

工學碩士 學位論文

비상발전기용 디젤엔진을 위한
진단기술에 관한 연구

A Study on Diagnosis Technology of Diesel Engine
for Emergency Generator

指導教授 柳 吉 洙

2005年 7月

韓國海洋大學校 大學院

컴퓨터工學科

朴 鍾 一

本 論 文 을 朴 鍾 一 의 工 學 碩 士 學 位 論 文 으 로 認 准 함

委 員 長 工 學 博 士 辛 沃 根 印

委 員 工 學 博 士 金 載 熏 印

委 員 工 學 博 士 柳 吉 洙 印

2005 年 7 月

韓 國 海 洋 大 學 校 大 學 院

컴퓨터 工 學 科 朴 鍾 一

목 차

Abstract	iv
제 1 장 서 론	1
제 2 장 전문가시스템을 이용한 디젤엔진 진단 시스템 구축	4
2.1 일반적인 전문가시스템의 구조	5
2.2 진단대상 선정	6
2.3 진단대상에 대한 진단지식 추출	7
2.4 진단지식 표현	8
제 3 장 JRules를 이용한 지식베이스 구축	16
3.1 전문가시스템 개발도구 JRules	16
3.2 디젤엔진의 지식베이스 구축	17
제 4 장 시스템 구현	24
4.1 시스템 전체 구성 및 기능	24
4.2 In-Touch 기반의 계측모듈 및 감시모듈	26
4.2.1 감시모듈의 GUI 화면 구성	28
4.3 고장진단 모듈	31
4.4 예측진단 모듈	32
4.5 진단모듈과 감시 모듈간의 인터페이스	33
4.6 시운전 데이터 분석	34
제 5 장 실 행	36
제 6 장 결 론	41
참고 문헌	42

그림 목차

그림 2.1 진단시스템 구축 단계	4
그림 2.2 일반적인 전문가시스템 구조	6
그림 2.3 디젤엔진 계통도	7
그림 2.4 전문가에 의하여 추출된 저온 냉각수 계통의 진단지식	8
그림 2.5 계통별 증상 및 상태 간의 관계	11
그림 2.6 각 계통별 기기간의 객체지향 표기법에 의한 관계	12
그림 2.7 진단데이터의 증상 추정 알고리즘	14
그림 3.1 JRules의 역할 및 기능	17
그림 3.2 지식베이스 구축 단계	18
그림 3.3 디플로이어 설정	19
그림 3.4 BOM에서 클래스 추가	20
그림 3.5 TRL로 작성한 룰 화면	21
그림 3.6 초기동작 설정화면	22
그림 3.7 규칙흐름도 작성화면	22
그림 3.8 룰 엔진 생성화면	23
그림 4.1 시스템 구성도	26
그림 4.2 In-Touch에서의 태그 설정화면	27
그림 4.3 배기가스 계통의 GUI 화면	30
그림 4.4 고장 진단의 흐름도	31
그림 4.5 예측 진단의 흐름도	32
그림 4.6 진단모듈과 감시모듈간의 인터페이스	34
그림 5.1 고장진단 실행 흐름도	37
그림 5.2 알람값으로 변경하기 위한 화면	38
그림 5.3 상태감시 및 고장진단 결과	38
그림 5.4 예측진단 실행 흐름도	39
그림 5.5 상태감시 및 예측진단 결과	40

표 목차

표 2.1 디젤엔진의 6계통 클래스화	10
표 4.1 INSPECT 데이터베이스	35

A Study on Diagnosis Technology of Diesel Engine for Emergency Generator

Jong-Il Park

Department of Computer Engineering, Graduate School,
Korea Maritime University, Busan, Korea

Abstract

In this dissertation, we describe an implementation of a diagnosis expert system for emergency diesel electric generators in nuclear plant. To construct the diagnosis expert system, the classification and the analysis on causal relationship in diagnosis domain is conducted in the first place. Based on the knowledge on the causal relationship of a human expert in the domain, knowledge base is constructed using JRules which is an object-oriented and rule-based expert system development tool.

The system consists of a diagnosis module, a measuring module and a monitoring module. The diagnosis module is composed of a fault-diagnosis submodule which is activated by alarm occurrences and an early-diagnosis submodule which analyses measured data in a periodical manner. The measuring module measures data necessary for diagnosis from the diesel engine. Finally, the monitoring module

provides graphic user interface. In this study, the measuring module is simulated by using data of a trial run obtained from DMDS, and the monitoring module is implemented by making use of a SCADA system, In-Touch.

To test the operation of the whole system, a set of simulated data is fed into the measuring module. The correct operation of the system could be confirmed by observing induced diagnosis results through the monitoring module.

제 1 장 서 론

원자력 발전소에서 정전과 같은 비상사태가 발생하는 경우 비상디젤발전기(EDG : Emergency Diesel Generator)는 원자로의 잔열을 제거하는 등 발전소 안전계통 주요 기기에 비상전원을 공급하기 위해 10초 내에 긴급 가동되고 1분 이내에 부하를 제공하여야 한다. 그러나 국내의 원자력 발전소의 비상전원계통으로 이용되고 있는 비상디젤발전기는 사용빈도가 적은 이유로 그 중요성이 실제보다 인식되지 못하고 있다[1].

원전에서 비상디젤발전기의 중요성은 PSA(Probabilistic Safety Assessment) 안전성평가에서 이미 확인되었을 뿐만 아니라, 미국에서는 이미 원전 경년 열화연구(Nuclear Plant Aging Research) 프로젝트의 일환으로 비상디젤발전기계통의 경년 열화(Aging)에 관련된 연구가 활발히 진행되고 있다[2][3]. 그 결과 산업계와 규제기관간의 성능 평가개념이 통계적인 시험결과에 대한 의존에서 경년열화/신뢰도 감시에 활용하기 위한 실시간 성능감시 및 경향분석으로 전환되고 있다.

국내에서는 최근에서야 비상발전기 계통에 대한 열화특성 및 신뢰도 등에 관심을 가지기 시작하고 있다[2]. 그러나 목적이 비상용이므로 상시 감시에 소홀해지기 쉬우며 고도의 전문지식을 가진 전문가를 채용하는 것에도 어려움이 따르게 된다. 그러므로 보다 체계적이고 종합적인 진단 분석 기술을 확보하고 효율적인 진단시스템을 활용하여 엔진 고장 또는 비정상 운전시에 당직자에게 시스템이 적절한 충고를 제공해줌으로써 고도의 전문지식을 가지지 못한 엔지니어가 근무 중이라도 위급한 상황에 쉽게 대처할 수 있도록 하는 것이 바람직하다. 이를 위해서는 비상디젤발전기에 대한 지능적인 진단기능을 가진 시스템 개발이 필수적이다.

진단형 전문가시스템은 발생 가능성이 있는 이상상태, 계측항목간의 인

과관계 및 관찰된 시스템의 동작에 대한 정보로부터 이상의 원인을 찾아내는 시스템을 말한다[4]. 적용분야에 따라 의료진단시스템, 고장진단시스템 등으로 분류할 수 있다. 고장진단시스템은 시스템의 운행에 있어서 극도의 안전성과 고장에 대한 빠른 조치를 요구하는 원자력 또는 화력 발전소의 온라인 고장진단시스템[6][7], 통신망 고장진단시스템, 전력계통 고장진단시스템 등에 많이 적용되어 오고 있다.

진단형 전문가시스템을 구축하기 위해서는 먼저 대상 분야에 적합한 지식표현법을 선정하여 지식베이스를 구축하여야 하며, 또한 이에 합당한 추론기구를 갖추어야 한다[8]. 지식표현법으로는 생성규칙(Production Rule), 의미망(Semantic Network), 프레임(Frame), 객체(Object) 등이 있다. 실제로 시스템을 구축하는 경우에는 이것들 중 어느 하나 또는 복합적인 형태를 이용하지만, 시스템이 복잡해질수록 복합적인 지식표현법을 이용하는 경향이 있다[9].

이상의 배경으로부터 본 논문에서는 고장진단형 전문가시스템 기법을 이용하여 원전의 비상디젤발전기의 감시 및 진단시스템 구축에 관하여 논한다. 이를 위해 전문가시스템 개발 도구 JRules와 SCADA(Supervisory Control And Data Acquisition) 시스템 In-Touch를 이용한다. 먼저 진단 모듈을 구축하기 위해서는 진단대상에 대한 지식베이스와 추론기구 구축이 필수적이며 전문가시스템 개발 도구를 이용하는 경우 지식베이스만 구축하면 된다. 지식베이스를 구축하기 위해서는 먼저 진단대상을 선정하고, 전문가의 협의에 의해서 진단지식을 추출하여야 한다. 추출한 진단지식을 컴퓨터가 인식할 수 있는 진단지식으로 변환하기 위해 진단대상의 클래스화 및 인과관계를 정의한 후, 최종 전문가시스템 개발 도구 JRules에서 제공하는 지식표현법으로 변경하여 지식베이스를 구축한다. 본 논문에서는 지식표현법으로 객체지향기법과 규칙기반 지식표현법을 이용한다.

지식베이스가 구축되면 감시모듈과 진단모듈을 연결함으로써 전체시스템이 완성된다. 계측모듈은 원전의 감시시스템 DMDS(Diagnosis Monitoring and Display System)의 시운전 데이터를 실시간 샘플링 데이터로 활용하고, 감시모듈은 In-Touch를 이용하여 GUI 화면을 구성한다. 또한 진단모듈과 감시모듈을 연결하기 위해 Java 언어를 이용하여 DDE(Dynamic Data Exchange) 통신 인터페이스를 작성한다.

본 논문의 구성은 다음과 같다. 2장에서는 전문가시스템을 이용한 디젤엔진 진단시스템 구축에 관하여 논하고, 3장에서는 JRules를 이용한 지식베이스 구축에 관하여 설명한다. 4장에서는 시스템의 구현에 관하여 설명하고, 제 5장에서는 구현한 시스템의 실행결과를 예시한다. 마지막으로 6장에서는 결론 및 향후의 연구 방향을 제시한다.

제 2 장 전문가시스템을 이용한 디젤엔진 진단시스템 구축

전문가시스템을 이용하여 디젤엔진의 진단시스템을 구축하기 위해서는 그림 2.1과 같은 흐름으로 진단시스템을 구축하게 된다. 먼저 진단대상을 선정 한 후, 진단대상의 전문가와 협의에 의해서 진단지식을 추출한다. 전문가로부터 추출된 진단지식을 컴퓨터가 인식할 수 있도록 표현하기 위해서는 우선 진단대상에 대한 클래스화와 인과관계를 정의하여야 한다. 인과관계가 정의되면 전문가시스템 개발도구를 이용하여 진단지식베이스를 구축하고, 최종적으로 감시모듈과 진단모듈을 연결함으로써 시스템 구성을 마친다. 본 장에서는 진단시스템을 구축하기 위해서 활용한 전문가시스템과 구축단계에 대해서 상세하게 살펴본다.

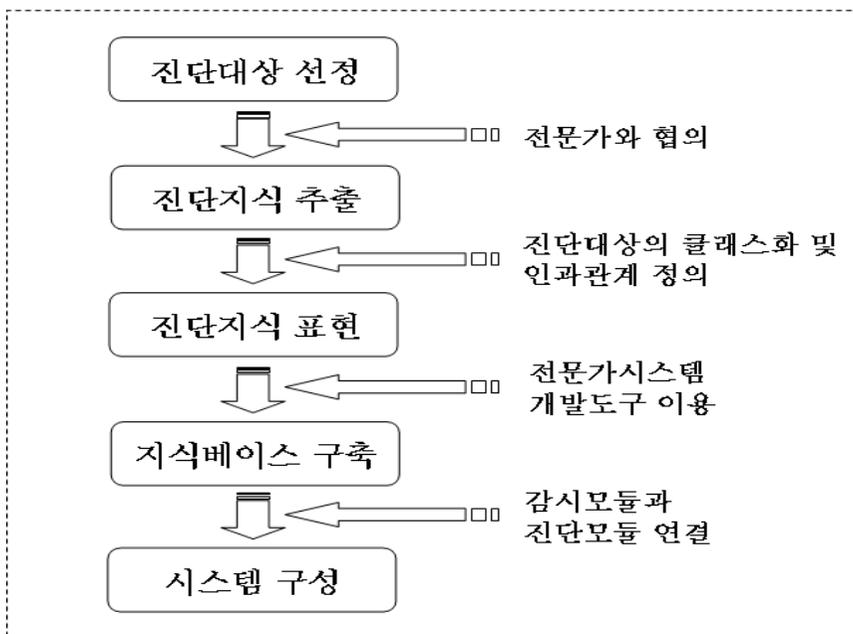


그림 2.1 진단시스템 구축단계

2.1 일반적인 전문가시스템의 구조

전문가시스템이란 특정분야의 인간 전문가들이 전문 지식을 수집 정리하여 주어진 특정 전문영역에 관한 문제를 컴퓨터의 추론능력을 이용하여 해결하는 컴퓨터 의사결정 및 자문시스템을 말한다[9]. 전문가시스템은 각종 장비의 고장진단, 의사결정 지원(Decision-making), 인사관리, 물품 재고관리, 통신 네트워크 고장진단, 의료진단, 화학 분자식으로부터 분자구조 추출, 지질 정보를 가지고 광의 종류와 위치를 결정하는 등 많은 분야에서 도입하고 있다.

