

## 증상 트리 모델을 이용한 이상 진단 전문가 시스템

김철진 · 오전근 · 윤인섭

서울대학교 공과대학 화학공학과  
(1990년 2월 2일 접수, 1990년 5월 17일 채택)

### A Fault Diagnostic Expert System Using the Symptom Tree Model

Chul Jin Kim, Jeon Keun Oh and En Sup Yoon

Department of Chemical Engineering, College of Engineering, Seoul National University  
(Received 2 February 1990; accepted 17 May 1990)

#### 요 약

이상 트리 모델(fault tree model)을 기반으로 하는 증상 트리 모델(symptom tree model)을 이상 진단에 적합하도록 개조하였으며, 큰 공정에 적용 가능하도록 하였다. 또 공정의 정성적인 모델인 증상 트리를 전문가 시스템에 지식 기반(knowledge base)으로 사용하기 위하여, frames과 production rules를 적절히 혼합한 형태의 지식 표현 방법을 제시하였다. 이와 같은 지식을 적절히 이용하여 가설과 검증(hypothesis and test)에 의한 진단 전략을 제시하였다. 이와 같이 제시된 증상 트리의 지식 표현 방법과 진단 전략을 실제 공정인 나프타 분해로에 적용하여 이상 진단 전문가 시스템인 EXFAST를 개발하였다. 개발된 EXFAST를 실제 사고 사례와 가상 사고 사례에 대하여 수행시켜 본 결과, 이상 원인을 초기에 진단할 수 있는 것을 확인하였다. 이로써 공정의 정성적인 모델인 증상 트리가 이상 진단 전문가의 지식으로 사용하기에 적합함을 알 수 있었다.

**Abstract** — The Symptom Tree Model(STM) based on the Fault Tree Model (FTM) is modified to be appropriate in fault diagnosis and to be applied to large scale processes. Knowledge representation hybridizing frames and production rules is presented. It is used to implement knowledge base for fault diagnostic expert system with the STM of qualitative model for chemical processes. In order to perform fault diagnosis by using symptom trees on the real-time basis, a diagnostic strategy with hypothesis and test is presented. In order to apply the presented knowledge representation method and diagnostic strategy to naphtha furnace process, EXFAST(EXpert system for FAult diagnosis using Symptom Tree model) is developed. This developed EXFAST is tested and showed successful results. It is found that the STM is quite adequate as a qualitative knowledge model in fault diagnostic expert systems for chemical processes.

#### 1. 서 론

대부분의 화학 공정에 있어서 공정의 정보가 작동했을 때 조업자에 의해서 그 대책이 강구되어 진다. 이 때 고장 원인을 탐색하기 위하여 조업자가 사용하고 있는 정보는 정보의 종류, 상호 관련된 공정 변수들의

값, 조업자가 인식하고 있는 대상 공정의 정신적인 모델(mental model) 등이다[1]. 그러나 이러한 조업에서는 작업 교대, 조업자의 변동, 휴가 등의 이유로 조업자마다 추론을 달리할 수 있고 스트레스, 피로, 태만 등과 불완전한 기억력으로 인하여 부정확한 추론을 할 수 있다. 또한 공정이 병합화(integration)되고 제어 시스

템(control system)이 도입됨에 따라 이상 진단(fault diagnosis)을 더욱 어렵게 하고 조업자들에게 많은 양의 데이터, 특히 수치적 데이터(numerical data)를 처리해야 하는 부담을 주게 된다. 이러한 이유로 공정에 정보가 발생하였을 때 그 원인을 탐색하는 자동화된 이상 진단이 필요하게 되었다[2-4].

이 자동화된 이상 진단(automated fault diagnosis)은 60년대에 위험성이 높은 원자핵 발전소에서 정보를 정성적으로 해석하는 것으로 시작하여 70년대에 들어오면서 화학공정에 도입되기 시작하였다[5, 6].

그러나 이와 같은 정보 해석에 의한 이상 진단은 수치적 데이터를 처리하는 방법으로 사용되었기에 동특성을 가진 공정에서는 불합리한 결과를 가져올 수 있다. 그래서 80년대에 들어오면서 수치적 데이터, 지식 데이터(knowledge data)를 처리할 수 있을 뿐만 아니라 이런 데이터를 가지고 추론을 할 수 있는 능력을 가진 인공 지능(artificial intelligence)의 한 분야인 전문가 시스템(expert system)이 이상 진단에 도입되었다[2-4, 7]. 이 전문가 시스템을 구현하는데 가장 중요한 요소는 대상 영역의 지식을 적절히 표현하는 방법과 표현된 지식을 이용하여 진단을 행하는 전략이다[8, 9].

