 단위로 표현한 규칙-비트 매트릭스와 절을 구성하는 각각의 사실 정보를 단위로 표현한 절-비트 매트릭스, 그리고 절-비트 매트릭스의 저장공간을 축소시킨 축소-절-비트 매트릭스 표현이 있다^[6,7].

규칙비트 매트릭스(Rule Bit Matrix : RBM)는 각 규칙의 결론과 모든 규칙들의 전제부를 비교하여 연결이 발견되면, 1 값을 주고, 그렇지 못하면 0 값을 준다. 즉, 규칙의 수가 10개이면 10×10, 100개이면 100×100개의 저장소가 필요하게 된다.

그림 2는 지식베이스를 규칙 비트 매트릭스로 구축한 예를 보여준다. 이 방법을 사용하면 병렬(parallel)로 처리가 가능하여 그 속도를 증진시키고, 규칙의 추가, 삭제 및 수정이 용이하고 추론 경로를 설명하는데 이용할 수 있으며, 의사결정 트리의 어디에서나 추론이 시작될 수 있으며, RBM의 사용으로 다중 결론을 상세화할 수 있다. 또한 환형 규칙(circular rule)의 타당성을 검토하는 알고리즘을 두어 검증할 수 있다는 장점에 비해 저장공간 문제와 규칙의 조건부에 포함된 fact 정보를 표현할 수 없다는 단점을 갖고 있다.

R1	<i>if Q, then X</i>	0	0	1	0	0	0	1	0	0	0	0
R2	<i>if M, then Y</i>	0	0	0	0	0	0	0	0	0	0	0
R3	<i>if W and X, then G and Z</i>	0	0	0	0	1	0	0	0	0	0	1
R4	<i>if A and B, then C</i>	0	0	0	0	0	1	0	0	0	0	0
R5	<i>if D or Z, then E</i>	0	0	0	0	0	1	0	0	0	0	0
R6	<i>if C and E, then F</i>	0	0	0	0	0	0	0	1	1	0	0
R7	<i>if X, then A</i>	0	0	1	0	0	0	0	0	0	0	0
R8	<i>if Y, then B</i>	0	0	0	1	0	0	0	0	0	0	0
R9	<i>if F, then S</i>	0	0	0	0	0	0	0	0	0	0	0
R10	<i>if G, then R</i>	0	0	0	0	0	0	0	0	0	0	0

그림 2. 지식베이스 예와 규칙 비트 매트릭스의 예
Fig. 2. Example Knowledge-based and Rule Bit Matrix.

절-비트 매트릭스(Clause-Bit Matrix:CBM)는 RBM의 단점인 절을 구성하는 각각의 fact 정보를 표현하지 못하는 단점을 해결하기 위해 연구된 방법이다. 이는 각 규칙의 전제부와 결론부에 인식번호(Identification number)를 주어 매트릭스로 표현한다. 이와 같이 규칙

의 전제부와 결론부를 연결 해 줌으로써 RBM에서 발생할 수 있는 충돌을 방지할 수 있다. 다음 그림 3은 그림 2의 지식베이스에 대해 구축한 절-비트 매트릭스의 예이다.

		전제부															
		1	2	3	3	4	5	5	6	6	7	8	9	10			
규칙#	↓	1	1	1	2	1	2	1	2	1	1	1	1	1			
절-비트 매트릭스	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	3	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	4	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	5	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	7	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	8	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

그림 3. 절-비트 매트릭스 표현
Fig. 3. Clause-Bit Matrix Representation.

절 비트 매트릭스 방법의 단점은 규칙의 수가 많아지면, 트리형태의 Rete나 TREAT 등의 방법에 비해, 기억공간의 손실이 더 크다는 점이다. 이러한 단점을 극복하기 위한 방안으로 소개된 방법이 Reduced CBM^[6]이다. Reduced CBM은 CBM을 구축하는 방법과 같이 각 절에 대해 비교하여 매트릭스를 표현하되, 비교 시에 한가지 조건을 검토한다. 조건이란, “완전한 부정값(“0” bit)을 갖는 절인가”의 불리언 값으로, 조건의 값이 긍정일 경우, 계속해서 검토를 해 나아가며 매트릭스를 생성시키고, 조건의 값이 부정일 경우, 해당 규칙의 절에서 해당 열과 행을 삭제시킨다. 이는 불필요한 정보를 삭제시킴으로써 기억공간의 손실을 줄이고자 하는데 목적이 있다.

생성 규칙의 표현방법 중 하나인 관계 리스트는 특수한 리스트를 사용하여 빠른 탐색과 수정의 용이성을 제공한다. 또한 최근에는 퍼지 개념을 도입할 수 있다는 장점도 제기되고 있다^[6]. 이는 규칙의 결론부에 대해 각기 연계관계를 갖고 있는 규칙의 전제부에 대한 정보를 갖고 있게하여 탐색 시에 곧장 찾아 들어갈 수 있게 하여 준다. 아래 그림 4는 그림 2의 예제 지식베이스에 대한 규칙을 중심으로 구축한 것으로 규칙 R1의 결론절은 규칙 R3과 R7의 조건만족을 위한 선 실행 규칙으로 전체 지식들의 활성화 순서를 용이하게 찾아갈 수 있다.

이와 같이 살펴본 방법들 중 트리만인 AND/OR의

조건을 읽어 들여 처리하는데 용이할 뿐 비트 매트릭스 방법에서는 절간에 존재하는 연산자 정보를 표현하지 못하고 있다.

결론절		조건절	
규칙#	절#	규칙#	절#
1	1	3	2
1	1	7	1
2	1	8	1
3	1	10	1
3	2	5	2
4	1	6	1
5	1	6	2
6	1	9	1
7	1	4	1
8	1	4	2

그림 4. 관계 리스트 표현
Fig. 4. Relational List Representation.

2.2 지식베이스 엔진

지식베이스 엔진(knowledge-base engine)은 추론을 시행하는 과정에서 필요한 프로시저들의 논리적인 묶음으로서 추론엔진기반, 블랙보드 기반, 그리고 이 둘을 혼합한 혼합구조로 나눌 수 있다^[8].

추론엔진기반의 지식베이스 엔진은 더 이상 참고해야 할 규칙이 없을 때까지 비교, 선택, 수행을 반복한다. 추론의 매주기마다 모든 규칙들을 현 상태의 사실들과 다시 비교해야 하기 때문에, 실제 추론에 사용되지 않는 규칙들까지도 검토하게 되어 시간의 낭비를 초래한다. 추론기반의 지식베이스 엔진은 규칙기반의 특정 형태에 적절하며, 구축이 쉽고 동일한 유형의 통합관리에 효율적이다.

