

진단 시스템을 위한 혼합형 추론 엔진

김진평¹, 이길재², 김문현³

경기도 수원시 장안구 천천동 300번지 성균관대학교 정보통신공학부

인공지능 연구실

payon@ece.skku.ac.kr, utp99@daum.net,

mhkim@simsan.skku.ac.kr

HybridInference Engine for System Diagnosis

Jin-Pyung Kim¹, Gil-Jae Lee, Moon-Hyun Kim³

School of Electrical and Computer Engineering, SungKyunKwan

University

300 ChunChun-Dong, JangAn-Gu, Suwon, KyungGi-Do 440-746,

Korea

payon@ece.skku.ac.kr, mhkim@simsan.skku.ac.kr

Abstract

본 논문에서는 진단시스템의 추론성능을 향상시키기 위한 방법으로서, 사례 기반 추론을 통해서 규칙 기반 추론의 단점을 보완하여 성능을 향상시키는 혼합형 추론 모델을 제안한다. 본 모델의 특징은 규칙 기반 추론의 확장성 문제와 규칙화 할 수 없는 예외적인 상황에 대한 문제점을 사례 기반 추론에서 사례로 저장하여 규칙 기반 추론의 단점을 보완하는데 있다. 이런 두 모델의 문제점을 해결하는 과정은 첫째로, 문제에 따라 규칙기반추론 모듈의 규칙 베이스를 통해서 적절한 규칙을 적용 후 추론을 적용하여 근접한 해를 얻어낸다. 두번째로, 규칙베이스에 저장되어 있지 않은 문제에 대해서는 사례 라이브러리를 검색하고 유사성 검사를 통해서 저장된 사례를 찾아 입력된 사례에 적용하여 문제를 해결한다. 셋째로, 해결된 문제에 대해서 수정작업을 통해 사례 라이브러리를 확장한다. 위와 같이 세 과정을 통해 본 논문에서 제안하는 방법론의 성과를 측정하기 위하여 정비 메뉴얼을 규칙화하여 규칙베이스를 구축하였고 전문가들의 경험적인 지식에 대해서는 사례라이브러리로 구축하였다. 또한 지식베이스를 통해서 진단을 수행하고 해결된 문제에 대해서 정확도 검사를 통해 진단의 정확성을 측정하여 혼합형추론엔진의 성능을 검증하였다.

1. 서 론

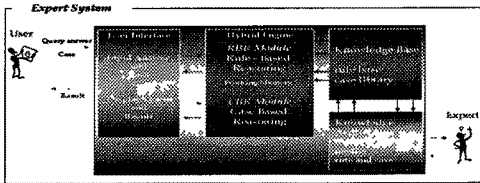
전문가의 지식과 경험을 바탕으로 구성된 지식과 효율적인 추론방법을 통해서 문제를 해결하며 다양한 사용자의 요구사항을 충족시켜주는 지능적 컴퓨터 프로그램을 전문가시스템이라고 한다[1]. 전문가시스템은 문제 분야에 대한 특정하고 다양한 지식을 기반으로, 현실적으로 유용성이 많은 분야에 여러 가지 추론기법을 토대로 적용되고 있다[2]. 규칙기반추론은 추론의 과정이 인간의 사고 과정과 유사하다는 특성과 강력한 도메인 이론을 가지고 있다는 특성 때문에 과거의 전문가 시스템에서 주로 사용되었다. 그러나 전문가시스템을 개발 시 시스템의 성능 향상을 위하여 규칙을 반복적으로 수정하고 추가해야