그림 2.2와 같이 일반적인 전문가시스템의 구조는 지식베이스(Knowledge Base), 작업메모리(Working Memory), 추론엔진(Inference Engine), 사용자 인터페이스(User Interface)로 구성된다.

(1) 지식베이스

지식베이스는 문제풀이의 규칙, 절차, 그리고 문제 영역과 관련된 고유한 사실(Facts) 등의 자료로 구성된다.

(2) 작업메모리

작업메모리는 사용자가 해결 하고자 하는 문제 또는 진위여부를 검토하기 위한 자료 등을 저장한다. 그리고 추론엔진이 지식베이스에서 가져온 규칙을 사용자가 입력한 자료에 적용하여 나온 결과를 저장하기도 한다.

(3) 추론엔진

전문가시스템의 핵심은 제어 구조(Control Architecture) 또는 규칙해석기(Rule Interpreter)라고도 알려진 추론엔진인데, 추론엔진은 결론의 도출과 형성을 위한 방법론을 제공하는 컴퓨터 프로그램을 말한다. 추론엔진은 사용자 인터페이스를 통하여 사용자로부터 제공된 입력을 받아들이며, 의사결정이나 진단과 같은 문제해결안을 생성하기 위하여 지식베이스내의 지식을 결합한다.

(4) 사용자 인터페이스

사용자와 전문가시스템 간의 의사소통을 돕는 도구이고, 사용자로부터 사실 또는 질의를 입력 받거나 추론 결과를 출력하는 기능을 가진다.

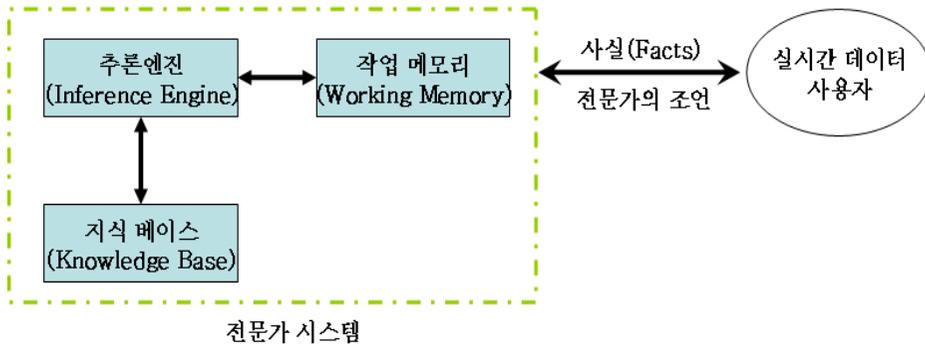


그림 2.2 일반적인 전문가시스템 구조

2.2 진단대상 선정

본 논문에서 진단대상은 원전의 비상발전기 디젤엔진 Pilestick PC2-5V를 대상으로 하고, 울진 5, 6호기의 감시시스템 DMDS에 의해서 감시되는 인자를 대상으로 한다. 그림 2.3과 같이 디젤엔진을 다음의 6계통으로 분류하였다. 분류한 6계통을 기반으로 디젤엔진을 클래스화하고 인과관계를 정의하였다.

- 저온냉각수 계통(LT Water Circuit)
- 고온냉각수 계통(HT Fresh Water Circuit)
- 윤활유 계통(Lubricating Oil Circuit)
- 연료유 계통(Fuel Oil Circuit)
- 소기 계통(Scavenger Air Circuit)
- 배기가스 계통(Exhaust Gas Circuit)

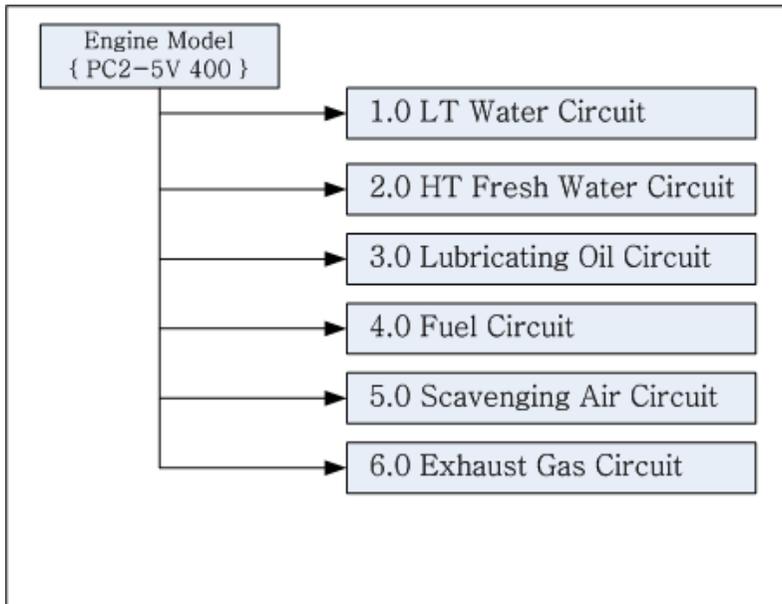


그림 2.3 디젤엔진 계통도

2.3 진단대상에 대한 진단지식 추출

진단대상이 선정되면 진단대상에 대한 진단지식을 추출하여야 한다. 진단지식을 추출하기 위해서는 전문가의 경험적 지식과 자료수집 및 분석이 필요하다. 본 논문에서는 진단대상 전문가의 도움을 받아 각 계통별 증상, 진단, 조치에 대한 자료들을 진단지식으로 작성하였다. 그림 2.4에서 저온 냉각수 계통(LT Water Circuit)의 증상에 관한 진단 및 조치의 흐름을 예시로 보면, 에어컨의 출구 CCW 냉각수 온도가 상승한다는 증상이 발생했을 때 진단결과와 전파되는 인과관계의 흐름을 보여주고 있다.

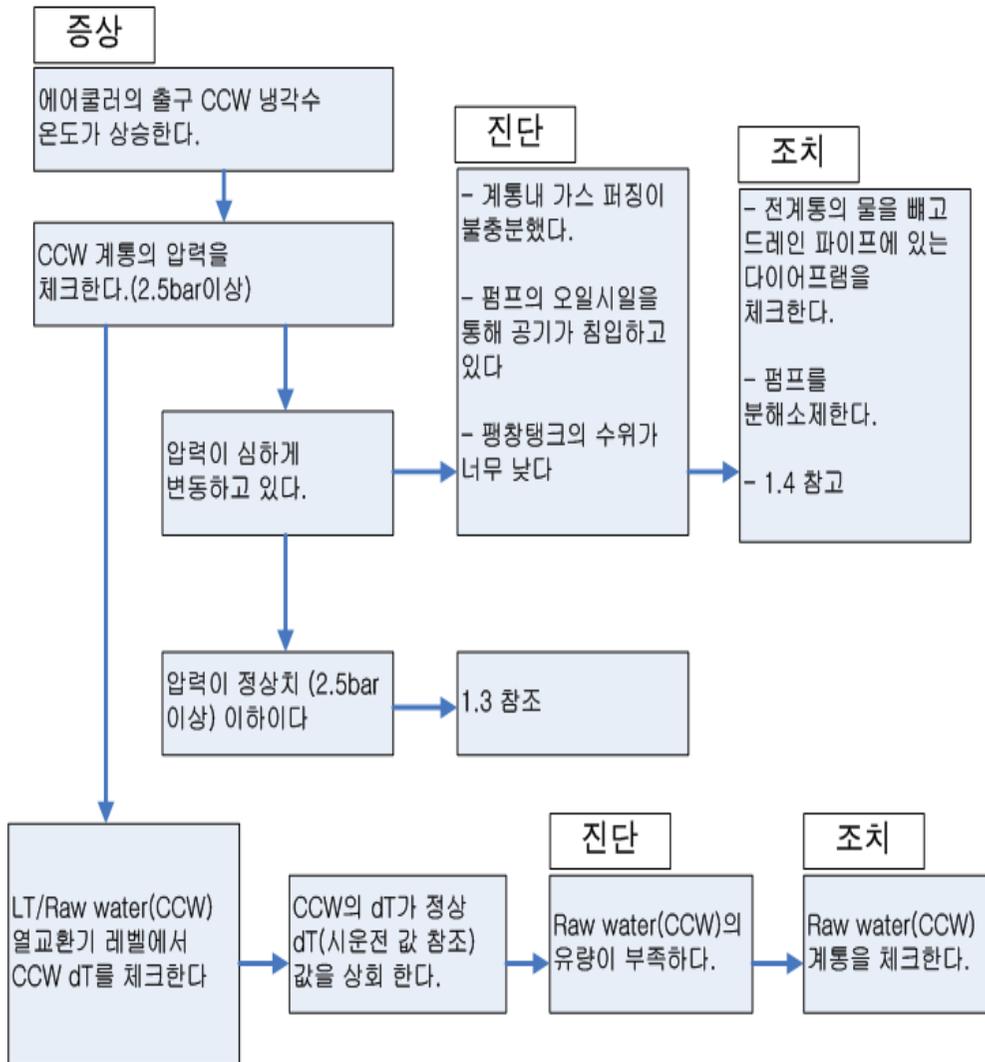


그림 2.4 전문가에 의하여 추출된 저온 냉각수 계통의 진단지식

2.4 진단지식 표현

전문가에 의해서 추출된 진단지식은 컴퓨터가 이용할 수 있는 진단지식으로 변경되어야 한다. 진단지식을 변경하기 위해서는 먼저 디젤엔진을 계층별로 분류하고 이것들의 속성 및 값을 체계적으로 표시할 필요가 있다. 본 논문에서는 객체지향기법을 이용하여 Java 언어로 디젤엔진을 클

래스화한다. 디젤엔진 6개의 계통 각각을 Class명, 각종 센서들을 Attribute로, 센서값을 Value로 정의한다. 표 2.1은 대상 디젤엔진에 대한 Class명, Attribute명, Value Type, Value로 분류하였다. 표에서 배기가스(EG System) 클래스의 속성은 ‘Cylinder Exhaust Gas Temperature’와 ‘Turbine Inlet Gas Temperature’, ‘Turbine Outlet Gas Temperature’를 나타내고 있고, Value는 아날로그 값을 가지고 있으므로 상태값으로는 High 또는 Low 값을 가지고, 증상값으로는 Increase, Decrease, Steady값 중 하나를 가진다.

표 2.1 디젤엔진의 6계통 클래스화

Class	Attribute	Value Type	Value	
			상태	증상
EG System	Cylinder Exhaust Gas (A1~A8 , B1~B8)	Temperature	High Low	Increase, Steady, Decrease
	Turbine Inlet Gas			
	Turbine Outlet Gas			
SA System	Engine Inlet Air Pressure (A , B)	Pressure	High Low	Increase, Steady, Decrease
	Engine Inlet Air Temperature (A , B)	Temperature		
LO System	Engine Inlet LO Pressure	Pressure	High Low	
	Bearing Temperature (N1~N9) Thrust Bearing Temperature (N1~N9) Generator Bearing Temperature Engine Inlet LO Temperature Thrust Bearing Temperature OI Thrust Bearing Temperature OE	Temperature	High Low	Increase, Decrease, Steady
	LO Level			
	Crankcase Gas Pressure	Pressure	Low	
	HT Water System	Engine Outlet HT Water Temperature	Temperature	Low High
HT Water Tank Level		Level	Low	
HT Water Pressure		Pressure	High	
LT Water System	Air Cooler CC Water Temperature	Temperature	Low High	Increase, Steady, Decrease
	LT Water Pressure	Pressure	High	
	LT Water Tank Level	Level	Low High	
FO System	DO Daily Tank Fuel Leakage Level (A, B)	Level	Low Very Low	
	DO Storage Tank Level	Level	Low High	
	DO Pressure	Pressure	Low	
	DO Strainer Differential Pressure	Pressure	High	

진단대상의 클래스가 정의되면 전문가가 정의한 진단지식으로부터 계통별 증상 및 상태 간의 관계를 도출하여야 한다. 그림 2.5는 배기계통에 배기밸브이상이라는 증상이 발생했을 때, 배기가스 온도가 High라는 진단데이터를 추출하여 다른 계통과의 인과관계를 도출하는 예를 보여주고 있다.