블랙보드 기반의 지식베이스 엔진은 다양한 전문지식, 목표, 신뢰도, 그리고 지식표현을 하나의 구조화된 틀로 병합시킬 수 있는 방법을 제공한다. 일반적인 블랙보드 시스템은 구조화된 블랙보드와 구조화된 각각의 부분에 위치하게 될 지식원과 이들을 통제하기 위한 컨트롤로 구성되며 실시간 응용에 이상적인 적합성을 제공한다^[20,21,22].

혼합 지식베이스 엔진은 앞에서 살펴본 추론엔진과 블랙보드의 특성을 조합한 엔진 구조이다. 전체적으로는 블랙보드 방식으로 제어하고 국부적으로는 도메인에 따른 제어를 하는 두 제어방식을 혼용하며, 현실적인 적용방법으로 일반화되어 있는 형태이다^[23].

다음 표 1에서는 추론엔진과 블랙보드 기반 엔진의 장단점을 분석하여 정리하였다.

표 1. 기존의 엔진 구조 분석
Table 1. Analysis of Conventional Engine structure.

중심 엔진	시스템	장점	단점
추론엔진 기반	OPSS MYCIN G2	규칙기반 같은 특정 형태에 적합 Tightly Coupled type 구축 용이 동일한 유형의 지식에 대한 통합관리에 효율적 주로 Backward 또는 Forward 추론 적용	지식표현의 다양성 제한 적용 도메인에 한정적 다중 에이전트 구조에 부적합
블랙보드 엔진	BB1 GBB	다양한 지식표현을 이용한 추론이 가능 Loosely Coupled type 동적으로 지정지식 적용이 용이 Opportunistic 추론에 용이 상대적으로 복잡한 문제해결 도메인에 적용	후향추론에 적합한 구조 단순하고 비 실시간 도메인 적용이 부적합

이 연구에서는 지식 표현을 규칙간의 상호 독립적이며, 수정이 용이하고 쉽게 설명이 가능하고 쉽게 접근할 수 있는 규칙과, 주어진 증상에 대해 모든 도메인 지식을 비교함으로써 발생하는 시간을 축소하여 추론의 신속성과 신뢰성을 추구할 수 있는 가설을 혼용한 지식구조를 제안하고 또, 지식구현을 위해서 생성규칙의 구현에 이용되었던 기존 방법들의 단점을 해결할 수 있는 확장된 절-비트 매트릭스(Extended Clause Bit Matrix:ECBM)를 이용하여 구축한다.

III. 지식베이스 설계

3.1 지식베이스 스키마

이 논문에서 접근하고자 하는 지식표현 목표는 먼저, 특정 도메인 지향으로 제한적인 성격이 아닌, 실세계의 일반적인 도메인에 제한 없는 적용 가능성과 지식공학자가 지식표현의 형태를 쉽게 이해하고 습득할 수 있어야 한다는 것이다. 또한 지식표현을 기반으로 하는 추론성능 측면에서는 우선 추론운영 방식이 제한되어 있지 않아 사용자의 요구에 의해 또는 적용될 지식 도메인의 성격에 따라 적절한 운영방식을 선택하여 수행할 수 있어야 한다^[8].

지식베이스의 지식은 크게 객체, 규칙 그리고 가설로 구성되며, 객체는 전반적인 지식표현을 위한 가장 기본

적인 자료구조로서 대상 장비에서 산출되는 증상데이터와 규칙, 가설로 표현한다. 지식은 도메인 지식과 이 도메인 지식의 상위 지식에 해당하는 메타 지식으로 계층화하고 메타 지식은 "Rule about rule"의 개념으로서 메타가설 추론기의 지식으로 이용되며, 도메인 지식은 규칙과 사실로 나누고, 규칙은 활성화 규칙과 비 활성화 규칙으로 다시 세분하였으며, 사실(fact) 또한 활성화 사실과 비 활성화 사실로 세분하였다.

이런 지식들의 연관성이 용이하도록 규칙과 가설 지식을 조직화한다. 가설들은 모여서 하나의 가설트리를 형성하고, 각각의 단말가설들은 가설의 검증에 사용될 규칙집합을 가지며, 이 논문에서는 이러한 가설검증에 이용되는 규칙집합을 도메인규칙이라 한다. 가설트리는 여러 개가 존재할 수 있다. 이 논문에서는 여러 가설트리가 존재할 경우를 가정하고, 이들 여러 가설트리에 대한 관리 및 진단시의 우선 순위를 나타내기 위해서 메타 지식 즉, 메타 규칙집합을 사용한다. 여기에서의 메타규칙은 추론의 흐름 또는 셸의 제어를 변경하는 역할이 아니라, 추론 시작 전의 초기화 작업이나 여러 개의 가설트리 중에서 어떤 가설트리를 먼저 고려할 것인가 그리고 추론을 종료할 때 나타내는 조건을 명시하는 역할을 한다.

객체와 규칙은 클래스를 중심으로 계층적인 관계를 갖는다. 하나의 클래스 "Ball Bearing"은 이를 구성하는 많은 객체들 즉, Cage 결합, 내륜결합, 외륜결합 등으로 구성되며 각 객체들도 이들 객체들을 구성하는 속성들 즉, 센서, 볼, 울, 축 등으로 구성된다. 규칙은 이러한 클래스나 객체들을 진단하기 위해 메타속성이나, 각 속성들에 해당하는 그룹으로 나누어 다음 그림 5 와 같은 관계를 갖는다.

규칙은 적어도 하나의 조건과 하나의 가설을 가져야

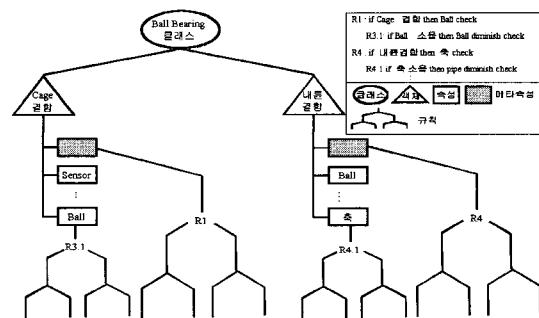


그림 5. 객체와 규칙간의 관계
Fig. 5. Relationship of Object and Rule.