하고, 확장하기 힘들다는 단점이 있다[3]. 전문가의 지식은 정형화된 규칙만으로 완벽하게 표현해 내기 어려우며, 많은 행위가 과거의 경험을 기초로 하고 있다. 특히 진단 분야에서는 과거의 사례와 비교하는 경우가 많으며, 많은 경우 자신의 경험을 기초로 진단하고 원인을 설명하게 된다. 이처럼 과거 사례와 비교하여 현재의 문제를 해결하는 방법을 사례 기반 추론이라 칭한다[4][5][6][7][8]. 사례기반추론은 규칙기반추론과 마찬가지로 경험적인 지식을 이용하는 반면 잘 정의된 되지 않고 정형화된 규칙으로 추출해내기 힘든 지식에 효율적으로 사용 가능하다. 또한 과거 경험으로부터 자동적으로 지식을 획득하는 사례기반추론은 지식의 획득만 아니라 지식의 증가 또한 용이하게 이루어져 규칙기반추론에서 발견된 문제점을

극복할 수 있는 추론기법으로 인정받고 있다. 따라서 본 논문에서는 일반적으로 정형화하기 쉬운 지식들에 대해서는 규칙으로 표현해 내고 정형화하기 힘든 예외적인 지식들에 대해서는 사례로 표현해내서 규칙과 사례의 상호 보완적인 형태를 취하여 규칙기반추론과 사례기반추론의 혼합형추론방법을 제안하고 진단의 성능을 향상시켰다. 본 논문의 구성은 다음과 같다. 2절에서는 관련(제안) 시스템의 구조와 시스템의 각 모듈별 구성 및 특징에 대해서 소개하며 3절에서는 제안시스템의 실험에 대한 결과 및 평가에 대해서 소개하며 4절에서는 결론을 기술하고 자 한다.

2. 시스템 구성

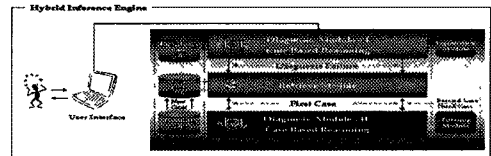
본 논문에서는 고장 진단을 수행하기 위하여 혼합형 추론 엔진을 개발하였다. 혼합형 추론 엔진은 규칙기반 추론 모듈과 사례기반 추론모듈을 병합한 형태이며, 각 모듈에 따라 규칙베이스와 사례베이스로 구성하여 과거의 지식을 저장하였다. 본 절에서는 규칙기반추론의 진단알고리즘 및 설명기능과 사례기반추론의 사례구조 및 유사도 측정, 사례조회에 대해서 기술한다.



[그림 1] 혼합형 진단시스템 구성도

- i) **User Interface:** 사용자가 추론엔진의 응답을 통해서 발생한 문제에 대해 진단을 수행하고, 사용자에게진단과정을 설명, 확인 하는 모듈이다.
- ii) **Hybrid Inference Engine:** 세부적으로 RBR(Rule-Based Reasoning)모듈과 CBR(Case-based Reasoning)모듈로 나뉘어 있으며, 사용자에게 의해서 입력된 문제를 각 모듈에 적용시켜 처리해내는 혼합형 추론 모듈이다.
- iii) **Working Memory:** 스택 구조 형태이며, 추론엔진에서의 추론알고리즘을 통해서 동작하는 모듈이다.
- iv) **Knowledge Base:** Rule Base와 Case Library로 구성되어 있으며, rule base는 경험적 지식들을 규칙화하여 저장하였으며, case library는 검색하기 용이하게 계층화하여 구성하였다. 추론과정에서 지식베이스의 지식을 바탕으로 문제를 해결한다.
- v) **Knowledge Acquisition:** 과거의 지식들과 사례 또는 추론과정에서 해결되거나 해결되지 못한지식들을 수정하여 지식을 확장시킨다.