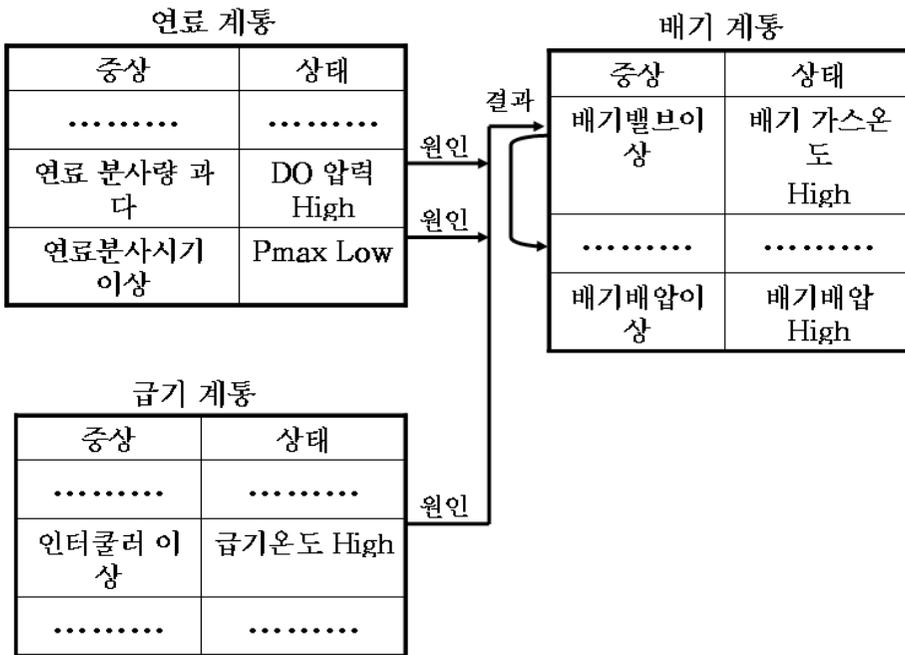


그림 2.5 계통별 증상 및 상태 간의 관계

이러한 관계를 도출하기 위해서는 일반적으로 센서값으로부터 설정 레벨에서의 상태값을 추출하고, 시계열 데이터 관계에 의해서 증상을 추출한다. 그리고 증상과 상태간의 관계를 설정함으로써 계통별 인과관계를 정의할 수 있다. 그림 2.6은 객체지향 기법으로 분류한 각 계통별 기기 간의 인과관계를 정의하기 위한 방법을 보여주고 있다.

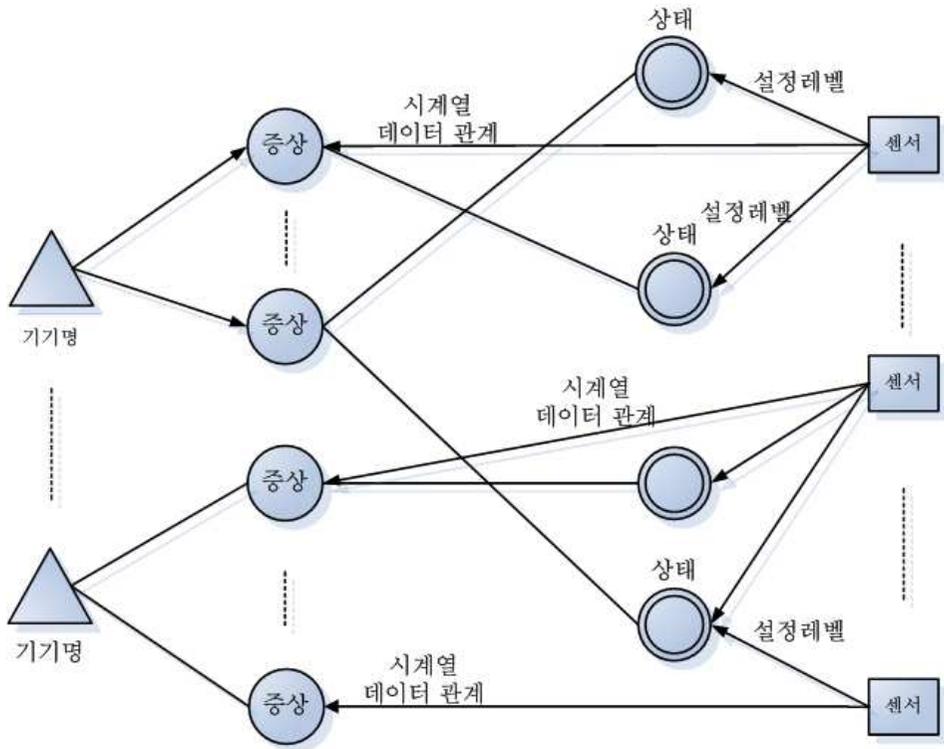


그림 2.6 각 계통별 기기간의 객체지향 표기법에 의한 관계

실시간으로 계측되는 데이터들은 아날로그와 디지털로 이루어져 있으며, 이 값들을 진단에 직접 이용하기 위해서는 진단지식이 복잡해지게 된다. 그러므로 이 값으로 센서의 상태와 시계열에 의한 경향을 파악하게 함으로써 진단지식을 체계화 할 수 있고 진단시간도 단축할 수 있다. 본 논문에서는 상태를 VL(Very Low), L(Low), N(Normal), H(High), VH(Very High)의 5가지 상태로 분류하고, 증상을 시계열데이터에 의해서 Increase, Steady, Decrease의 3가지로 분류한다.

상태값은 실시간 데이터의 순간 데이터로부터 얻어지지만, 증상값은 순차적인 시계열 데이터를 분석하여 현재 데이터 값이 증가, 감소, 또는 일정

한가를 판단할 필요가 있다. 본 논문에서는 이러한 판단을 위해서 증상 추정 알고리즘을 진단부에서 판단하게 프로그램을 하였다. 증상 추정 알고리즘은 그림 2.7과 같이 배기가스 계통의 A1 실린더를 가지고 예를 들어 보여준다. 여기에서는 배기가스 계통 A1 실린더의 현재 온도를 TE200_C 변수에 저장하고 시계열 데이터에 의한 다음 실린더 온도를 TE200_1 변수에 저장한 후 TE200_C와 TE200_1을 비교하여 Trend1 변수의 값을 설정하게 되고, 그 다음 실린더의 온도를 TE200_2 변수에 저장한 후 TE200_1과 비교하여 Trend2 변수 값을 설정하게 된다. Trend1, Trend2 변수값이 설정 레벨의 값에 만족하게 되면 증상값을 추출한다. 이 사이클을 지속적으로 반복하면서 증상값을 추출한다.

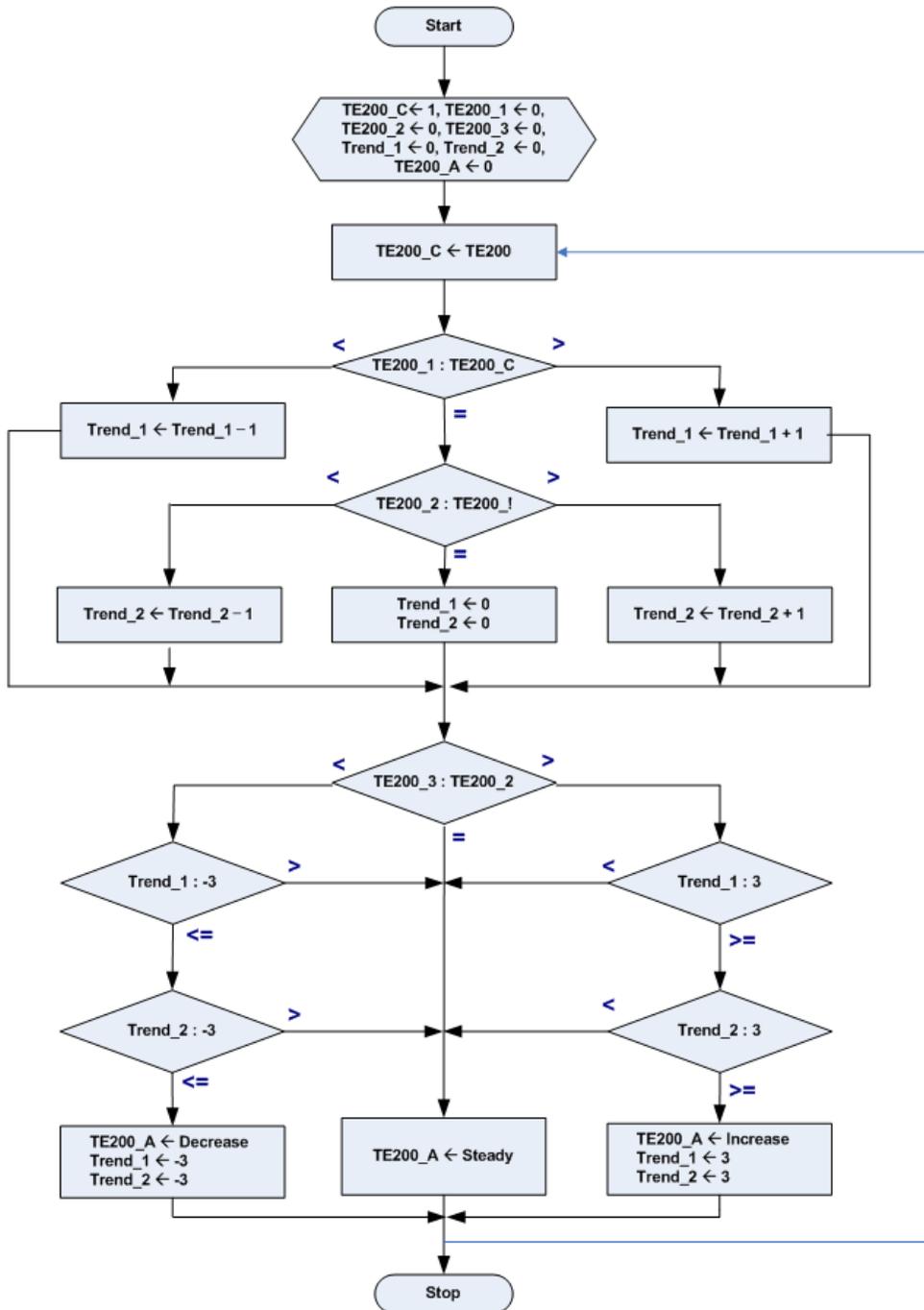


그림 2.7 진단데이터의 증상 추정 알고리즘

이상에서 정의된 인과관계를 가지고 컴퓨터에서 이용 가능한 진단 지식으로 표현하기 위해서, 본 논문에서는 전문가시스템 개발도구 JRules를 이용한 규칙기반 진단지식으로 변경하여 지식베이스를 구축 한다. 제 3장에서는 전문가시스템 개발도구 JRules를 이용하여 지식베이스를 구축하는 방법에 관하여 논한다.

제 3 장 JRules를 이용한 지식베이스 구축

본 장에서는 전문가시스템 개발도구 JRules를 이용하여 지식베이스를 구축한다. 지식베이스를 구축하기 위해서 대상 디젤엔진을 Java 언어로 클래스화하고, 앞장에서 전문가에 의해서 추출한 진단지식과 정의된 인과 관계를 이용하여 지식베이스를 구축한다. 먼저 전문가시스템 개발도구인 JRules에 대해서 살펴보고, JRules 이용해서 디젤엔진의 지식베이스를 구축하는 것에 관하여 논한다.

3.1 전문가시스템 개발도구 JRules

진단시스템을 구축하기 위해 ILOG사의 JRules라는 전문가시스템 개발 도구로 사용하였다. 이 도구는 순수 자바 코드로 구현되어 있어서 자바 애플릿 혹은 어플리케이션과 연동하여 손쉽게 전문가시스템을 구현할 수 있다. 규칙(Rules)들의 빠른 매칭이 가능하도록 Rete 알고리즘을 사용하고 있어서 실행시간을 단축시킬 수 있으므로 실시간 시스템에 적합하다고 할 수 있다. 또한 BRE(Business Rule Engine) 통합개발 환경을 제공하고 있어서 룰 엔진(Rule Engine), 룰 에디터(Rule Editor), 디버거(Debugger), 프로파일러(Profiler) 등을 지원한다[11][16]. 따라서 JRules에서 제공하는 룰 에디터, 디버거 등의 윈도우 환경을 이용하여 빠르게 개발할 수 있으며, 기존의 시스템에 쉽게 결합시킬 수 있다. JRules의 룰 엔진의 구조는 추론엔진, 룰셋(Rule Set), 아젠다(Agenda), 작업메모리로 구성되어 있다.

그림 3.1은 JRules의 역할 및 기능을 보여주고 있다. 사용자는 룰 빌더(Rule Builder)에서 진단 대상에 대한 진단지식을 작성하게 되고 레파지토리(Repository)에 저장하게 된다. 저장된 룰과 규칙흐름도(RuleFlow)를

Java 기반의 어플리케이션에서 이용가능 하도록 배포하게 되면 룰셋이 생성된다. 외부 어플리케이션에서는 JRules의 룰 엔진과 룰셋만을 이용하며 현재 상태값이나 증상값을 작업메모리에 저장하면 룰 엔진이 동작하게 된다. 룰 엔진에서는 작업메모리의 사실들에 기반으로 룰셋의 적절한 진단지식을 이용하여 실행한 결과를 작업메모리에 추가한다. 이러한 실행-추가 사이클을 반복한 후 최종적으로 추론한 진단결과를 사용자에게 알려주게 된다.