한다. 규칙의 조건은 절로써 표현되며, 두 개 이상의 절은 논리연산자 AND나 OR에 의해 결합된다. 규칙에 포함된 가설은 동작(action)으로 표현되며, 또 다른 가설과 논리연산자에 의해 관계를 갖는다. 그림 6은 규칙과 가설사이의 관계를 나타내고 있다. 그림 6에서 가설을 중심으로 규칙 R1과 R2의 OR관계는 가설 내에 포함된 논리 연산자로서 예를 들면 “축 결합은 연결된 기어의 불균형 또는 연결부위의 느슨함으로 발생된다”와 같은 가설일 경우 갖을 수 있는 연산자이다.

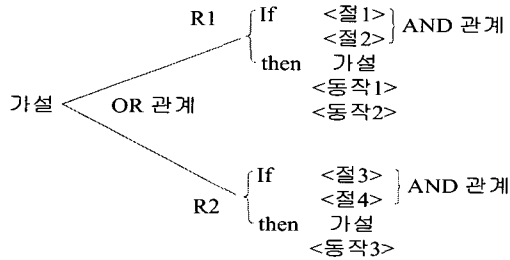


그림 6. 규칙과 가설간의 관계
Fig. 6. Relationship of Rule and Hypothesis.

3.3 지식구현을 위한 ECBM

ECBM은 x축에 규칙의 전제부 절을, y축에 결론부 절을 표현한 2차원 테이블인 CBM을 확장하였다. CBM은 규칙의 전제부와 결론부의 절들의 연관관계는 명확하게 표현할 수 있으나 각 절들 사이를 표현하는 연산자는 표현할 수 없다. 이러한 단점을 보완한 ECBM은 CBM의 구조위에 AND/OR 상태 표시 테이블을 추가하여 3차원으로 구성함으로써, 추론 시에 규칙의 조건을 고려하여 완전한 추론을 할 수 있게 한다. 즉 기존의 CBM에 AND/OR 연산이 가능하도록 조건상태 표시를 할 수 있는 비트를 추가하여 그림 7과 같은 3차원 구조를 갖게 하였다. 예를 들어, 기존의 CBM을 이용하여 추론을 할 경우에는 규칙이 “If A AND B Then C”라 했을 때, 전제부의 “AND”에 대한 별도의 조치가 없이 먼저 읽어들이게 되어지는 A에 대한 탐색, 그 후 B에 대한 탐색을 해 나아간다. 그러나, 여기에서 ECBM은 규칙에서 AND를 인식하여 저장하고 추론 시 이를 이용하여 AND조건이 모두 만족할 때 추론이 계속될 수 있도록 한다.

그림 8은 그림 2의 예제 규칙을 중심으로 구축한 ECBM을 나타낸다.

ECBM의 장점은 첫째, 일단 생성된 후 이를 이용할

수 있는 추론운영 방식에 제한을 두지 않는 것으로, 이는 연구의 주요목적인 범용성을 갖는 지식베이스 엔진을 가능하게 해준다. 둘째, 지식의 검증이 용이하다. 일반적으로 잘못된 정보를 갖고 추론을 한 부정확한 결과를 산출하지 않기 위해 지식의 검증이 반드시 필요하지만 트리표현 등 기타방법을 이용한 지식베이스 구축에서는 검증이 쉽지 않다. 셋째, 특정 도메인에 국한되지 않아 범용성을 갖는다. 반면에 ECBM을 지식구축 방법으로 이용할 경우 규칙의 수가 많아질수록 그에 비례하여 많은 널 값을 저장해야 하므로 이에 따른 메모리의 손실을 고려해야 한다.

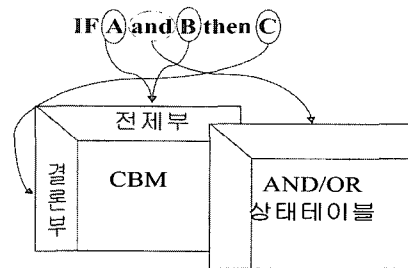


그림 7. ECBM 생성원리
Fig. 7. Principle of ECBM.

		AND/OR																
		전제부																
		0	0	2	2	1	2	0	0	0	0							
결론부	0																	
	0			1	2	3	3	4	4	5	5	6	6	7	8	9	10	
	0			1	1	1	2	1	2	1	2	1	2	1	1	1	1	
	2	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0	0	3	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	0	4	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	0	0	5	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0	0	7	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	8	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
0	0	9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

그림 8. 지식베이스와 ECBM
Fig. 8. Knowledge base and ECBM.

IV. 지식베이스 엔진

4.1 지식베이스 엔진구조

지식기반 시스템의 여러 구성요소 중에서 지식베이스를 기반으로 기호추론을 수행하여 의사결정을 위한 결론을 도출해내는 모듈이 바로 지식베이스 엔진이다. 지식베이스 엔진에서 수행하는 추론전략의 중요한 두 가지 형태는 전향추론과 후향추론이다. 이러한 추론전략

의 수행은 RETE를 기반으로 추론을 수행하는 OPS5의 경우 스케줄러, 패턴매처, 실행기의 세 가지 모듈들로 수행하고, 블랙보드를 기반으로 수행하는 BB계열 시스템들은 블랙보드를 중심으로 컨트롤에 의해 추론 전략을 수행하도록 하고 있다. OPS5는 전향추론에 적합한 구조이고, BB계열 시스템들이 채택한 블랙보드 구조는 후향추론에 적합한 구조라는 차이점이 존재한다.

그림 9 는 이 논문에서 제안하는 궁극적인 지식기반 시스템의 구성전략을 나타낸다. 각각의 지식원은 추론 엔진 또는 블랙보드가 중심이 되어 또 다른 지식원과 상호 통신을 통해 여러 형태의 지식을 공유할 수 있도록 구성하였다.

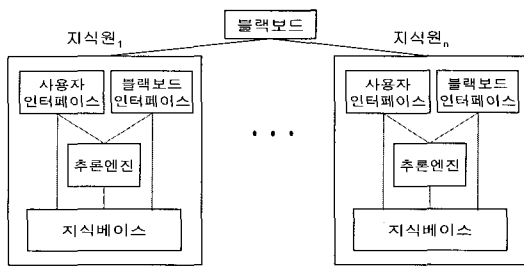


그림 9. 제안 지식베이스 엔진구조
Fig. 9. Structure of Proposed Knowledge base Engine.