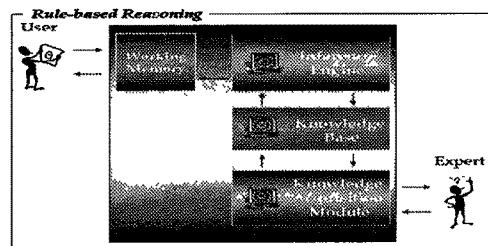
본 논문에서는 경험적인 지식을 규칙화하여 처리한 규칙기반추론과 규칙화하기 힘든 예외적인 지식에 대해서는 사례기반추론으로 표현하여 상호보완적인 요소를 갖춘 혼합형추론시스템을 제안하였다. 혼합형 추론엔진의 전체적인 구조는 [그림 1]과 같으며, 혼합형 추론의 진단과정은 [그림 2]와 같다. 사용자화면을 통하여 입력된 문제에 대해서 지식베이스를 바탕으로 규칙기반 추론과정을 진행하며, 진단에 성공한 문제에 대해서는 설명모듈을 통해서 사용자에게 진단과정을 설명해준다. 진단에 실패한 예외적인 문제에 대해서는 사례기반추론 엔진의 검색모듈로 이동한다. 검색모듈을 통해 저장사례와의 유사성 평가 후 검색된 사례를 활용하여 진단을 수행한다. 진단에 실패한 경우에는 유사한 다른 검색사례를 통해 재 진단을 수행하며, 진단에 성공한 경우는 수정과정을 통해서 사례베이스에 추가되어 사례를 확장하게 된다.



[그림 2] 혼합형 추론엔진 진단과정

2-1.Rule-based reasoning (규칙기반추론)

규칙기반추론(Rule-based reasoning)은 전문가의 경험적인 지식들을 규칙화하여 생성규칙으로 표현하여 발생한 문제에 대해 적용 가능한 규칙을 추출하여 추론과정을 진행한다. 하지만 지식에 대해서 규칙으로 정의하기 힘든 상황에는 적용하기에 제한되고, 전문가로부터 지식을 획득해야 하므로 지식의 확장성에 한계를 나타내고 있다. 본 논문에서는 규칙기반 추론과정에서 Stack형태의 구조인 Working memory를 사용하여 규칙기반추론을 효과적으로 수행하였으며, [그림 3]과 같다.



[그림 3] 규칙기반추론 구성도

- * **Knowledge Base:** 규칙화된 영역전문가의 경험적인 지식이 장치중심적으로 구조화하여 저장하였으며, 추론에 적합한 규칙을 제공해준다.

* **Knowledge Acquisition:** 새롭게 발생한 규칙이나 수정이 필요한 규칙에 대해서 수정 및 규칙화하여 지식베이스를 확장한다.

* **Inference Engine:** 적합한 지식들을 이용하여 추론 알고리즘을 통해 발생한 문제를 해결하며, 스택 형태의 구조인 Working Memory를 사용하여 추론한다.

2-2. Case-based reasoning (사례기반추론)

사례 기반 추론(Case-Based Reasoning)은 새로운 문제를 해결하기 위해 과거의 사례와 경험을 토대로 가장 유사한 사례를 찾아 문제를 해결하는 방법을 따르고 있다. 이러한 사례 기반 추론은 문제에 대한 해결 과정을 기록함으로써 지식 습득을 용이하게 하며, 탐색된 해를 저장함으로써 새로운 문제에 대해 대처할 수 있는 효율성을 가지고 있다. 또한, 입력된 사례에 대한 유사도를 측정하여 메모리로부터 유사한 사례들을 검색하여 새로운 문제를 해결할 수 있는 적용성을 가지고 있다. 하지만, 사례 기반 추론은 유사도를 근거로 이전에 사례들을 검색하는데 있어서 데이터베이스상에서의 사례 데이터 양이 방대하면, 검색기간이 길어지거나 부적절한 사례들을 유추할 수 있다는 단점이 있다. CBR의 Architecture는 4개의 모듈인 **RETRIEVE, REUSE, REVISE, RETAIN** 로 구성되어 있다.

i) **Retrieve Module:** 새로운 문제에 따른 입력 사례를 해결하기 위해 과거 사례의 지식 데이터베이스에서 과거 사례를 검색하고, 새로운 사례와 유사도를 평가하는 모듈이다.

ii) **Reuse Module:** 검색된 사례를 입력 사례에 대한 문제 해결에 적용하는 모듈이다.

iii) **Revise Module:** 검색된 사례의 해를 이용하여 문제를 해결하지 못했을 시 이를 분석하여, 실패의 원인을 찾아내고 제안된 해를 수정하는 모듈이다.

iv) **Retain Module:** 현재의 진단에 대한 과거의 고장 진단 및 경험을 활용하는 엔진으로 새로운 사례를 추출하여 문제에 대한 해결과정을 기록하여 지식을 습득을 한다.