전문가 시스템내에서 대상 공정의 지식을 표현하는 방법은 조업자의 경험, 판단, 경험 법칙(heuristics)을 If-Then 룰(rule)로 표현하는 경험적 접근법, 공정의 정성적인 모델을 이용하는 방법과 공정의 정량적인 모델을 이용하는 방법 등이 있다. 그러나 화학 공정 이상 진단 전문가 시스템에서는 정성적인 모델과 정량적인 모델이 공정 지식을 표현하는 방법으로 널리 사용되었는데, 정성적인 모델로는 공정 변수 상호간의 관계를 digraph로 표시하여 이로 물을 표현한 SDG(Signed Directed Graph)[1], 공정을 계층구조로 표현한 모델[4, 10], 인공 신경 회로망(artificial neural network)을 이용한 모델[11], 공정 변수의 변화와 원인들간의 국부적인(local) 인과 관계를 표현한 SSTM(Symptom Sub-Tree Model)[12], 각 원인에 대한 거시적인(macroscopic) 인과 관계에 의한 증상 형태를 표현한 FCD(Fault Consequence Digraph)[19] 등이 있다. 정량적인 모델은 공정의 물질 수지식 또는 에너지 수지식 등과 제어기의 제어 방정식을 이용하여 공정에 대한 지식을 표현하는 방법과 측정 불가능한 상태 변수를 추정하는 방법 등이 있다[2]. 그러나 최근에는 경험적 지식 표현과 공정의 정량적 모델과 정성적 모델의 표현 방법을 혼합하여 공정 지식을 표현하고자

하는 방법이 일반적으로 사용되고 있다[13].

그리고 지식을 이용하여 문제를 해결하는 전략은 인식(recognition)과 유도(derivation)가 있다. 인식에 의한 문제 해결 방법은 가정과 검증(hypothesis and test)에 의해 행해지는 것으로 어떤 이상 원인을 가정하고 가정된 원인들을 차례로 검증해 나가는 방법이다. 유도에 의한 방법은 존재하는 지식을 이용하여 어떤 체계적인 추론 절차(reasoning procedure)를 이용하여 이상 원인을 찾아내는 방법이다[14-16].

본 연구의 목적은 이상 트리 모델(fault tree model)을 기본으로 하는 증상 트리 모델(symptom tree model)을 공정에 대한 정성적인 지식 모델로 사용하기 위하여 증상 트리를 지식으로 표현하는 방법을 제시하고, 이 지식을 이용하여 이상 진단을 행할 수 있는 진단 전략을 개발하여 실제 공정에서 on-line real time으로 이상 진단을 행할 수 있는지를 평가하는 것으로, 증상 트리 모델과 이를 이용한 이상 진단 전문가 시스템을 개발하는데 중점을 두고 있다.

## 2. 증상 트리 모델(Symptom Tree Model : STM)

증상 트리는 이상 트리를 기초로 하는 공정에 대한 정성적인 그래프 모델로서 1986년 Han 등[17]이 이것을 이용하여 전문가 시스템을 개발하였다. Han 등은 공정에 위험한 사건을 표시하는 공정 변수의 변화를 top event로 하여 이것을 일으키는 모든 원인과 공정 변수의 관계를 한 트리상에서 표시하고, 이것으로부터 트리에서 나타나는 공정 변수 변화에 대한 증상 트리를 얻을 수 있는 방법으로 트리를 표시하였다. 그러나 본 연구에서는 측정 가능한 공정 변수 변화에 대해 각각을 top event로 하여 이것에 영향을 주는 모든 가능한 원인들을 연결하는 증상 트리를 개발하여 이상 진단 전문가 시스템의 지식을 표현하는 모델로 사용하였다.

### 2-1. 사용 부호

증상 트리는 Boolean Logic으로 표현되어 있는 그래프 모델이므로 작성할 때 몇 가지 부호들이 사용되었는데 그것을 크게 두 가지로 나눌 수 있다.

#### 2-1-1. 논리 부호(logic symbols)

논리 부호란 top event로 정의된 위험한 사건에 영향을 미치는 상태 변수 또는 공정 변수와 원인들을 열거할 경우에 사용되는 Boolean Logic을 나타내는

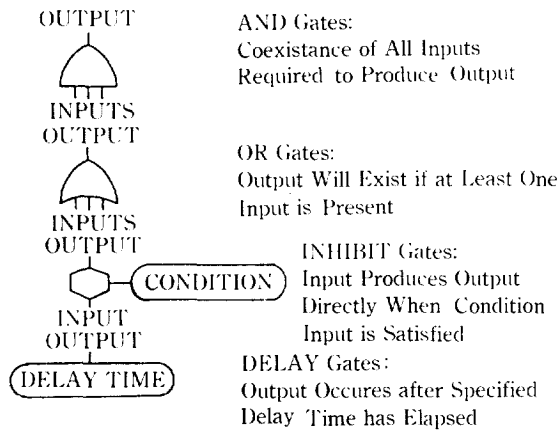


Fig. 1. Logic symbols.