혼합형 구조는 표 1에서 언급한 기존 구조와 비교하여 다음과 같은 장점을 갖는다.

- 첫째, 다양하고 전문적인 지식베이스 수용이 쉽다.
- 둘째, 상이한 문제해결 표현과 경험을 허용하는 기본틀의 통합이 용이하여 여러 지식 형태의 적용이 요구되는 복잡한 도메인에 쉽게 적용할 수 있다.
- 셋째, 문제해결의 동적 제어가 가능하여 다양한 추론 기법을 적용함으로써 융통성이 부여된다.

본 연구에서는 위의 장점에 근거하여 전향추론과 후향추론 모두에 적합하고, 다중 대리자 구조로의 확장을 쉽게 하기 위해 추론엔진과 블랙보드를 혼합한 형태의 지식베이스 엔진을 제안한다.

이 시스템의 지식베이스 엔진에서 블랙보드는 블랙보드와 에이전다로, 추론엔진은 스케줄러, 실행기, 패턴매처, 그리고 이벤트생성기로 구성되며, 전체적으로는 블랙보드 방식으로 제어하고 국부적으로는 추론엔진의 흐름에 의해 제어된다.

추론엔진은 이벤트 생성기, 패턴매처, 스케줄러, 실행기로 구성된다. 이벤트 생성기는 추론엔진이 기동할 수

있도록 이벤트를 생성하여 블랙보드에 전달한다. 이 시스템에서 이벤트는 'data-driven' 방식으로서 모니터링 결과로 입력되는 자료를 이벤트로 규정한다. 이벤트 생성기에서 생성된 이벤트를 시작으로 시스템의 내부 제어주기에 의해 추론주기를 수행한다. 패턴매처는 블랙보드에 존재하는 입력값(이벤트)에 대해 활성화될 규칙을 찾아내는 패턴매칭이 가장 주된 기능이다. 이 논문에서는 검색속도가 우수하고 특정 도메인에 국한되지 않는 범용성을 갖으며, 구현이 용이한 ECBM을 사용하여 패턴매칭을 수행하도록 하였다. 스케줄러는 에이전다에 누적된 실행 가능한 지식원들 중에서 가장 적당한 하나를 선정하여 실행기에서 실행되도록 하는 모듈로서 충돌해결 전략을 실행한다. 여러 가지 충돌해결 전략 중 이 시스템에서는 선입선출만을 지원하며 다른 전략은 추후 과제로 남겨둔다. 실행기는 추론주기의 마지막 단계를 수행하는 모듈로서 지식원의 실행부를 실행한다. 실행의 결과가 단말결과인 경우 사용자에게 질의를 한 후 결론 여부를 결정하거나, 곧바로 결론으로 제시한다. 실행결과가 결론이 아닌 경우에는 새로운 사실로서 블랙보드에 입력되면서 새로운 이벤트를 발생시켜 추론을 반복시킨다. 다음 그림 10 은 이와 같은 실행에 대한 흐름도 이다.

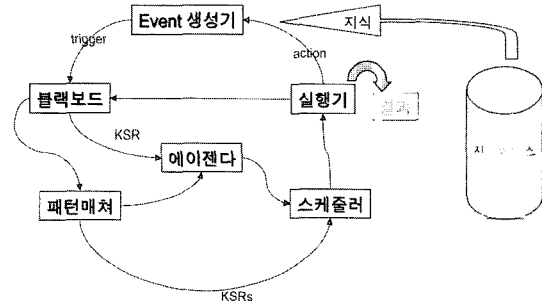


그림 10. 엔진 세부 수행 흐름
Fig. 10. Detailed Execution Flow of Engine.

4.2 ECBM을 이용한 추론전략

본 장에서는 제안한 ECBM을 추론 시에 이용하는 지식의 구축형태로 하고, 전향추론과 후향추론을 하는 과정을 보인다.

추론과정을 설명하기 전에 사전 정의적인 요소들은 다음과 같다.

첫째, 블랙보드는 단순 작업메모리 기능을 수행하며, 사용자나 컴퓨터 검토에 의해서 입력된 초기치를 갖는다.

둘째, 추론방법은 사용자의 선택에 의해 결정된다.
 셋째, 추론 함수의 호출 시에 사용되는 값은 블랙보드의 초기치와 지식베이스이다.
 넷째, 충돌전략은 선입선출(FIFO)을 기본값으로 설정한다.
 ECBM을 이용한 전향추론과 후향추론의 수행순서는 그림 11 과 같다.

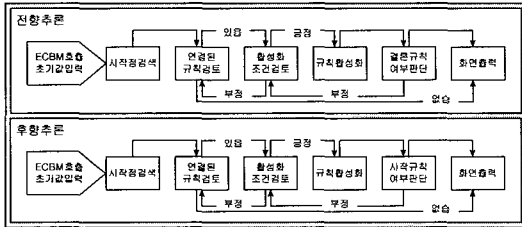


그림 11. ECBM을 이용한 전향·후향추론의 흐름도
 Fig. 11. Flow Diagram of Forward/Backward chaining using ECBM.

ECBM을 이용한 추론은 매트릭스의 형태를 기본으로 하고 있기 때문에 추론방법이 다르더라도 수행 순서는 같다. 다만 전향추론일 경우, 연결된 규칙 검색부분과 규칙 검토 부분에서 백트래킹(backtracking)과 충돌문제 해결 등의 다소 복잡한 알고리즘을 사용한다는 점이 다르다.

V. 구현 및 평가

5.1 실험 도메인 및 실험지식 획득

삼성중공업에서 제시한 중공업 기계를 구성하는 회전부품의 고장진단 도메인을 대상으로 지식기반 시스템을 설계·구현하여 항만하역 컨테이너 크레인(Transfer Crane)에 적용을 목적으로 한다. 크레인은 크레인을 구성하는 회전부품의 고장징후 판단을 목적으로 센서를 설치하여 실시간으로 전송되는 증상 값을 중심으로 고장 부품을 탐지하고 지식베이스에 있는 진단지식을 제공하는 역할을 수행한다. 대상 도메인에서는 증상 데이터의 실시간 획득과 이벤트 기반 진단추론과 향후 실시간 시스템으로의 확장을 기본 요구사항으로 제시하였다.

도메인 지식획득은 삼성중공업 중앙연구소로부터 제공받았으며, 이 지식을 구성요소 트리와 증상 트리로 구분 설계하여 지식베이스를 구축하였다.