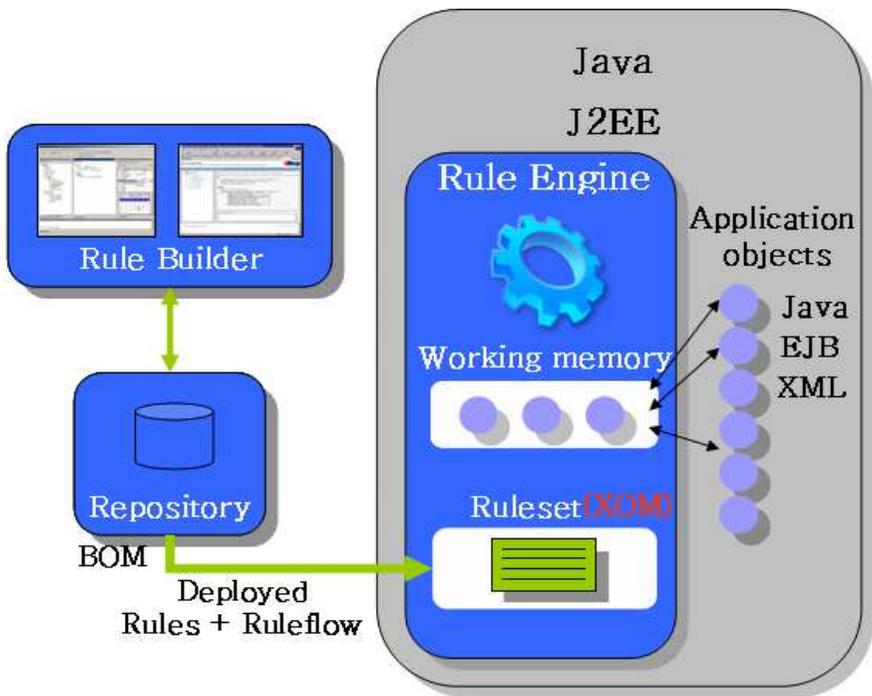


그림 3.1 JRules의 역할 및 기능

3.2 디젤엔진의 지식베이스 구축

디젤엔진에 대한 전문가의 진단지식과 정의된 인과관계를 가지고, JRules를 이용하여 규칙형 진단지식으로 변환하여 지식베이스를 구축한

다. 이 경우 그림 3.2와 같은 절차에 의해 지식베이스를 구축한다.

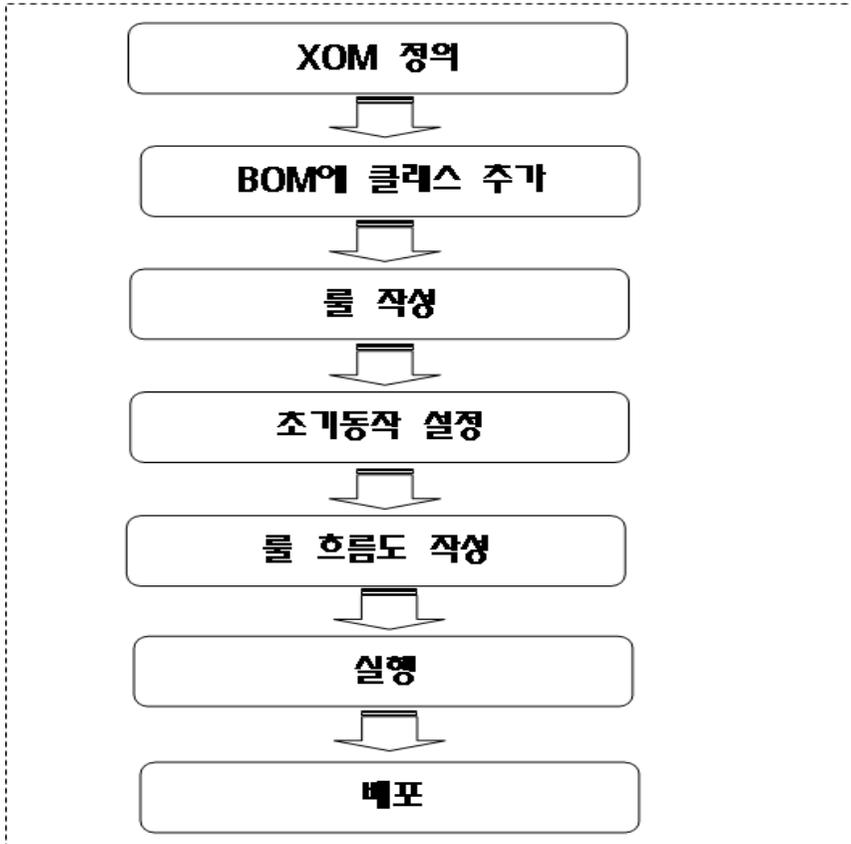


그림 3.2 지식베이스 구축 단계

(1) XOM(eXecution Object Model) 정의

XOM이란 Java 클래스나 XML 스키마 혹은 웹 서비스와 같은 구현하고자 하는 다양한 오브젝트들을 레퍼런스하는 모델이다. 레파지토리에 저장하고 나면 룰 빌더는 레파지토리 위치에 상응하는 가상 드라이버를 생성하게 된다. XOM을 정의했다면 존재하는 레파지토리를 오픈했을 때 자동으로 XOM 클래스 패스를 찾게 된다. 그림 3.3은 레파지토리에 사용자

클래스 파일의 클래스 패스를 설정하는 디플로이어 설정(Deployer Settings) 화면이다. 사용자 클래스 패스를 설정하면 JRules와 외부 Java 파일이 연동하게 된다.

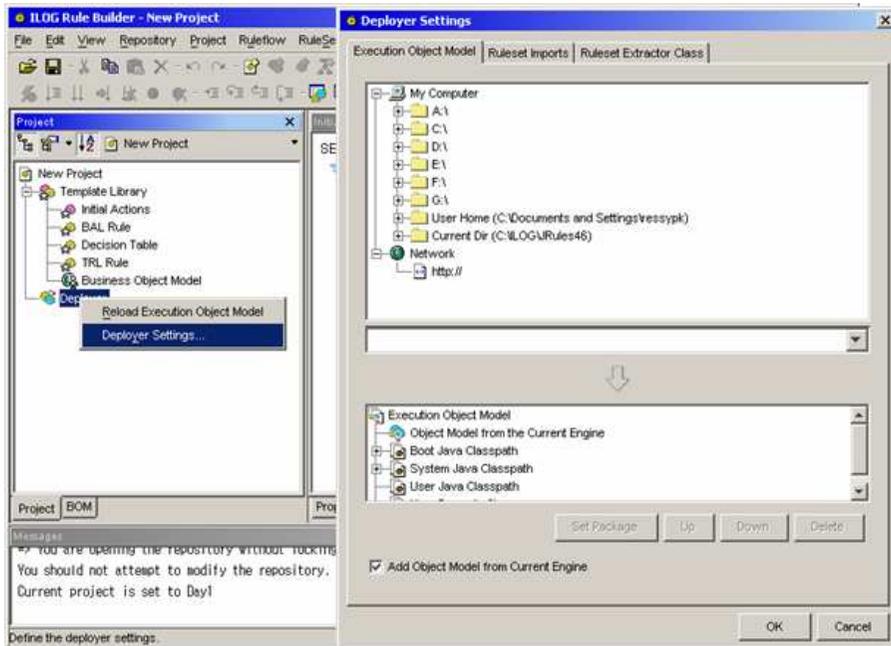


그림 3.3 디플로이어 설정

(2) BOM(Business Object Model)에 클래스 추가

BOM은 프로젝트가 생성되면 자동으로 빈 BOM을 생성하게 된다. BOM은 레파지토리에서 템플릿 라이브러리(Template Library)의 한 부분이다. 템플릿 라이브러리에서의 BOM은 정의한 클래스들을 가지고 룰 작성이나 결정 테이블(Decision Table)을 만들 때 사용된다. XOM을 정의했다면 정의한 클래스들을 임포트하여 룰 빌더에서 사용할 수 있다. 그림 3.4는 사용자 클래스 파일들을 임포트하는 화면이다. 임포트하게 되면 그림 3.4와 같이 BOM 클래스 파일들이 추가된다. 그리고 클래스 파일이 변

경되었다면 Update Class 버튼을 선택하여 변경된 부분을 수정한다.

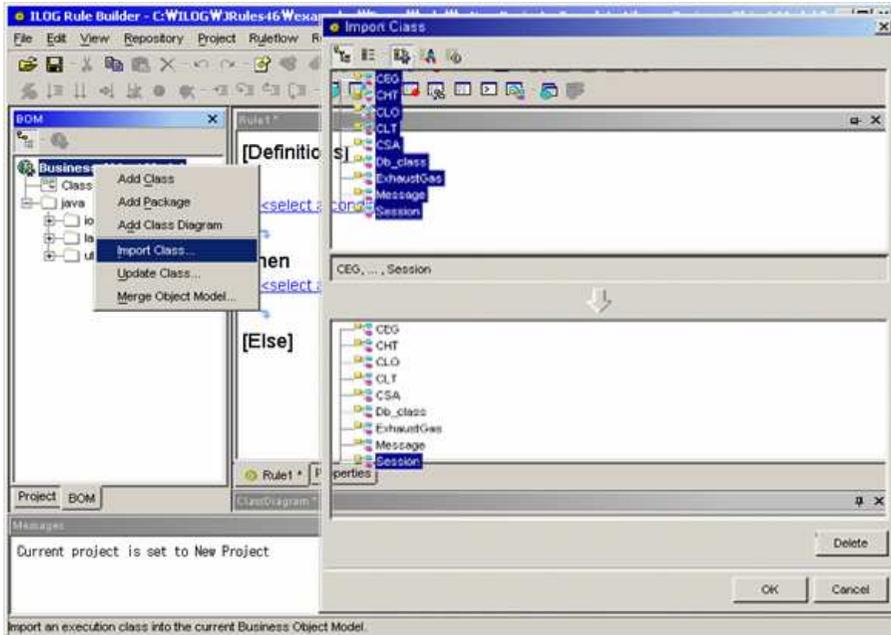


그림 3.4 BOM에서 클래스 추가

(3) 룰 작성

패키지를 생성한 후 진단에 필요한 룰을 생성할 수 있다. JRules에서는 3가지의 룰 작성 방법, 즉 IRL(Ilog Rule Language), BAL(Business Action Language), TRL(Technical Rule Language)을 이용할 수 있다. IRL은 실시간 규칙작성 언어로서 룰 엔진이 실행될 때 이용한다. TRL은 개발자를 위한 룰 작성 언어이며 프로그래밍에 능숙한 자를 위한 언어이다. BAL은 비즈니스 규칙 작성자를 위한 룰 작성 언어이며 규칙작성이 간단한 반면 제한적이기도 하다. 그림 3.5는 TRL로 작성한 룰이다. 조건절과 결론절로 나누어져 있다. 룰을 실행했을 때 조건절이 사실이라는 것이 확인되면 결론절에서 이벤트가 발생하게 된다.

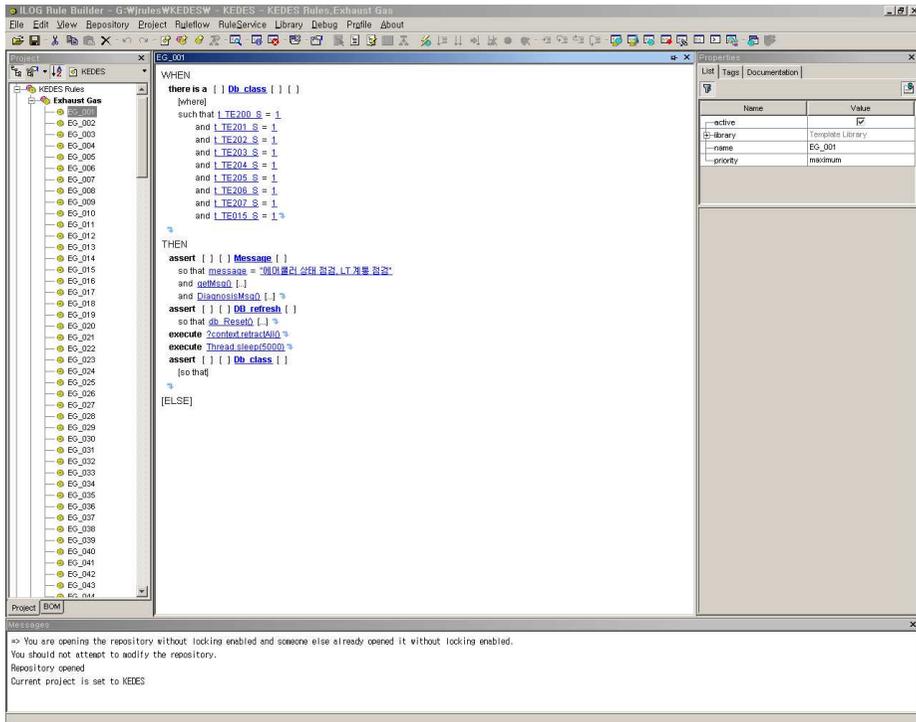


그림 3.5 TRL로 작성한 룰 화면

(4) 초기동작(Initial Action) 설정

룰 빌더 실행 시에 Assert, Retract, Update, Modify, Apply 등의 함수들을 이용하여 초기동작 값을 설정할 수 있다. Assert는 생성자에 따라 객체를 생성하고 초기화한다. 초기화하게 되면 작업메모리에 객체를 생성하고 초기화 된다. Retract는 작업메모리에서 객체를 제거할 때 사용한다. Update는 작업메모리에 존재하는 객체를 변화시키지 않고 단지 변화되었음을 룰 엔진에게 알림으로써 재평가를 수행한다. Modify는 작업메모리에 존재하는 객체를 변화시키고 룰 엔진에게 변화된 사실을 알림으로써 재평가를 수행한다. 변화된 사실이 룰 엔진에게 알려지지 않을 경우, 재평가가 이루어지지 않는다. Apply는 룰 엔진에게 알리지 않고 각 절을 재평가할 때 사용한다. 그림 3.6은 초기동작을 설정한 화면이다.