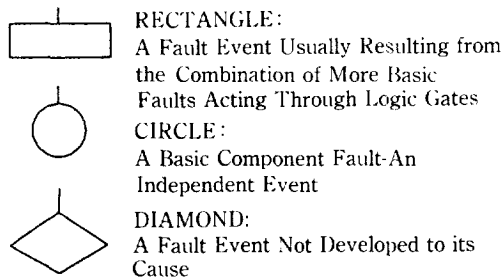


Fig. 2. Event symbols.

부호로서 주로 AND gate와 OR gate가 사용된다. 이 논리 부호들은 Fig. 1과 같다.

### 2-1-2. 사건 부호(event symbols)

공정의 상태, 변수의 상태와 원인들을 표시하기 위해 사용되는 부호로서 주로 RECTANGLE, CIRCLE, DIAMOND가 사용된다. 이 사건 부호들은 Fig. 2와 같다.

### 2-2. 작성 단계

이상 트리를 작성하는 단계는 다섯 단계로 나눌 수 있다.

첫 단계는 대상 공정의 영역을 확정하는 단계로 트리를 작성함에 있어서 트리가 공정 영역밖으로 확장되는 것을 막기 위함이다.

둘째 단계는 공정의 위험한 사건을 top event로 확정하는 단계로 이 top event를 방지하기 위해 트리가 작성되는 것이다.

셋째 단계는 공정 상태, 공정 변수와 원인들간의 인과 관계를 찾는 단계로 공정의 정상적인 해석을 하는 단계이다.

넷째 단계는 고려되어야 할 원인들의 범위를 확정

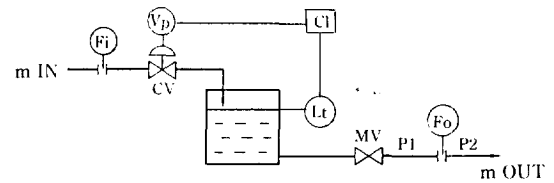


Fig. 3. A water tank with four sensors.

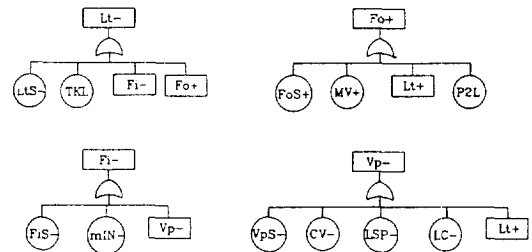


Fig. 4. Symptom sub-tree related with water tank level decrease.

하는 단계로 너무 세부적인 원인들을 고려하여 트리가 무한정 커지는 것을 막기 위함이다.

다섯째 단계는 고려된 원인들과 공정 상태와 공정 변수들간에 정상적인 관계를 부호들을 사용하여 트리를 완성하는 단계이다[18].

### 2-3. 증상 트리 작성

Fig. 3에 나타난 공정은 1982년 Yoon 등[12]과 1986년 Han 등[17]이 사용한 공정으로 4개의 측정 가능한 공정 변수가 있고, 조(tank)의 액위를 제어하기 위한 되먹임 제어 루프(feedback control loop)가 있는 간단한 공정이다.

#### 2-3-1. 증상 부 트리 모델(Symptom Sub-Tree Model : SSTM)

증상 부 트리 모델은 1982년 Yoon 등[12]이 이상 트리 모델로부터 개발하여 이상 진단에 사용한 모델로서, 공정 변수의 변화에 대한 국부적인(local) 인과 관계를 표현한 모델이다. Fig. 3의 공정에서 조 액위 감소에 대한 부 증상 트리는 Fig. 4와 같다.

Fig. 4에 나타난 바와 같이 조 액위 감소에 대한 부 증상 트리는 액위 감소에 국부적으로 미치는 영향들, 즉 액위 측정기 고장, 조 누출, 유입 유량 감소, 유출 유량 증가를 연결함으로써 트리가 완성된다. 이 트리에 나타나는 변수의 상태, 즉  $F_1^-$ ,  $F_o^+$ 에 대한 인과 관계는 각각을 top event로 하는 트리로 표현되어 진다. 여기서 (-)는 감소하는 현상을, (+)는 증가하는 현상을 나타낸다.