이 논문에서 구현된 고장진단 지식기반 시스템은 단일 사용자용으로 Windows 95 운영체제하에서 Visual C++ 4.2를 사용하였고, 실험을 위해 삼성중공업 중앙연구소에서 제공된 지식을 이용하였다.

지식베이스 구축에서는 지식편집을 위한 인터페이스 “클래스 생성기”와 “객체 생성기”를 사용하였다. 클래스 생성기는 제공하는 인터페이스를 이용해서 지식기반 시스템에서 사용되는 클래스를 정의할 수 있으며, 센서의 공통된 속성을 정의하는 클래스를 생성하고 생성된 속성의 편집이 가능하다. 예제 클래스 Sensor는 센서설치방향인 D0, 신호의 종류 DI, 모니터링 목표를 의미하는 D2와 속성범위를 나타내는 스펙트럼 속성 D 3~D17, 과 추론 실행에 이용되는 변수 Z, A, B, C, D 등을 갖는다.

객체 생성기는 클래스와 객체간에 속성상속의 자동화와 함께 객체생성 폼에서는 객체를 생성할 수 있으며, 추가적으로 객체 계층도를 보여주는 브라우저를 지원하고, 지식베이스 파일로는 클래스와 객체 정보를 갖는 파일을 생성한다.

이 시스템은 회전부품의 고장징후 판단을 목적으로 회전체에 센서를 부착한 후 징후 탐지에 의해 자동적인 고장진단을 수행하는데, 고장진단을 위해 가설을 사용한다. 실험데이터에 대한 가설은 그림 12 에 나타나 있다. 가설은 상위가설과 하위가설 또는 단말가설과 비단말가설로 구분되는데, Ball Bearing 결함을 나타내는 가설 H-1은 상위가설과 비 단말가설이고, Cage 결함을 나타내는 가설 H-2는 하위가설과 단말가설임을 알 수 있다.

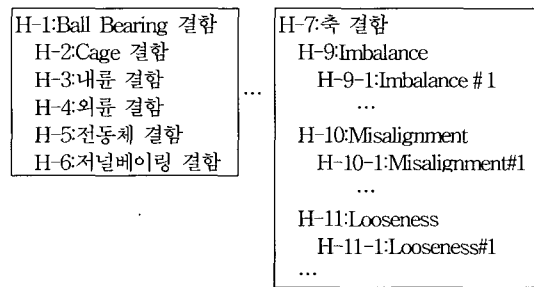


그림 12. 실험 데이터(가설)
 Fig. 12. Experimental Data(Hypothesis).

이들 가설 지식은 가설의 편집과 가설획득, 또 획득된 가설의 계층 구조를 구조적으로 보여줌으로써 가설 획득의 용이성을 제공하는 가설 편집기로 작성할 수

있다.

가설트리에는 여러 개가 존재할 수 있으며, 여러 가설 트리에 대한 관리 및 진단시의 우선순위를 나타내기 위해서 메타규칙 집합을 사용한다. 메타규칙은 추론의 흐름 또는 셸의 제어를 변경하는 역할이 아니라, 추론 시작전의 초기화 작업이나 여러 개의 가설트리 중에서 어떤 가설 트리를 먼저 고려할 것인가 그리고 추론의 종료점을 나타내는 조건을 명시하는 역할을 한다.

이 시스템에서, 규칙은 비 단말가설을 검증하기 위한 메타규칙과 단말가설을 검증하기 위한 도메인규칙으로 구분되는데, 메타규칙에 의해 부품의 고장징후 판단을 할 수 있게 된다. 실험데이터에 대한 메타규칙은 그림 13 에 나타나있다.

```

If H-12 is True Then H-1 is Check
If H-1 is False Then H-14 And H-13 is Check
If H-9 is True
Then H-10 is Check And H-11 is Check
. . .
If H-10 is True Then H-1 is Check
And H-6 is Check And Diagnosis is End

```

그림 13. 실험 데이터(메타규칙)
Fig. 13. Experimental Data(Meta Rule).

규칙편집기는 사용자에게 규칙획득 및 관리기능을 제공하기 때문에 지식획득의 용이성과 효율적인 지식 관리를 지원해준다. 규칙편집기는 가설을 선택, 규칙편집, 규칙수정 등의 연산을 제공하고, 내부적으로 새로운 가설선택 및 종료 시 규칙집합에 대한 관계리스트, ECBM, 규칙의 질, 그리고 가설에 대한 지식베이스가 구축된다. 그림 14 는 단말가설 H-8(Oil Whirl)의 검증을 위한 도메인규칙이다.

```

(n=1~13)
If Z of Sensor-n is False And D1 = 0 of Sensor-n
And D2 = 1 of Sensor-n
Then A of Sensor-n is True And B is B+1
If A of Sensor-n is True And D4 ≥ 0.5 of Sensor-n
Then C is C+1 And D of Sensor-n is True
If C/B > 0.5 And User is True
Then H-8 is True

```

그림 14. 실험 데이터(도메인규칙)
Fig. 14. Experimental Data(Domain Rule).

그림 15는 규칙편집기에서 규칙 내용을 입력하는 화면을 단계별로 나타낸 것이다. 우선 규칙 이름을 정하

고, 입력할 규칙이 메타규칙이나 도메인규칙 중 어떠한 집단에 속하는가를 정의한 후 상위 가설을 선택하여 준다.

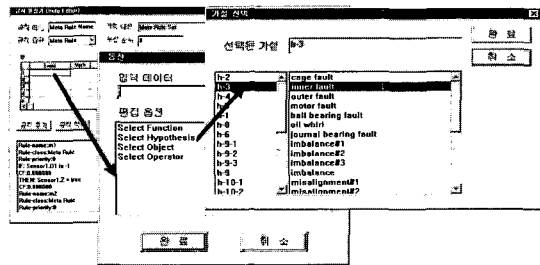
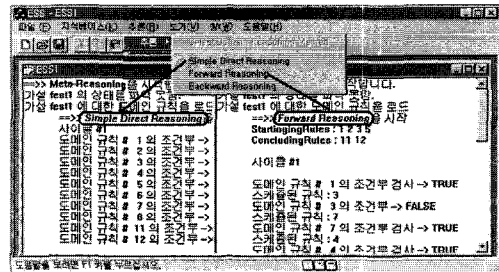


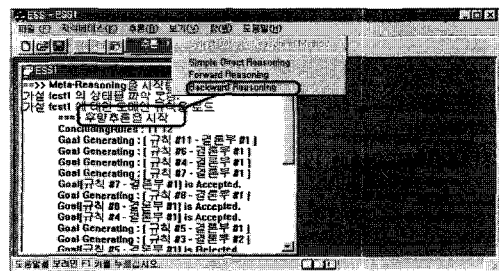
그림 15. 규칙 내용 입력
Fig. 15. Input Content of Rule.