입력된 사례와 저장된 사례간의 유사성을 어떤 형태로 정의할 것인가에 대해서 사례기반추론에서는 매우 중요한 문제이다. 실제 전문가들의 추론과정에 있어서 일관된 척도로 유사성을 측정해 내는데에는 많은 문제점이 있다. 따라서 일정한 측정기준을 선택하여 유사성을 측정해야 할 필요성이 있다. 본 연구에서는 각 사례간의 유사성을 측정해내는데 각 사례를 이루고 있는 증상으로서 유사도를 측정하는 기준을 선택하였다. 사용자를 통해서 각각의 특징들에 대한 문자열이 입력되고 그 문자열을 통해서 특징들의 고유한 키값이 검색된다. 이러한 고유의 키 값을

검색 후 검색된 키값을 포함하고 있는 사례에 대해서 검색이 실시된다. 이와 같은 검색이 이루어진 후 입력된 특징들과 검색된 사례의 특징들에 대한 유사도 측정을 하게 된다. 이런 과정에서 각각 특징들에 대한 유사도 Local-Similarity가 먼저 측정이 되고 그 유사도들의 합인 Grobal-Similarity가 측정이 된다. 이는 계층화구조로 구성하였으며, 불필요한 노드에 대한 검색을 실행하지 않아서 검색 시간을 단축할 수 있고, 유사도 측정에 있어서 두 단계의 유사성을 측정하기 때문에 보다 정확한 유사도 측정을 할 수 있다. 각각의 사례는 증상, 원인, 조치로 구성하였으며 검색모듈을 통해 유사한 사례를 검색한다. 본문에서는 K-NN(Nearest Neighbor) 방법을 통해서 사례의 유사성을 결정하는 방법을 이용하였다. 이는 입력된 사례의 특징과 저장된 사례의 특징을 순차적으로 비교하는 방법이다. 그러나 이 방법은 사례라이브러리의 양이 증가함에 따라서 비용도 증가 하는 단점이 있다. 이에 본 논문은 사례를 계층화 하여 입력하고, 각각의 특징들에 대한 색인 부여를 통해 검색의 효율성을 증가시켰다. 식 (1)에서 C^n 은 저장된 사례, C 는 입력된 사례, W_i 는 특징에 대한 가중치, V_C, V_{C^n} 는 입력, 저장사례의 특징들에 대한 평가값으로 정의된다. 식(2)는 K-NN 방법에서 입력된 사례 C 와 저장된 사례 C^n 의 유사도를 계산하는 식이다.

$$C = (S, C, R)$$

$$S : Symptom \quad C : Cause \quad R : Remedy$$

$$\min [\sum_i \{W_i * |V_i - Case_i|\}]$$

$$Sim(C, C^n) = \sum_a \omega(a) Sim_a \left(\frac{V_C(a)}{V_{C^n}(a)} \right) \quad (2)$$

입력된 사례의 증상을 검색된 사례의 증상과 비교하여 각 사례에 대해서 식(2)는 계층구조로 표현된 특징들에 대한 유사도를 측정한다. 식(2)에서 A_s 은 입력사례의 Single-level의 속성, A_m 은 Multi-level의 속성, A_s^n 은 저장 사례의 Single-level의 속성, A_m^n 은 Multi-level의 속성이 다. 1) 두 특징 중 하나의 Multi-level 속성이 없는 경우에는 유사도 Sim_a 를 0.5로 평가하였고, 2) 두 특징들이 Multi-level의 속성을 가지며 같은 경우에는 Sim 는 1로 평가하였다. 3) 두 특징들이 Multi-level의 속성을 가지지 않으며 Single-level이 같은 경우 Sim_a 가 1이고 4) 두 특징들의 Multi-level의 속성이 있으나 같지 않으면 저장된 사례의 Multi-level의 수로서 나눠준 값을 Sim_a 로 평가한다.