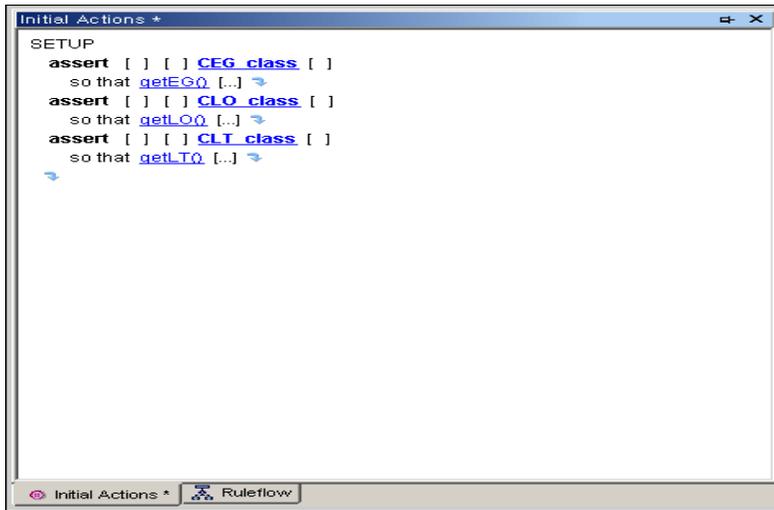


그림 3.6 초기동작 설정화면

(5) 규칙흐름도 작성

규칙흐름도에서는 룰 빌더에서 작성한 룰을 실행 순서를 정의할 수 있다. 그림 3.7은 규칙흐름도를 작성한 화면이다.

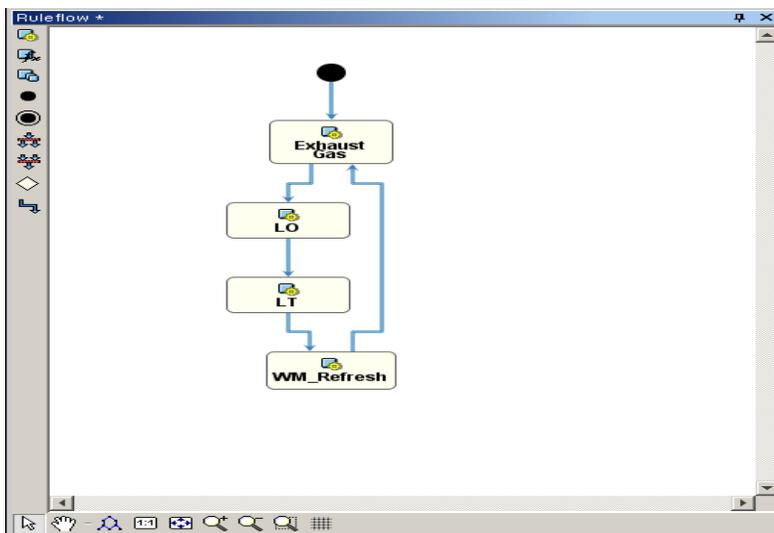


그림 3.7 규칙흐름도 작성화면

(6) JRules 상에서의 룰 빌더 실행

룰 빌더를 실행하면 Agenda, Working Memory, Trace, Output 패널에서 현재 상태의 설정된 상태를 볼 수 있다. 그림 3.8은 실행했을 때 룰 엔진이 생성되는 화면이다.

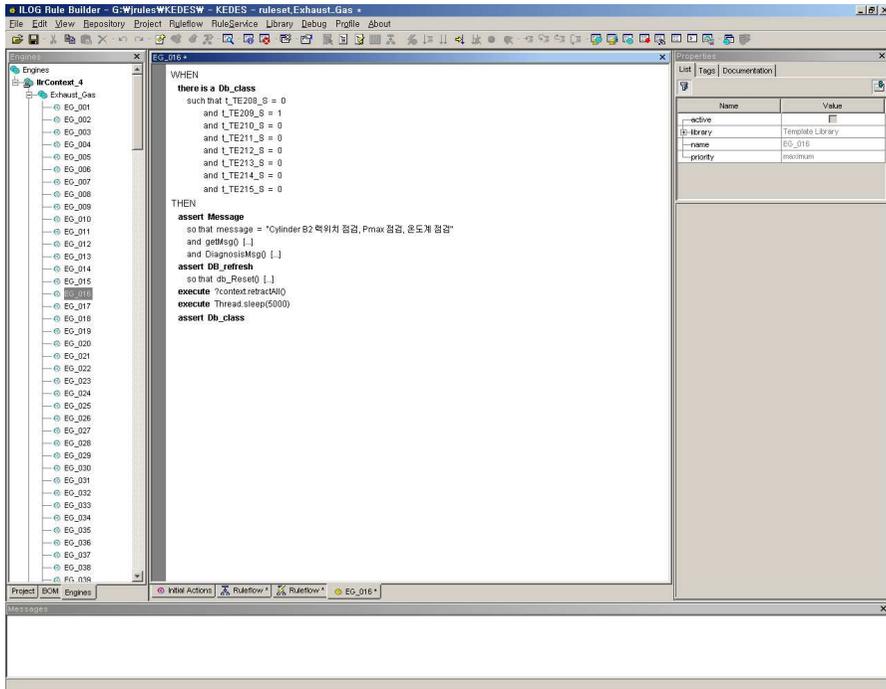


그림 3.8 룰 엔진 생성화면

제 4 장 시스템 구현

4.1 시스템의 전체 구성 및 기능

비상디젤엔진용 진단 시스템의 구축을 위해 먼저 비상디젤엔진계통의 상태 감시인자들을 기반으로 하여 진단데이터의 선정 및 진단데이터의 이상상태와 증상간의 인과관계를 도출하여 지식베이스를 구축한다. 구축된 지식베이스와 JRules의 추론기구를 결합하여 진단모듈을 완성하면 감시모듈 및 계측모듈과 연결하여 진단인자를 가져오거나 진단결과를 감시부에 출력하는 기능을 추가하여야 한다. 본 논문에서는 그림 4.1과 같이 계측모듈로 원전의 DMDS 감시시스템의 시운전 데이터를 실시간 샘플링 데이터로 활용하였고, 감시모듈로는 SCADA 시스템 In-Touch를 이용하여 GUI 화면을 제작하였으며, JRules 기반의 진단모듈과 연결시켰다. 원전의 비상발전기 디젤엔진의 감시모듈과 진단모듈은 다음과 같은 전략으로 개발하였다.

(1) 울진 5, 6호 비상발전기 디젤엔진의 진단데이터와 증상과의 인과관계를 규칙기반 진단지식으로 만들기 위해 진단대상 전문가의 협의에 의해 관련 자료를 수집하여 많은 분석과 논의를 거쳐 전문가의 경험적 진단방식의 지식으로 정리하였다. 이렇게 정리한 지식을 전문가시스템 개발도구인 JRules를 이용하여 규칙형 진단지식으로 변환하여 지식베이스에 저장하였다.

(2) 울진 5, 6호 비상발전기 디젤엔진의 DMDS 시운전데이터로 보관된 DB를 분석하였다. 실시간데이터를 얻기 위해서는 현장의 각 센서로부터 직접 통신라인을 통하여 전송받아야 하지만, 본 논문에서는 여러 가지 제

약조건 때문에 프로토타입으로 구현할 계획으로 울진 3, 4호 비상발전기 디젤엔진의 DMDS 감시시스템의 시운전 데이터가 MySQL로 저장되어 있기에 이 데이터를 가상의 실시간 데이터로 이용하도록 In-Touch의 내장 함수를 사용하여 소프트웨어를 작성하였다.

(3) 비상발전기 디젤엔진의 상태를 실시간으로 감시하여 이를 표시해주기 위해 엔진을 계통별로 분류한 후, 이를 GUI 형태로 화면상에 표시하고 이 화면상에 실시간 데이터도 함께 표시하도록 기능을 설계하였다. 이를 위해 In-Touch를 이용하여 비상발전기 엔진의 계통별 MIMIC 다이어그램, 실시간 트렌드, 히스토리 트렌드, 알람 히스토리 리스트, 현재 알람, 고장진단결과표시, 예측진단결과표시 기능들을 구현하였다.

(4) 각 실시간데이터의 상태를 감시하여 알람 발생시에는 이를 화면상에 표시하고 사용자가 인식하도록 하였으며, 진단이 필요한 경우에는 진단모듈에서 진단하도록 하였다. 이를 위해 In-Touch를 이용하여 각 센서의 정상범위를 설정하여 이 범위를 초과하였을 경우 알람을 발생하게 하고 이를 화면상에 표시하도록 하였다.

(5) 진단모듈에서는 규칙형 진단지식 및 현재 상태와 증상을 기반으로 고장진단과 예측진단을 도출하고, 감시모듈과의 통신을 위해 Java 언어를 이용하여 DDE 통신 인터페이스를 작성하였다.

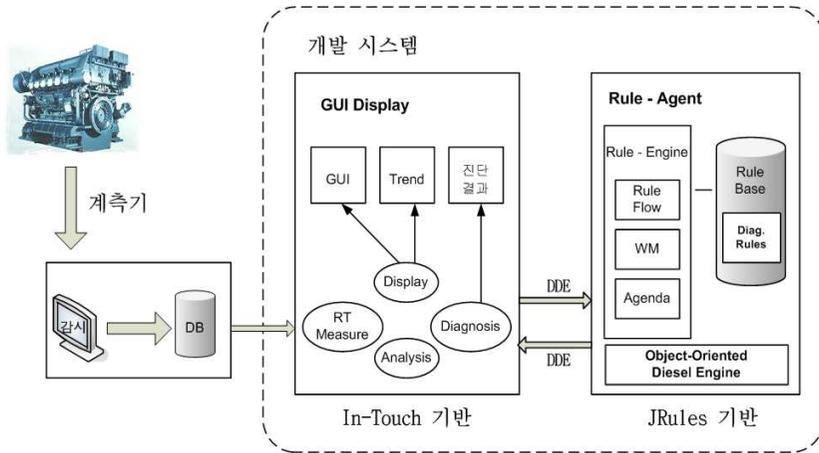


그림 4.1 시스템 구성도

4.2 In-Touch 기반 계측모듈 및 감시모듈

감시 및 진단시스템은 EDG 운전 중 운전상태의 감시, 기기들의 정상상태 유지 확인 및 정비, 엔진의 비정상 상태 조기발견에 있음을 주목하고 GUI 환경을 통한 중앙 집중적인 운전감시를 위해 울진 5, 6호 EDG 파이프 계통을 기반으로 하여 SCADA 시스템 In-Touch를 이용하여 GUI 화면을 완성한다. GUI 화면 구성은 EDG 운전 중 감시되어야 할 중요 인자를 중심으로 제작한다. 울진 5, 6호 비상디젤발전기 엔진의 DMDS 감시시스템의 시운전 데이터베이스와 연동하기 위해 ODBC(Open Database Connectivity)를 이용하여 시스템을 구성한다. 그리고 감시되어야 할 중요 인자로서 화면에 표시되는 알람 인자들은 EDG로부터 실시간으로 수집할 수 있는 데이터에 한해서만 표시한다.

알람 인자들은 다음과 같이 아날로그형 데이터와 디지털형 데이터로 나누

어진다.

(1) 아날로그형 데이터인 경우

데이터 값을 실시간으로 숫자, 바, 트래드, 형태로 표시되도록 제작하고, 알람 발생은 In-Touch 자체기능을 이용하며 알람 발생 한계값은 원전용 디젤기관에서 제시하는 한계값을 기준으로 알람이 발생할 수 있도록 제작한다.

(2) 디지털형 데이터인 경우

데이터의 온-오프 상태 정보를 GUI 화면에 표시되도록 제작한다. 정상일 때는 Normal로 표시가 되며, 반대로 작동시에는 알람이 발생할 수 있도록 제작한다.

그림 4.2는 ‘Tagname Dictionary’ 화면을 이용하여 센서값을 설정하는 경우이다. 여기에서는 센서의 태그네임을 정의하고, 알람 발생 한계값을 설정할 수 있다.

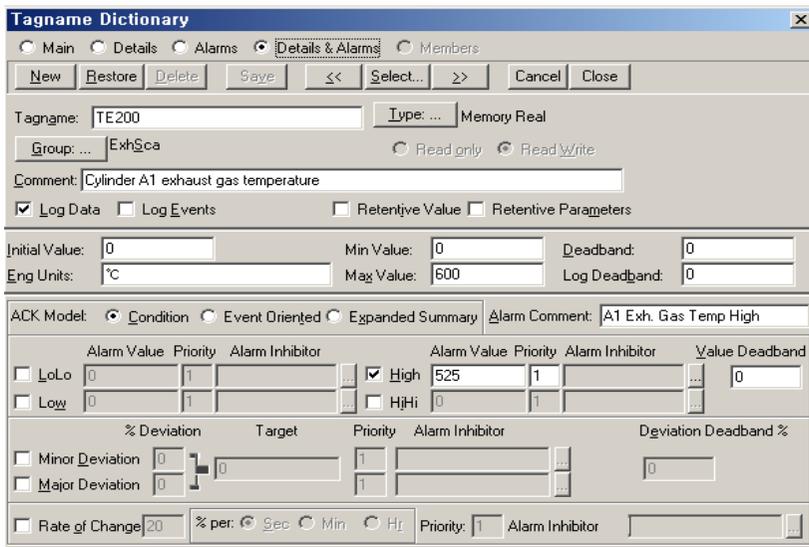


그림 4.2 In-Touch에서의 태그 설정 화면

4.2.1 감시모듈의 GUI 화면 구성

감시 모듈은 SCADA 시스템 In-Touch를 이용하여 GUI 화면을 구성하고 다음과 같이 윈도우 화면을 구성한다.