5.2 추론실행 및 결과분석

추론 실행을 하기 전에 사용자의 선택에 의해 원하는 지식베이스를 읽어오고, 읽어온 ECBM을 갖고 적절한 추론운영방식을 선택하여 추론을 행한다. 다음은 각기 추론운영 방식에 따라 결과를 화면으로 출력하여 보여 준다. 화면으로 추론과정에 대한 정보를 나타내주는 것과 동시에 별도의 명령의 실행이 없이 자동적으로 텍스트 형태로 저장될 수 있도록 하였다. 그림 16 은 직접 추론, 전향 추론 그리고 후향 추론의 실행 예이다.



(a) 직접추론과 전향추론 실행 예



(b) 후향추론 실행 예

그림 16. 추론 실행 예
Fig. 16. Example of Execution of Reasoning.

그림 16에서 보여주듯이, 전향추론은 로드되어진 ECBM에서 시작규칙(starting rule)을 찾아 이에 대해서 추론을 수행한다. 구현된 전향추론에는 충돌되어진 규칙들을 임시 저장하는 장소로 어젠더를, 이를 해결하는 스케줄러를 사용하고 있다. 여기서 사용되는 스케줄링 방법은 선입선출법과 규칙 ID 순서가 통합된 형식을 취한다. 후향추론은 ECBM에서 결론규칙(concluding rule)을 찾아 이에 대해서 추론을 수행한다. 이 논문에서 구현한 후향추론은 추론을 수행하면서 가설들의 트리를 만들지만 이를 이용하지는 않는다. 생성된 가설 트리를 그래픽으로 보여주어 사용자가 좀 더 쉽게 이해하고 용이하게 접근할 수 있도록 인터페이스 부분에 대한 강화가 필요하며 이는 향후 보완 사항으로 남겨 두었다.

전향추론과 후향추론을 매트릭스를 이용하여 수행할 경우 매트릭스의 행과 열 중 어느 쪽을 먼저 고려의 기준으로 선정하는가에 따라 다르지만 유사한 기본적인 절차 갖는다. 매트릭스의 변형인 ECBM을 이용한 추론 역시 이러한 성격을 갖고 있어 전향추론과 후향추론의 기본적인 절차는 같다. 다만 전향추론일 경우, 연결된 규칙 검색부분과 규칙 검토부분에서 백트래킹과 충돌문제 해결 등의 다소 복잡한 알고리즘을 추가적으로 사용한다는 점이 다르다. 전향추론과 후향추론 외에도 이 논문에서 제안한 ECBM을 생성하는 과정에서 구축되어지는 관계 리스트(relational list)를 이용하여 해를 출력해 주는 단순 직접추론을 추가 구현하였다.

5.3 제안 시스템 평가

5.3.1 패턴매칭기법의 특성

효율적인 추론을 위해서 의미망, 프레임, 그리고 규칙 등으로 표현된 지식에 대해 추론 시 지식들 간의 패턴 매칭을 위해 많은 중간 표현기법이 존재한다. 패턴매칭을 위해 존재하는 이러한 기법들은 추론의 효율에 직접적인 영향을 미친다. 이 논문에서는 2차원 테이블인 CBM에 AND/OR 상태 표시 테이블을 추가하여 3차원으로 구성함으로써 추론 시에 규칙의 조건을 고려하여 완전한 추론을 할 수 있는 ECBM을 제안하였다. 이 기법은 매트릭스 표현이기 때문에 패턴매칭 속도가 기존 기법들 보다 빠르고 대다수의 도메인에 적용이 용이하다. ECBM은 매트릭스 구조로 한 지식베이스로 규칙을 적용할 수 있는 특성을 가진 실세계의 도메인이라면 대부분 수용할 수 있는 일반성을 가지고 있으나, Rete

는 트리 구조 형식으로 트리 구조를 이룰 수 있는 구조, 즉 트리 구조를 이룰 경우 효율적인 도메인 형태에만 적용할 수 있는 특수성을 갖는 기법이다.

5.3.2 추론기법의 특성

이 논문에서 채택한 추론엔진의 추론운영 방식은 사용자가 선택하여 전향추론, 후향추론이 실행될 수 있도록 하고, 지식구현 방법은 이 논문에서 새롭게 제안하는 방법으로 표준화와 모듈화를 제공해줄 뿐만 아니라 유연성 있는 방법인 규칙에 AND/OR표현을 추가·제안한 비트 매트릭스 형태로 구축한 ECBM을 이용하도록 설계 구현하였다.

패턴 매치가 Rete 알고리즘을 사용할 경우 생성되어지는 Rete망과 ECBM의 특성을 간단히 살펴보면 표 2와 같다. ECBM은 매트릭스 구조로 지식베이스로 생성 규칙을 적용할 수 있는 특성을 가진 실세계의 도메인이라면 대부분 수용할 수 있는 일반성(generalization)을 가지고 있으나, Rete는 트리 구조 형식으로 트리 구조를 이룰 수 있는 구조, 즉 트리 구조를 이룰 경우 효율적인 도메인 형태에만 적용할 수 있는 특수성(specialization)을 갖는 메카니즘이다.

표 2. Rete와 ECBM의 특성비교

Table 2. Properties Comparison of Rete with ECBM.

구조 항목	ECBM	Rete
방식	비트 매트릭스 구조	트리 구조
장점	추론운영방식에 제한이 없음 지식베이스의 검증이 용이 및 운영 용이 추론 속도가 빠름 특정도메인에 국한되지 않아 범용성을 갖음 AND/OR 연산수행	다수의 객체와 다수의 패턴 매칭에 효율적임 전향추론에 효율적임 규칙의 삽입 삭제 용이
단점	규칙의 수가 많을 경우, 분할 기법이 필요 메모리의 효율성이 낮음	메모리의 효율성이 낮음 Rete망 구성이 어려운 도메인들이 많음 추론운영방식에 제한이 있음

ECBM 생성과정의 중간단계에서 관계 리스트가 구축되며, 이를 이용하여 직접추론의 수행도 가능하다.