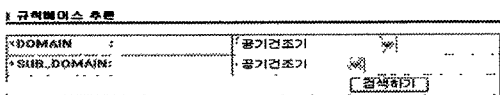
- 1) $A_m=0 \vee A_m^n=0$ $Sim_a = 0.5$
- 2) $(A_m \neq 0 \wedge A_m^n \neq 0) \wedge A_m = A_m^n$ $Sim_a = 1$
- 3) $(A_m=0 \wedge A_m^n=0) \wedge A_s = A_s^n$ $Sim_a = 1$
- 4) $(A_m \neq 0 \wedge A_m^n \neq 0) \wedge A_m \neq A_m^n$ $Sim_a = \frac{1}{(A_m^n count)}$ (3)

3. 실험

본 논문의 실험은 JSP프로그램 언어와 Oracle 9i를 사용하여 구현하였다. 실험은 크게 사례기반추론의 진단 과정과 규칙기반추론의 진단과정으로 분리하여 진행하였다.

3-1. 규칙기반추론의 구현

규칙기반추론에서의 지식은 규칙과 객체의 형태로 표현된다. 규칙기반추론은 현재 가장 많이 사용되고 있는 지식 표현 방법으로, 고장진단을 위한 규칙 또는 지식의 형태로 자연스럽게 표현할 수 있다. 본 논문에서는 규칙을 Domain과 Sub-domain 형식으로 계층적 구조로 구성하여 규칙베이스의 양이 방대해 질 경우 검색에 따른 손실을 줄임으로서 추론과정을 효율적으로 수행할 수 있다. Domain은 장치를 이루고 있는 주요구성품을 기반으로 구성하였고 Sub-domain은 주요구성품을 이루는 부품들로서 구성하였다. [그림 4]는 사용자 화면에서 Domain과 Sub-domain을 공기건조기로 선택하여 Domain을 공기건조기로 선택하고, Sub-domain은 공기건조기의 하위노드인 공기건조기로 선택하였다. 검색된 규칙들은 조건부에 대한 True / False의 선택에 의해서 규칙의 진단에 평가를 한다. 평가를 진행하는 과정에서 And 구조의 형태로서 규칙의 조건들 중 하나라도 만족하지 않는다면 다음 규칙에 대한 평가를 하게 된다. 조건을 만족했을 시에는 최종적으로 결론에 도달한 규칙기반추론의 진단과정을 설명한다 [9].



[그림 4] 규칙기반추론 인터페이스

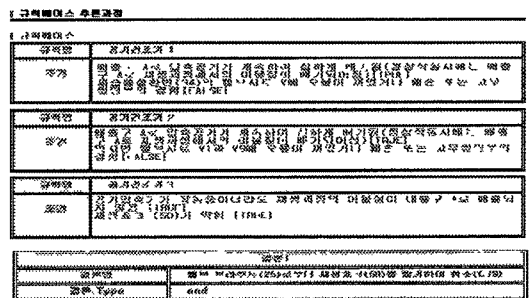
진단과정은 사전에 정의된 규칙에 의해서 수행하며 본 논문에서는 70개의 규칙을 규칙베이스에 저장하였다. 규칙의 결론은 인접목표(Immediately Goal), 추론완료(End)와 같은 두 가지의 형태로 구성하였는데 인접목표는 어떤 규칙의 결론이 다른 규칙의 첫 번째 조건과 동일할 수 있다. 예를들면 규칙(전공변환변 3)의 결론은 규칙(밸브 브

라켓트 수리)의 조건으로 구성되어있다. 이는 규칙(전공변환변3)의 조건에 대한 결론을 도출했을 경우, 결론에 대한 조치를 인접규칙(밸브 브라켓트 수리)에 적용하여 실행하는 결론이며, [그림5]와 같다.