(1) Alarm History Window

알람은 고장의 원인이 될 수 있는 EDG 운전 조건들의 경고를 말하며 EDG 운전자의 응답을 필요로 하는 것을 포함한다. 이 윈도우는 현재 발생한 알람뿐만 아니라 과거 발생한 알람과 알람발생요인의 조치 및 처리 후 정상상태로 되돌아오는 것까지 확인할 수 있도록 제작한다.

(2) Bearing Temperature Window

EDG 각 베어링 온도를 한 화면에서 관찰할 수 있으며 전체 베어링의 정보 한계값 및 현재 평균값을 막대그래프와 텍스트를 통해 실시간으로 확인할 수 있도록 제작한다.

(3) CC & LT Water System Window

설계도면을 바탕으로 각 탱크, 중요 펌프, 쿨러 및 파이프만을 간결하게 표현했으며 EDG에 유입되는 해수와 공기의 온도의 경우에 아날로그 값으로 실시간 확인 및 관찰을 위해 텍스트로 제작한다.

(4) Compressed Air System Window

설계도면을 바탕으로 각 부속기기와 공기 파이프만을 간결하게 표현하여 제작한다.

(5) Exhaust Gas & Intake Air System Window

설계도면을 바탕으로 EDG의 과급기 및 각 파이프를 표현한다. 각 실린더 및 과급기 In/Out 온도의 경우에는 막대그래프와 텍스트를 통해서 실시간으로 확인할 수 있도록 제작하며 알람 한계값 및 현재 평균값을 확인할 수 있도록 제작한다.

(6) Fuel Oil System Window

설계도면을 바탕으로 각 탱크, 중요 펌프, 연료 누수탐지기 및 파이프만을 간결하게 표현하여 제작한다.

(7) History Trend Window

이전의 시간과 날짜로부터 데이터의 히스토리를 제공하는 화면으로서 같은 측정항목에 대해서 서로 다른 시간대의 히스토리를 비교 및 확인할 수 있도록 제작한다.

(8) HT Water System Window

설계도면을 바탕으로 각 탱크, 중요 펌프, 쿨러 및 고온냉각수 파이프만을 간결하게 표현하며 EDG에서 유출되는 고온냉각수 온도의 경우에는 아날로그 값으로 실시간 확인 및 관찰을 위해 텍스트 형태로 표시되도록 제작한다.

(9) L.O. System Window

설계도면을 바탕으로 각 탱크, 중요 펌프, 윤활유 섬프 탱크 및 파이프만을 간결하게 표현하며 EDG에 유입되는 저온냉각수 온도 및 압력의 경우에는 아날로그 값으로 실시간 확인 및 관찰을 위해 텍스트 형태로 표시되도록 제작한다.

(10) Real Time Trend Window

EDG로부터 실시간 입력되는 아날로그 데이터 값을 트렌드 형태로 실시간 확인 및 관찰이 가능하도록 제작한다. 확인 및 관찰이 가능한 항목으로는 배기가스 온도, 소기계통 온도, 메인베어링 온도를 선택할 수 있도록 제작한다.

(11) Fault-Diagnosis & Early-Diagnosis Window

알람이 발생했을 때의 고장 진단과 시계열 분석에 의한 예측진단 결과를 표시한다. DDE 통신을 이용해서 감시모듈의 고장 진단창 및 예측진단

창에 출력하게 된다.

그림 4.3은 배기가스 계통의 MIMIC 표시부, 현재 알람부, 진단부, 메뉴부 등으로 구성된 화면을 보여주고 있다.

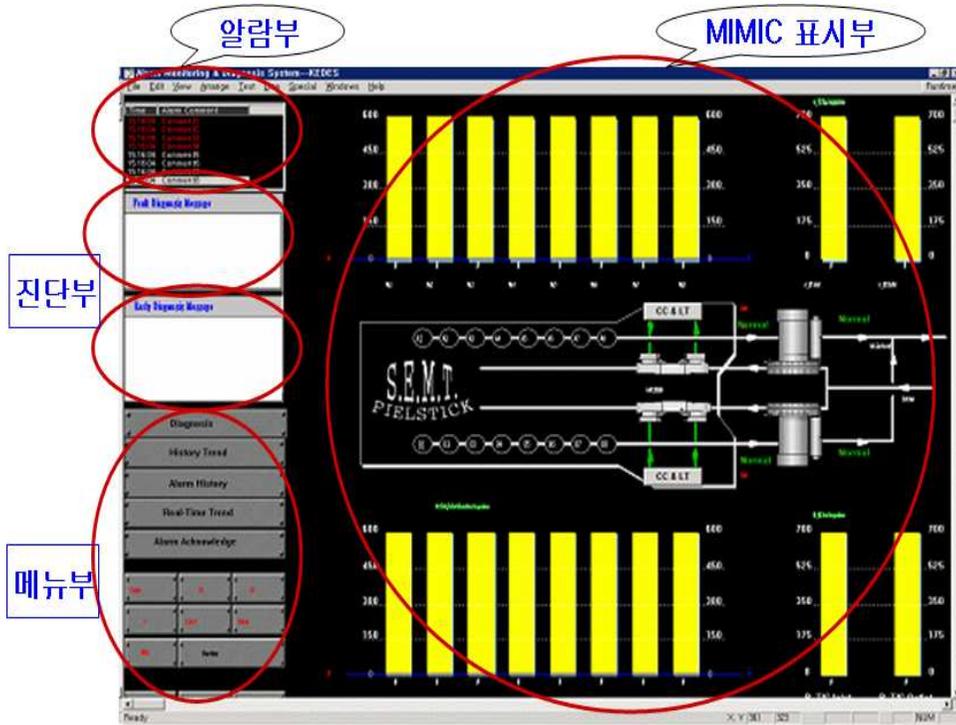


그림 4.3 배기가스 계통의 GUI 화면

4.3 고장진단 모듈

고장진단은 그림 4.4와 같은 흐름으로 진단이 이루어진다. DMDS 감시 시스템의 데이터베이스로 부터 시운전 데이터를 1초 간격으로 샘플링하여 감시부에서 센서값으로 이용하고, 이 센서값을 DDE 통신을 이용하여 JAVADDE 클라이언트에서 전송 받아 진단부에서의 센서값으로 이용한다. 진단부에서는 알람이 발생할 경우 현재 센서값으로부터 설정 레벨에 따른 상태값과 시계열 데이터를 이용한 증상 추정 알고리즘에 의해 증상값을 추출한다. 추출된 상태값과 증상값을 작업메모리에 저장한다. 작업메모리에 저장되면 추론엔진은 지식베이스와 작업메모리에 저장된 상태값과 증상값을 가지고 진단을 시작하게 되며 진단결과를 도출한다. 최종적으로 진단부에서는 진단결과를 JAVADDE 서버를 이용하여 감시부에 전송하게 되고, 감시부에서는 전송받은 진단결과를 고장진단 결과창에 출력하게 한다.

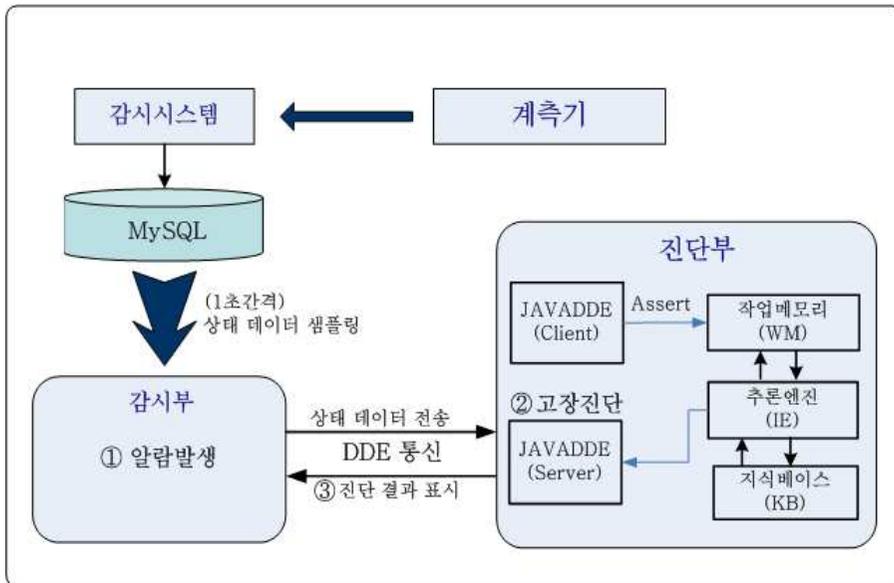


그림 4.4 고장 진단의 흐름도

4.4 예측진단 모듈

예측진단은 그림 4.5와 같은 흐름으로 진단이 이루어진다. 먼저 감시부에서는 시운전 데이터를 1초 간격으로 샘플링하여 센서값으로 이용하고, 이 센서값을 DDE 통신을 이용하여 JAVADDE 클라이언트를 통해 전송 받게 된다. 진단부에서는 예측진단을 위해 먼저 시계열에 의한 데이터를 증상 추정 알고리즘에 의해 증상을 추출한다. 여기에서 추출된 센서들의 증상값을 작업메모리에 저장한다. 증상값들이 Increase 또는 Decrease인 경우가 발생하면 추론엔진은 지식베이스와 작업메모리에 저장된 센서들의 증상값을 가지고 진단을 시작하며 진단결과를 도출한다. 도출된 진단결과를 JAVADDE 서버를 이용하여 감시부에 전송하고, 감시부에서는 전송받은 진단결과를 예측진단 결과창에 출력한다.

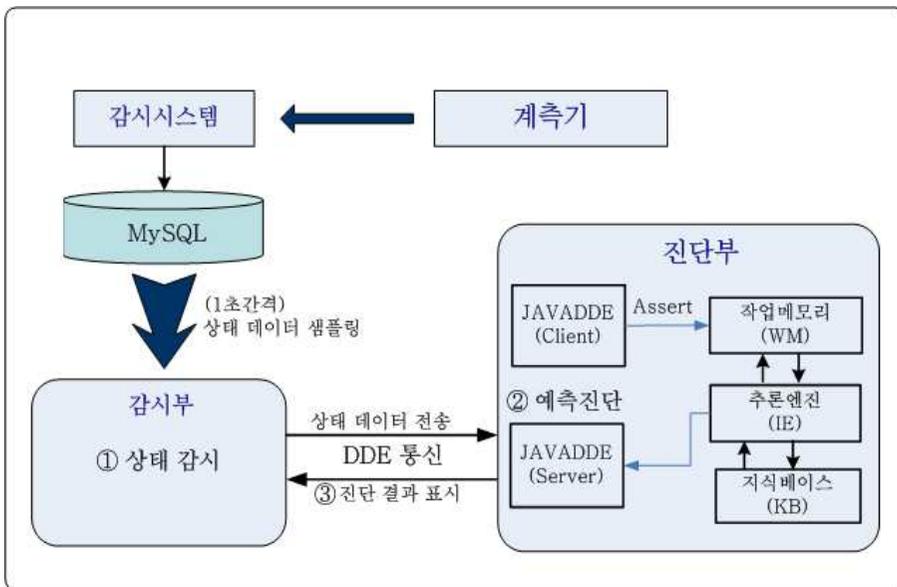


그림 4.5 예측 진단의 흐름도

4.5 진단모듈과 상태감시 모듈간의 인터페이스

DDE는 윈도우즈 환경에서 어플리케이션 간에 서로 데이터를 주고받을 수 있도록 마이크로소프트에 의해 디자인된 통신 프로토콜이다. 이는 동시에 실행되는 두개의 프로그램간에 클라이언트/서버의 관계를 형성한다. 서버 어플리케이션은 자신의 데이터를 원하는 다른 어플리케이션으로부터 요청을 받으며 데이터를 제공한다. 요청하는 어플리케이션을 클라이언트라 부르고, 몇몇 어플리케이션은 클라이언트인 동시에 서버가 될 수 있다.

본 논문에서는 진단모듈과 감시모듈간의 인터페이스를 DDE 통신을 이용하여 Java 언어로 작성하였다. 진단모듈은 계측데이터를 감시모듈에서 가져오는 경우에는 JAVADDE 클라이언트 모델로 구성하고 진단 결과를 감시모듈에 전송할 경우에는 JAVADDE 서버 모델로 구성하였다. 그림 4.6은 진단모듈과 감시모듈간의 DDE 통신의 흐름을 보여주고 있다. 여기에서 In-Touch와 DDE 통신을 하기 위해서는 In-Touch에서 'TagName Dictionary'를 이용하여 토픽 네임(Topic Name)과 아이템 네임(Item Name)을 설정해주어야 하며, Java 언어를 이용하여 클라이언트/서버 모델로 JAVADDE를 작성한다. 클라이언트에서는 In-Touch에서 샘플링 되는 센서값들을 가져오게 되고, 현재 상태값과 증상값을 추출하게 된다. 이렇게 추출된 값들을 작업메모리에 저장하게 된다. 그리고 추론엔진에서 추출된 진단결과를 JAVADDE 서버에서 감시부로 전송하게 된다.