표 3은 ECBM을 이용한 추론엔진을 실험하기 위해 주어진 도메인으로부터 지식베이스를 구축하기 위해 규칙들을 입력하는 형태를 보여주고 있다. 여기서 사용된 규칙은 삼성 중공업의 수많은 도메인 중 13개의 센

서를 통해 들어오는 자료를 받아 진단하는 도메인을 현장 작업자에게 제공받아 표 3과 같은 형식에 맞추어 정형화를 시켜 규칙을 생성하였다.

표 3. 사용된 규칙의 예

Table 3. Example of using Rule.

Rule-name:R1	Rule-name:R2
Rule-class:Domain Rule	Rule-class:Domain Rule
Rule-id:1	Rule-id:2
IF:	IF:
Tokens: <i>Sensor1.Q</i> is TRUE	Tokens: <i>Sensor1.M</i> is TRUE
Speech: float-var is true	Speech: float-var is true
CF:0.000000	CF:0.000000
THEN:	THEN:
Tokens: <i>Sensor1.X</i> = TRUE	Tokens: <i>Sensor1.Y</i> = TRUE
Speech: float-var = true	Speech: float-var = true
CF:0.000000	CF:0.000000
Rule-priority:0	Rule-priority:0

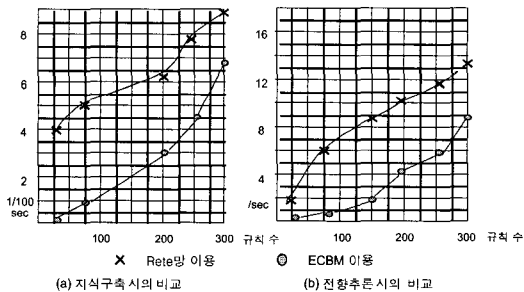


그림 17. 전향추론의 속도 비교
Fig. 17. Speed comparison of Forward Reasoning.

위 그림 17은 지식구축을 하는데 걸리는 시간과 추론에 걸리는 시간을 트리기반 구현방법인 Rete와 비교한 것이다. 추론에서는 Rete의 특성 상 전향추론만 비교를 하였다. 지식구축을 하는데 걸리는 시간을 검토하여 트리로 구축할 경우 보다 더 효율적임을 그래프로 제시하였다. 그래프를 근거로 ECBM으로 구축할 경우 구축하는데 걸리는 시간은 $O(n)$ 이며, 트리 구조로 저장할 경우 즉 구축할 경우는 $O(n \log 2n)$ 이라는 값을 갖는다.

ECBM은 전향추론뿐만 아니라, 후향추론·직접추론을 할 수 있는 융통성을 지닌 것이 장점이지만, 메모리를 많이 차지한다는 점과 아직 많은 검토를 거친 검증된 구조가 아니라는 단점이 있다.

VI. 결 론

이 논문에서 설계·구현한 시스템의 기본 구조는 지

식베이스, 지식베이스 엔진 그리고 사용자 인터페이스로 구성된다.

지식베이스 구축을 위한 표현형태로는 객체지향 데이터 타입을 기반으로 하여 가장 널리 사용되고 표현하기 편리한 규칙과 더불어 규칙응용 및 추론의 효율을 위해 가설을 결합한 표현구조를 제시하였다. 이러한 구조는 큰 지식베이스를 가설에 기반을 둔 작은 규칙 집합의 단위로 분리시킴으로서 지식공학자가 더욱 쉽게 지식베이스를 개발하고 유지 보수할 수 있도록 하였다.

지식베이스 엔진은 블랙보드와 추론엔진의 혼합형 구조로서, 기존의 블랙보드나 추론엔진 기반 시스템과는 달리 이벤트 생성기, 실행기, 스케줄러, 패턴매처로 구성된 추론엔진을 블랙보드 구조와 독립적으로 두고 상호 연계관계를 유지하는 새로운 엔진 구조를 설계, 구현하였다. 실시간에서 좋은 추론 결과를 나타내고 있는 블랙보드 구조를 기반으로 한 시스템 구조의 특징은 다중 표현된 지식베이스를 이용 가능하며, 지식기반 시스템을 다중 대리자 구조로 확장이 용이하다는 것이다.

지식베이스 구축 방법으로는 비트 매트릭스를 근간으로 하여 제안한 ECBM을 선택하였다. ECBM은 구축 속도가 트리 구조보다 빠르고, 지식베이스의 검증 및 운영이 용이하며, 대부분의 도메인에 적용이 가능하다는 보편성을 갖는다. 또한, ECBM을 이용할 경우 사용자의 선택에 의해 전향추론, 후향추론, 직접추론을 수행할 수 있다는 장점도 추가적으로 확보할 수 있었다.

또한 사용자에게 친숙한 인터페이스를 제공하기 위해서 윈도우 기반의 클래스-객체 생성기와 가설-규칙 편집기의 지식베이스 구축 보조 시스템을 개발하였으며, 추론 인터페이스를 구현하였다.

이 논문에서 구현된 시스템을 실제 업무에 적용시 비용에 비해 효과가 상승하는 파급효과를 가질 것이다. 즉, 높은 임금을 받는 전문가에 대한 일부 대체가 가능하고, 전문가에 대한 업무지원으로 보다 창의적인 업무 기능의 확장을 기대할 수 있으며, 지식베이스에 저장된 전문지식을 적극 활용할 수 있을 것이다.

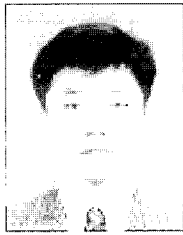
향후 연구내용으로는 구현된 시스템에 방대한 지식베이스를 유지하고 관리하기 위한 지식관리기와 추가로 지식을 검증하여 획득하고 구축하는 지식획득 모듈이나 자기학습 모듈의 추가 등의 개발이 요구된다. 또한 충돌해결 전략에 있어서 효율적인 추론을 위한 불확실성이 취급되어야 하고, 사용자에게 신뢰성 제공을 위

한 설명기능이 추가되어야 한다. 이 논문에서 구현한 지식베이스 엔진에 실시간 개념을 도입하여 실시간 시스템으로 확장하는 연구가 필요하며, 다양한 영역에 대해 이 시스템을 적용시켜야 할 필요가 있다.