규칙명	Rule 전공변환변 1
조건	If (전공변환변의 제어입력 <그림 1의 Cv>제어불가) Then (인접목표 생성 공칭장치 불행) (인접목표 불행) (완료)
결과Type	End
규칙명	Rule 전공변환변 2
조건	If (전공변환변의 제어입력 <그림 1의 Cv>제어불가) Then (압력변환기 불행) (압력변환기 불행) (완료)
결과Type	End

[그림 5] 규칙베이스의 예

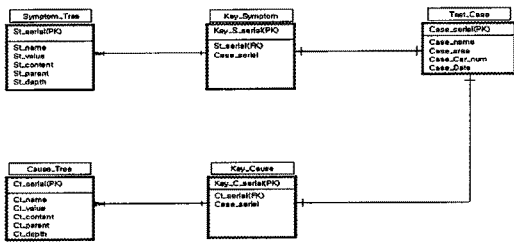
[그림6]은 [그림4]에서 입력값인 Domain-Sub Domain을 공기건조기로 선택하였을 경우 진단하는 과정이다. 공기건조기 1의 규칙이 생성되며 사용자는 규칙에 대한 질의 응답을 한다. 각각의 규칙이 사용자의 응답에 만족했을 경우 최종적으로 결론에 도달했을 경우 추론엔진은 결론을 제공해준다. 만족하지 않았을 경우에는 다음 규칙을 제공하며 사용자는 이에 대해 다시 질의응답을 하게 된다.



[그림 6] 규칙추론 설명

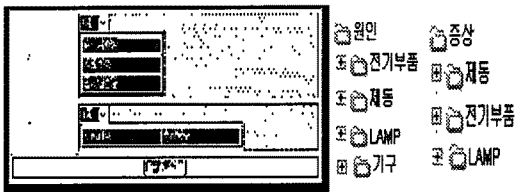
3-2. 사례기반추론의 구현

규칙기반 시스템의 개발시 지식이 증가함에 따라서 직면하게 되는 문제점들을 고려해 볼 때, 효율적인 사례 저장 및 검색 방법이 사례기반 시스템의 성공에 매우 중요하다 [5]. 따라서 본 논문에서는 사례 검색시 패턴 매칭의 양을 줄이기 위해서 각각의 증상, 원인을 Tree구조의 형태로써 각각 테이블로 사례베이스에 구축하였다. 증상, 원인의 특성들은 관계형 데이터베이스에서 Key_Symptom, Key_Cause 테이블의 하나의 튜플의 속성이 되며 Key_Symptom, Key_Cause의 특성들은 Test_Case의 튜플의 속성이 된다. [그림 7]은 Test_Case 테이블과 증상과 원인을 표현한 Tree테이블간의 관계형 데이터베이스 구조를 나타낸 DB구성도이다.



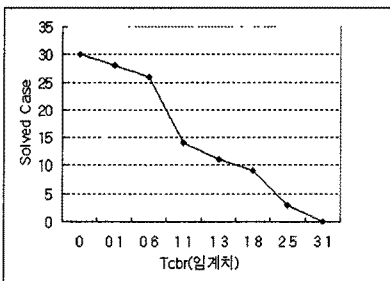
[그림 7] Case Library

[그림 8]은 사례기반추론과정에서의 특징에 대해 트리형태로 저장되어 있는 화면이다. 각각의 부품에 대한 노드를 생성하고 그 부품에 관련된 속성들을 하위노드로서 생성하였다.