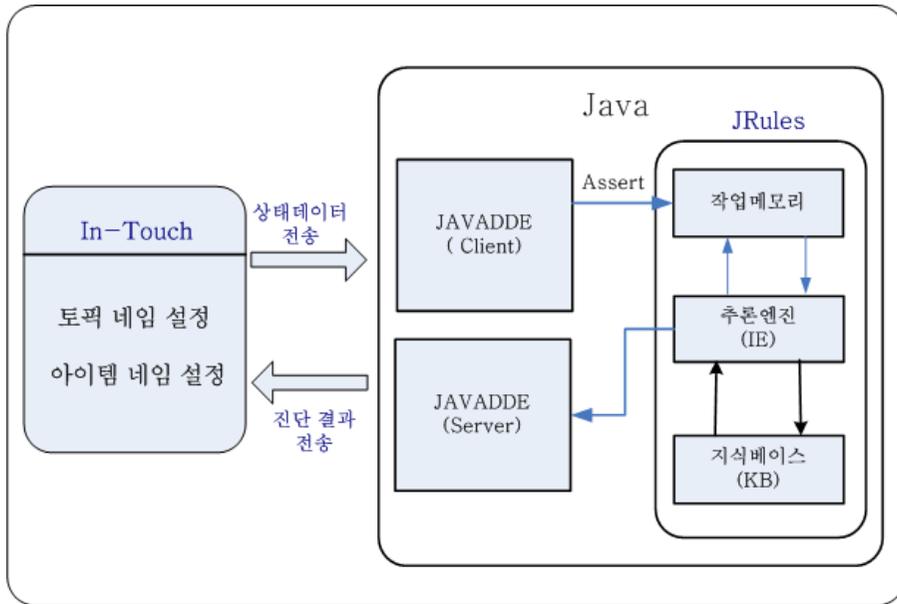


그림 4.6 진단모듈과 감시모듈간의 인터페이스

4.6 시운전 데이터 분석

울진 5, 6호 비상발전기엔진 DMDS의 시운전데이터로 저장된 데이터베이스를 분석하였다. 이 데이터베이스는 MySQL로 저장되어 있으며 데이터베이스의 이름은 INSPECT로 되어있다. 먼저 INSPECT DB의 테이블은 총 10개로 이루어져 있고, 표 4.1과 같이 구성되어 있다. INSPECT DB를 분석해 본 결과 'fastacquisition_points_dg56' 테이블은 정지시 데이터가 대부분이라 사용하지 않았고, 'shorttrends_dg56_stopped' 테이블은 엔진 정지 시 매 15분 간격으로 저장되어 있으며, 'shorttrends_dg56' 테이블은 엔진운전 중 매 15초 간격으로 저장되어 있었다. 본 논문에서는 'shorttrends_dg56' 테이블을 사용하여 감시부에서 일정시간 간격으로 데이터를 읽어 들여 이 값을 실시간 센서값으로 활용하였다.

표 4.1 INSPECT 데이터베이스

테이블	레코드 수	타입	크기
designation	481	MylSam	84.6KB
eventrecorder	15814	MylSam	3.1MB
fastacquisition_header_dg56	733	MylSam	126.5MB
fastacquisition_points_dg56	171208	MylSam	314.5KB
longtrends_blocks_dg56	34	MylSam	32.0KB
longtrends_curves_dg56	17	MylSam	2.70KB
longtrends_segments_dg56	17	MylSam	3.1KB
pmax	36	MylSam	100.2KB
shorttrends_dg56	38019	MylSam	72.1MB
shorttrends_dg56_stopped	21986	MylSam	41.2MB
테이블 10개	248345	--	431.2M

‘designation’ 테이블은 진단에 필요한 진단데이터들을 정의하고 있다. 진단데이터는 총 235개로 구성되어 있고 각각 아날로그 데이터 52개, 디지털 데이터 101개, 통신기기 데이터 82개로 구성되어 있다. 본 논문에서는 비상발전기 디젤엔진만을 대상으로 하여 상태감시 및 진단을 수행하는 것을 목표로 하였으므로 디젤엔진과 직접적으로 관련되는 91개의 진단데이터만 사용하였다. ‘shorttrends_dg56’ 테이블을 분석한 결과 엔진 운전 중에 매 15초 간격으로 235개의 진단 데이터를 저장하고 있으며, 연속측정이 360회(90분)이상인 연속된 레코드들을 정리하여 이 데이터들을 실험 대상으로 이용하였다.

제 5 장 실행

원전의 비상발전기용 디젤엔진을 대상으로 구축한 상태감시 및 진단시스템의 상태감시 및 진단결과를 확인하기 위하여 시운전 데이터를 조작한 후 실행하여 테스트 했다. 감시부에서는 원전의 DMDS 감시 시스템 시운전 데이터로 저장된 INSPECT 데이터베이스의 'shorttrends_dg56' 테이블을 ODBC로 연결하여 실시간 샘플링 데이터로 사용하였고, 진단부에서는 고장진단과 예측진단 두 가지로 구성하여 Java 언어로 배포하여 실행하였다. 고장진단의 경우 알람이 발생하면 현재 알람 리스트에 알람이 추가되고, 추론엔진이 알람에 대한 현재 센서의 상태값과 구축된 지식베이스를 가지고 추론을 시작하여 진단결과를 감시부에 전송하고 감시부에서는 이결과를 고장진단 결과창에 제대로 표시하는지를 확인하였다. 그리고 예측진단의 경우 시계열 데이터에 의해 추출한 증상값과 지식베이스를 이용하여 추론엔진이 진단을 시작하고, 진단결과를 감시부에 전송하여 감시부에서 이결과를 예측진단 결과창에 제대로 표시하는지를 확인하였다.

구축한 시스템은 고장진단의 경우 그림 5.1과 같은 흐름으로 실행을 하였다. 먼저 감시부와 진단부를 실행하였으며, 이때 샘플링되는 데이터는 정상값을 가지고 있기 때문에 임의로 샘플링되는 데이터의 값을 알람이 발생하도록 변경하였다. 변경 후 다시 감시부를 실행하여 알람이 발생하는지, 그리고 진단부가 알람에 대한 진단을 시작하는지를 확인하였다.

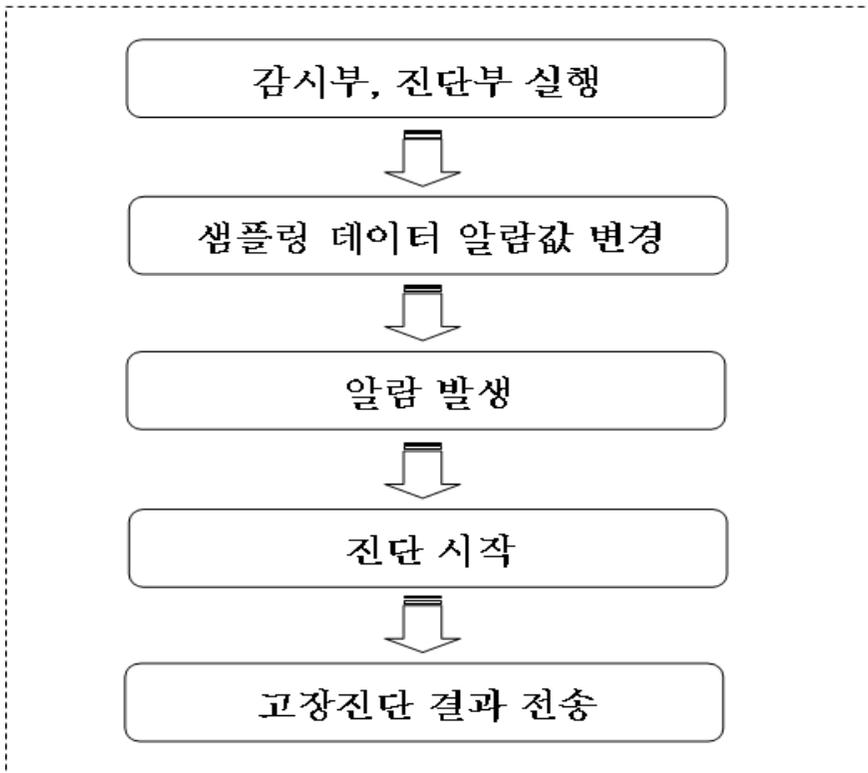


그림 5.1 고장진단 실행 흐름도

고장 진단의 경우는 그림 5.2와 같이 하나의 실린더 A1의 값을 580으로 알람이 발생하도록 값을 변경하여 실행하였다. 실행한 화면은 그림 5.3에서 배기가스 계통에 대한 실시간 MIMIC 다이어그램과 알람에 대한 고장진단 결과를 보여주는 화면이다. MIMIC 표시부에서는 알람이 발생하면 막대그래프가 붉은색으로 깜빡이게 되고, 현재 알람에서는 알람이 리스트에 추가된다. 그리고 진단부에서는 알람이 발생한 현재 센서값을 작업메모리에 저장하게 된다. 추론엔진은 작업메모리에 저장된 값과 지식베이스의 진단지식을 가지고 추론을 시작하며, A1 실린더의 과부하라는 진단결과를 DDE통신을 이용하여 감시부에 전송한다. 감시부에서는 고장진단부에 진단결과를 출력하게 된다.

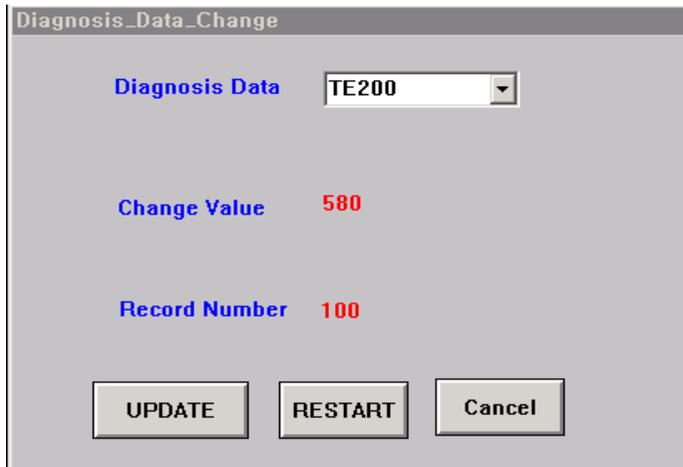


그림 5.2 알람값으로 변경하기 위한 화면

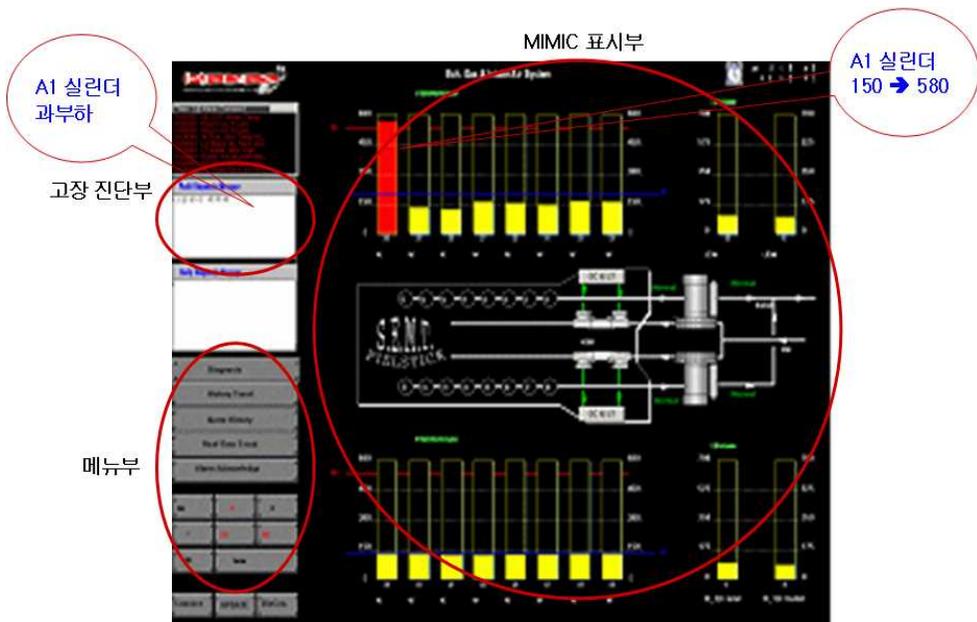


그림 5.3 상태감시 및 고장진단 결과

예측 진단의 경우 그림 5.4와 같은 흐름으로 실행하였다. 먼저 감시부와 진단부를 실행하였으며, 이때 샘플링되는 데이터는 일정한 정상값을 가지

고 있어서 예측진단을 수행하지 않기 때문에 임의로 샘플링되는 데이터의 값을 감소하도록 변경하였다. 변경 후 다시 감시부를 실행하면 샘플링되는 데이터를 진단부에서 DDE통신을 이용하여 가져가고 샘플링되는 데이터의 증상을 추출한다. 진단부에서는 증상값을 가지고 진단을 시작하고 진단결과를 다시 DDE 통신을 이용하여 감시부에 전송하였다. 감시부에서는 전송받은 진단결과를 예측진단 결과창에 출력하였다.