참 고 문 헌

- [1] Rychener, M., "Expert System for Engineering Design", *Expert Systems*, 2(1), 1985, pp.30~44.
- [2] Miller, R.K. and T.C. Walker, "Artificial Intelligence Applications in Business Management", Prentice-Hall, 1988.
- [3] Waterman, D.A., "A Guide to Expert Systems. Reading", MA : Addison-Wesley, 1986.
- [4] Shortliffe, E.H., S.G. Axline, B.G. Buchanan, T.C. Merigan and S.N. Cohen, "An Artificial Intelligence Program to Advise Physicians Regarding Antimicrobial Therapy", *Computers and Biomedical Research*, Vol.6, 1973, pp.544~560.
- [5] Szolovits, P. and S.G. Pauker, "Categorical and Probabilistic Reasoning in Medical Diagnosis", *Artificial Intelligence*, Vol.11, 1978, pp.115~144.
- [6] Moti Schneider, Abraham Kandel, Gideon Langholz, and Gerard Chew, "Fuzzy Expert System Tools", John Wiley & Sons Ltd., 1996.
- [7] Jeong Hun Shin, Myoung Ryun Oh, Kwang Jin Oh, Yang Won Ryee, Keun Ho Ryu, and Young Hun Kim, "Design and Implementation of ECBM for Inference Engine", *The Transactions of the KOREA Information Processing Society*, Vol. 4, No 12. 1997.
- [8] Giarratano, Joseph and Gary Riley, "Expert System: principles and programming", PWS Publishing Company, 1994.
- [9] Newton S. Lee, "Programming with P-Shell", *IEEE EXPERT*, Summer 1986.
- [10] Sanja Vranes and Mladen Stanojevie, "Prolog/Rex-A way to Extend Prolog for Better Knowledge Representation", *IEEE Trans. on Knowledge and Data Engineering*, 6(1), Feb. 1994.
- [11] Calmet, J., I.A. Tjandra, and G. Bitlencourt, "MANTRA : A shell for Hybrid knowledge Representation", *IEEE*, 0-8186-2300-4191, 1991.
- [12] Frederick Hayes-Roth, "Rule-Based Systems", *Communications of the ACM*, 28(9), Sep. 1985.
- [13] Keravnou, E.T. and L. Johnson, "Competent Expert Systems : A case study in fault diagnosis", Kogan Page Ltd., 1986.
- [14] Kim, W., "Introduction to Object-Oriented Databases", MIT Press, 1990.
- [15] Robinson, D., "Object-Oriented Software System", Byte, Aug. 1981.
- [16] Schmoldt, L. Daniel and H. Michael Rauscher, "Building Knowledge-based systems for Natural Resource management", ITP, 1996.
- [17] Forgy, C.L., "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem", *Artificial Intelligence* 19, 1982, pp.17~37.
- [18] Hanson, N. Eric and Yu-Wand Wang, "A Performance Comparison of the Rete and TREAT Algorithms for Testing Database Rule Conditions", *IEEE*, 1992.
- [19] Daniel P. Miranker. "TREAT : A better match algorithm for AI production system", In Proc. AAAI National Conference on Artificial Intelligence, Aug. 1987, pp.42~47.
- [20] Frederick Hayes-Roth, Victor Lesser, "Focus of Attention in the Hearsay-II Speech Understanding Systems", *Proc. of IJCAI-77*, 1977, pp.27~35.
- [21] Nii, H.P., E. Feigenbaum, J. Anton and A.J. Rockmore, "Signal-to-Symbol Transformation : HASP/SIAP Case Study", *The AI Magazine*, 3(1), 1982, pp.23~35.
- [22] Nii, H.P., "The Blackboard Model of Problem Solving", *AI Magazine*, 7(2), 1986.
- [23] Scales, J. Daniel, "Efficient Matching Algorithms for the SOAR/OPS5 production system", *STAN-CS-86-1124*, June 1986.

저 자 소 개



田根 瑛(正會員)

1993년 : 군산대학교 전산학과 졸업. 1995년 : 충북대학교 대학원 전산학과(이학석사). 1997년 : 충북대학교 대학원 전산학과 박사과정(수료). 1997년~현재 : 군장대학 컴퓨터응용학부 조교수. <관심분야>

시간지원 데이터베이스, 전문가 시스템, 멀티미디어 데이터베이스 등



李 羊 遠(正會員)

1979년 : 송실대학교 전산학과 졸업. 1983년 : 연세대학교 산업대학원 전산전공(공학석사). 1994년 : 송실대학교 전자계산학과(공학박사). 1979년~1986년 : 국방 연구원. 1986년~현재 : 군산대학교 컴퓨터

정보과학과 교수. <관심분야> 영상 데이터베이스, 패턴 인식, 인공지능, 멀티미디어 데이터베이스 등



申 盛 允(正會員)

1994년 : 군산대학교 전산학과 졸업. 1997년 : 군산대학교 대학원 정보통신공학과(공학석사). 2000년~현재 : 군산대학교 대학원 컴퓨터과학과 박사과정. 1998년~현재 :

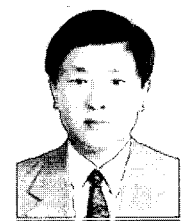
군장대학 컴퓨터응용학부 겸임교수. <관심분야> 컴퓨터 비전, 패턴인식, 인공지능, 멀티미디어 데이터베이스 등



柳 根 鎬(正會員)

1976년 : 송실대학교 전산학과 졸업. 1980년 : 연세대학교 산업대학원 전산전공(공학석사). 1988년 : 연세대학교 대학원 전산전공(공학박사). 1976년~1986년 : 육군군수 지원사전산실(ROTC장교), 한국전

자통신연구소(연구원), 한국방송통신대 전산학과(조교수) 근무. 1989년~1991년 : Univ. of Arizona 연구원 (TemplS Project). 1986년~현재 : 충북대학교 전기전자 컴퓨터공학부 교수. <관심분야> 시간데이터베이스, 시공간데이터베이스, 객체 및 지식기반 시스템, Temporal GIS, 지식기반 정보검색시스템, 데이터 마이닝, 데이터베이스 보안 및 Bio-Informatics



申 定 薰(正會員)

1982년 : 송실대학교 전산학과 졸업. 1991년 : 충북대학교 대학원 전산학과(이학석사). 1999년 : 충북대학교 대학원 전산학과(이학박사). 1992년~현재 : 익산대학 컴퓨터

과학과 교수. <관심분야> 데이터베이스, 정보검색, 지식기반시스템 등