[그림 8] 사례기반추론 입력화면

증상에 의해 검색된 사례 원인의 정확도를 다음과 같이 하였다. D_C 는 저장된 사례의 원인이고 D_C 은 입력 사례의 원인이다. 정확도를 검사하기 위해서는 식 (3)과 같이 유사성 검사 후 유사한 사례의 원인과의 검사를 통해서 정확도를 검사하게 된다. 식(4)는 원인이 같을 경우 해당 사례에 대해서 측정값에 1을 증가시킨다. 본 논문에서는 원인에 대한 측정값이 임계치보다 높다고 판단되었을 시 추론에 성공했다고 판단하여 문제에 대한 정확도 측정을 하였다. [그림 11]은 유사도 검사 후 각각의 정확도를 계산한 결과가 임의의 임계치를 정한 후 그 임계치보다 정확도가 높은 경우 해결된 사례(Solved Case)라고 정의하며 임계치가 높아짐에 따라서 해결된 사례가 반비례함을 알 수 있는 그래프이다.



[그림 11] 임계치에 따른 문제 해결 사례

표1 임계치에 따른 해결 사례의 수

임계치	Solved Case
0.1	28
0.6	26
1.1	14
1.3	11
1.8	9
2.5	3
3.1	0

$$\text{if } D_C = D_C$$

$$\text{Count} = \text{Count} + 1 \quad (4)$$

4. 결론

본 논문에서는 문제에 대한 보다 정확한 진단을 위하여 기존의 진단시스템보다 효율적이고 정확한 추론방식으로 규칙기반추론과 사례기반추론의 상호 보완적인 요소를 선택하여 혼합형 추론시스템에 대해서 제안하였다. 규칙기반추론에서는 규칙에 대해서 계층화 형태로 저장하여 적절한 규칙을 적용시키기 쉽게 하였으며, 규칙추론의 과정을 설명하는 기능을 추가하여 사용자에게 보임으로써 진단과정을 확인 및 질의응답을 통해서 좀 더 인지적인 설명을 취할 수 있게 규칙설명기능을 구현하였다. 사례기반추론에서는 사례에 대해 계층화 형태로 저장해서 지식의 양이 많아짐에 따라 검색의 시간이 증가하는 일반적인 검색 방법의 단점을 보완하였으며 입력된 문제에 대해 적합한 지식을 제공해 줌으로써 검색의 정확성을 높였다. 또한 유사도 검사를 통해서 검사된 유사사례의 원인을 통해서 사례의 정확성을 검사하였고, 임계치를 적용하여 해결된 문제에 대해서 신뢰성을 주었다. 그러나 자동적으로 수정? 경신해 나가는 부분에 대해서 좀 더 자동화하는 연구가 요구된다.

References

- [1] D. A. Waterman (1986). A Guide to Expert System. Addison-Wesley.
- [2] J. Dunkin (1994). Expert System Design And Development. Macmillan.
- [3] Golding, A R. & Rosenbloom, P.S. (1991).Improving Rule-Based Systems through Case-Based Reasoning AAAI-91, pp22-27
- [4]Watson, L., 1997, Applying Case-Based Reasoning, Morgan Kaufmann Publishers, Inc.
- [5] Bareiss, E.R., Porter, B.W. and Wier, C.C.,1988, "Protos: An Exampler-based Learning Apprentice," InternationalJournal of Man-Machine Studies, Vol.29,

pp549-561.

[6] Feret, M.P. and Glasgow, J.I., 1993, Hybrid Case-Based Reasoning for Diagnosis of Complex Device," Proc. of the National Conf. on Artificial Intelligence(AAAI-93), pp168-175

[7] Surma, J and Vanhoof, K, 1995, "Integrating Rules and Cases for the Classification Task," Proceeding of ICCBR-95, pp325-334.

[8] Lee, J. S. and Y. X. Xon, "A Customer Service Process Innovation using the Integration of Data Base and Case Base." Expert Systems with Applications, Vol. 11, No. 4 (1996), pp543-552

[9] M. C. Tanner A. M. Keuneke(1991). The roles of the Task Structure and Domain Functional Models. IEEE Expert System JUNE pp. 50-57.