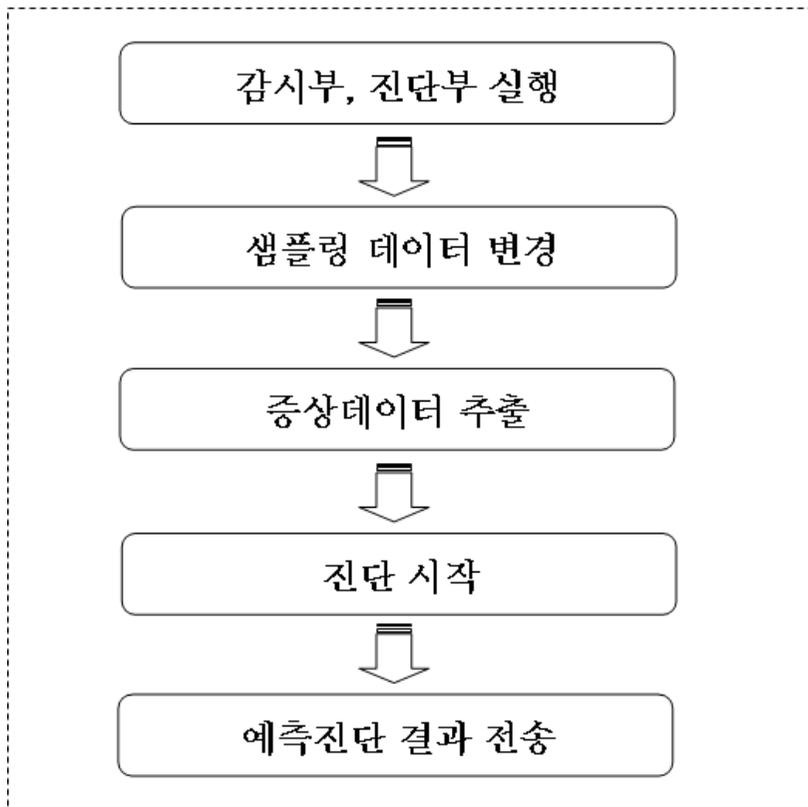


그림 5.4 예측진단 실행 흐름도

그림 5.5에서 예측 진단의 실행 화면을 보여주고 있다. 여기에서 배기 가스 계통의 A1 실린더의 값을 감소하게 임의로 설정하였으며, 실린더

A2에서 A7까지는 일정하게 하였다. 실제 예측 진단의 경우에는 알람이 발생하지 않고 앞으로의 경향을 추정하기 위해서 진단을 수행한다. 감시부에서 샘플링되는 실시간 데이터를 JAVADDE 클라이언트에서 가져와서 증상 추정 알고리즘에 의해 증상값을 추출하고, 추출된 증상값을 작업메모리에 저장한다. 추론엔진은 작업메모리에 저장된 증상값과 지식베이스의 진단지식을 이용하여 추론을 시작하며, A1 실린더 랙위치 점검, Pmax 점검, 온도계 점검이라는 진단결과를 도출하였다. 이 진단결과를 JAVADDE 서버를 이용하여 감시부에 전송하고 감시부에서 예측진단부창에 진단결과를 출력하였다.

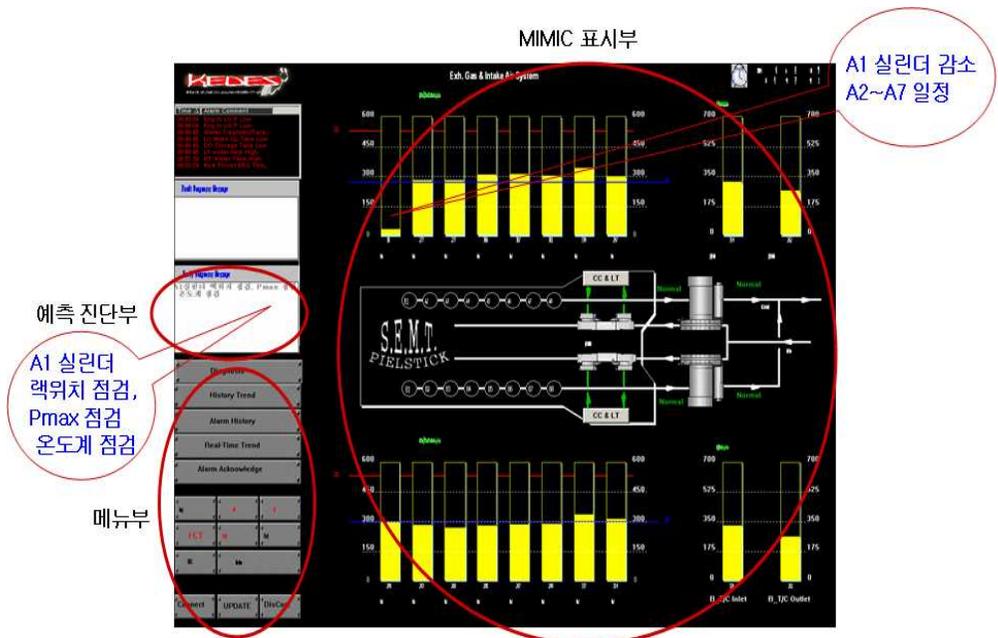


그림 5.5 상태감시 및 예측진단 결과

제 6 장 결론 및 향후과제

본 논문에서는 비상발전기 디젤엔진을 위한 진단기술에 관해서 논하였다. 지식 표현 기법으로는 객체지향 표현 기법과 규칙기반 표현법을 이용하였으며, 지식베이스를 구축하기 위해서 진단 대상에 대한 클래스화 및 인과관계를 정의하였다. 정의된 인과관계를 JRules를 이용하여 진단 지식으로 변경하여 지식베이스를 구축하였다.

감시모듈과 계측모듈은 SCADA 시스템 In-Touch 이용하여 구성하였고, 진단모듈은 전문가시스템 개발도구 JRules를 이용하여 구축하였다. 진단은 알람에 대한 현재 센서값의 상태값을 가지고 진단하는 고장진단과 시계열 데이터에 의해 추출한 증상값을 가지고 진단하는 예측 진단의 두 종류로 구현하였다. 또한, 진단모듈과 감시모듈간의 연결을 위하여 Java 언어를 이용하여 DDE 통신 인터페이스를 구현하였다.

현재까지 진단지식을 3계통에 한해서 작성하였기 때문에 향후 보완하여 실용적인 시스템으로 완성하여야 하며, 실시간 데이터의 증상을 3종류로 국한하였으나 정밀한 진단을 위해서는 추후에 확장할 필요가 있다.

참고 문헌

- [1] 조권희, “국외 원전 고장사례 분석을 통한 국내 비상디젤엔진 계통 신뢰도개선 연구”, 한국 박용기관학회, 춘계학술대회 논문집, pp.137-147, 2004
- [2] 조권희 외 4명, “국외 원전 고장사례 분석을 통한 국내 비상디젤엔진 계통 신뢰도개선 연구”, 한국 박용기관학회, 춘계학술대회 논문집, pp.137-147, 2004
- [2] 최재성, 조권희, “비상디젤발전기 기계계통 열화평가 및 신뢰도 개선 방안 연구”, 한국원자력안전기술원, KINS/ HR-366, 2000.
- [3] 최재성, 조권희, “비상디젤발전기 기계계통 열화평가 및 신뢰도 개선 방안 연구”, 한국원자력안전기술원, KINS / HR-435, 2001.
- [4] 이재규 외 5명 공저, “전문가시스템 원리와 개발”, 법영사, 1996.
- [5] 이재규 외 5명 공저, “전문가시스템의 응용과 사례분석”, 법영사, 1996.
- [6] 모경주 외 3명, “화학 공정의 이상진단을 위한 조업지원 시스템의 개발”, 전문가시스템학회지, 제 2권, 제 1호, pp.11~26, 1996.
- [7] 정학영, 박현신, “경보처리 기반 진단 시스템 개발”, 전문가시스템학회지, 제 4권, 제 1호, pp.103~113, 1998.
- [8] R. Khosla and T. Dillon, “Enabling Technology for Diagnostic Applications“, (Industrial and Engineering Application of Artificial Intelligent and Expert Systems) : Proceedings of the Eight International conference on held in Melbourne, Australia, pp.263~272, 1995.
- [9] 최욱현, “하이브리드 지식표현을 이용한 선박 엔진 고장진단 전문가시

- 시스템의 구현에 관한 연구”, 석사학위 논문, 1999.
- [10] http://www.aistudy.co.kr/expert/expert_system.htm, 2003.
- [11] ILOG JRULES for developers “Customer Education”, 2004.
- [12] 김달현, “선박엔진용 실시간 고장진단시스템의 구현에 관한 연구”, 석사학위논문, 2001.
- [13] 최재곤, “선박 기관실 모니터링 시스템을 위한 실시간 데이터 처리에 관한 연구”, 석사학위논문, 1998.2.
- [14] HSD-SEMT Pielstick 16PC2.5V-400 of Ulchin nuclear power plant unit 5&6 Service information, 2003.
- [15] InTouch 7.1 Basic Training Manual, 2002.
- [16] ILOG JRULES 4.6 Rule Builder Tutorial, 2003
- [17] InTouch 9.0 Advanced Training Manual, 2004.
- [18] IndustrialSQL Server 8.0 Basic Training Manual, 2004.
- [19] 이현우, “Java Programming Bible for JDK 1.4”, 영진닷컴, 2003.
- [20] 서환수, “MySQL 시스템 관리와 프로그래밍 : 자바, PHP, 펄, C, 파이썬”, pp.370~394, 한빛미디어, 2002.

감사의 글

본 논문이 완성되기까지 여러모로 부족한 저를 제자로 받아주시고 따뜻한 사랑과 인내로 지켜봐 주신 류길수 교수님께 깊은 감사를 드립니다. 또한 바쁘신 와중에도 본 논문의 심사를 맡아 세심한 지도를 해 주신 신옥근 교수님, 김재훈 교수님께 감사드립니다. 또한 대학원에서 제게 가르침을 주신 박휴찬 교수님과 비록 강의를 듣지는 못하였지만 자상하게 대해 주신 손주영 교수님, 이장세 교수님께도 감사드립니다. 그리고 여기까지 저를 있게 한 하창승 교수님께도 깊은 감사의 뜻을 전합니다.

연구실에서 같이 공부한 성창규 선배님, 병수형, 김태진님, 원라경님께게도 감사의 마음을 전하고 싶습니다. 그리고 인공지능 연구실 황동희 선배님, 하희천 선배님, 최대석 선배님, 최재곤 선배님, 최욱현 선배님, 윤갑렬 선배님, 김재환 선배님, 그 외 인공지능 연구실 선배님들께 깊은 감사를 드립니다.

대학원 석사과정을 마칠 때까지 학교생활을 함께하며, 후배들에게 많은 힘이 되어 주신 하동경 선배님, 박정임 형수님, 항상 DB랩에 가도 즐겁게 대해주시고 부족한 논문을 위해 조언을 아끼지 않으신 이성대 형님, 훈훈한 형같은 곽용운 선배님, 학업에 있어서 선봉이 되는 감승철님, 연구에 전념하시는 유강주님, 지금은 졸업했지만 항상 저의 고민을 받아주는 형석선배, 지리산 꼭 같이 가고 싶은 남언규 선배님, 술 잘 마시고 털털한 문성미님, 흥반장같은 박철현님, 같이 졸업은 못했지만 항상 편안하게 대해준 김강민님, 연구에 몰두하는 사장님 같은 카리스마가 넘치는 박은진님, 학교로 다시 돌아와 학업에 열중인 유흥섭님, 순진하고 착한 전성환님, 말 잘 안듣는 백현철님, 활발한 이석현님, 밤샘으로 학교를 깨끗이 지키는 성주용님 그 외 같이 생활한 학부생과 대학원생들에게도 깊은 고마

움을 전합니다.

대학원 생활동안 저에게 잘해주시고 자상하신 강군호 조교님께도 고마움을 전하고 내년에는 꼭 이쁜 형수님 만나길 바랍니다. 그리고 항상 마음씨 좋으신 김경언 조교님께도 감사의 마음을 전합니다. 그리고 영원한 나의 벗 전상근님, 말 잘하고 유머감각 있는 전병도님, 아버지라 부르는 조주영님, 한은정님, 왕 언니같은 최정빈님 그 외 나의 군생활에서 빛이 바라게 했던 해병대 선·후배님께 깊은 감사를 드립니다.

항상 철부지 막내 동생을 걱정하고 따뜻한 사랑과 관심을 가져주는 박광순 누님, 동생에게 힘과 용기를 북돋아 주는 박영순 누님, 조카들을 위해 열심히 일하시는 박임순 누님, 우리집의 기둥인 박석순님께도 깊은 감사와 고마움을 전합니다.

끝으로, 세상에서 가장 사랑하는 나의 평생 동반자인 아내, 아직 뱃속에서 세상을 보기위해 기다리는 자식, 못난 저를 항상 사랑으로 보살펴 주시고 묵묵히 지켜봐 주신 장인어른과 장모님, 그리고 평생 저희 6남매를 키우기 위해 농사일에 전념하신 아버지와 어머니께 이 논문을 바칩니다.