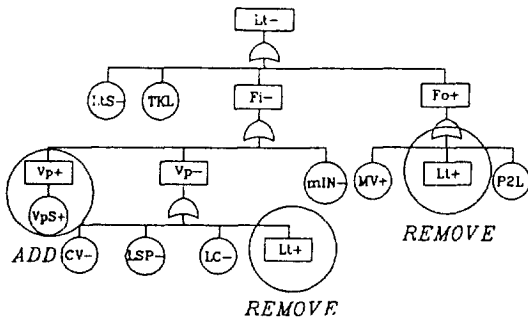


Fig. 5. Symptom tree for water tank level decrease.

### 2-3-2. 증상 트리 작성

증상 트리를 작성하기 위해서 몇 가지의 가정이 도입되었는데 그것은 다음과 같다.

1. 진단할 원인들이 top event로 정의한 공정 변수 변화에 대한 트리상에 존재한다.
2. 단지 하나의 원인이 공정 변수의 교란을 일으킨다.
3. 제어 루프가 있는 공정에서 이상 원인이 발전될 수 있는 경우는 제어기가 saturation된 경우이다.

위와 같은 가정을 도입하여 Fig. 3의 공정에 대해서 증상 트리를 작성하면 Fig. 5와 같다.

Fig. 5의 증상 트리는 Fig. 4의 증상 부 트리를 공정 변수 변화에 대해서 계속적으로 연결함으로써 얻어질 수 있다. 그러나 여기에서 고려되어야 할 요소는  $L_t^+$ 와  $V_p^+$ 에서  $V_pS^+$ 이다.  $L_t^+$ 의 경우, top event로  $L_t^-$ 가 발생하였다는 가정이 포함되어 있으므로 top event 사상에 위반된다. 이와 같이 위반된 사상은 제어 루프를 포함하는 공정에서 발생하게 되는데 이것을 제거함으로써 제어 루프를 해석하였다. 그리고 제어 밸브 위치 측정기 고장( $V_pS^+$ )의 경우, 유입 유량 감소( $F_i^-$ )를 일으키는 원인은 제어 밸브 위치 감소( $V_p^-$ )와 입력 유량 감소( $mIN^-$ )이나, 제어 밸브 위치 측정기 고장( $V_pS^-$ )도 실제로 유입 유량의 감소를 일으킬 수 있는 원인이 된다. 여기에서 국부적인 공정 변수들간의 인과 관계를 보면 제어 밸브 위치 감소가 유입 유량 감소를 일으키는 원인으로 고려될 수 있으나, 전체적인 인과 관계에서 제어 밸브 위치 증가( $V_p^+$ )도 유입 유량의 감소의 원인이 될 수 있다. 위와 같은 방법으로 제어 루프의 해석과 전체적인 인과 관계를 증상 트리 모델에 포함하여 증상 트리를 작성하였다.

## 3. 증상 트리 모델을 이용한 지식 표현 및 이상 진단 전략

### 3-1. 전체 공정 변수에 대한 frames의 구조

Fig. 3의 공정은 측정 가능한 공정 변수가 4개가 존재한다. 이 공정 변수 각각은 frame 구조로 나타낼 수 있다. 이것은 Fig. 6과 같다. Fig. 6에서 보면 Fig. 3의 공정을 WATER.TANK.VARIABLES라 하고, 그 아래에 각각의 공정 변수들을 연결하였다. 이 각각의 공정 변수는 여러 개의 slot들로 구성되어 있다. 여기에서 HIGH.FAULT.CANDIDATES는 공정 변수가 증가하는 정보가 올랐을 때의 증상 트리를 저장하는 slot이고, LOW.FAULT.CANDIDATES는 감소하는 정보가 올랐을 때의 증상 트리를 저장하는 slot이며, STATUS는 현재 공정 변수의 값이 설정된 alarm band를 어떤 방향으로 위반하였는지를 나타내는 slot이고 그 값은 NORMAL, HIGH, LOW의 값을 가질 수 있으며, TENDENCY는 현재 공정 변수의 값이 operation band를 어떤 방향으로 위반하였는지를 나타내는 slot이고 STATIC, INCREASED, DECREASED의 값을 가질 수 있다. Alarm band란 공정 변수가 이 범위를 위반할 때 공정 상태가 심각한 상태에 있는 것을 나타내는 범위이며, operation band는 alarm band보다는 작지만 공정 변수가 어떤 원인에 의해서 교란되어지는 범위를 나타낸다.

### 3-2. 증상 트리의 production rules로 표현

Fig. 3의 공정에서 측정 가능한 공정 변수가 4개가 존재하는데 각각의 변수에 대해서 증가와 감소에 대한 증상 트리가 작성되었다. 이 작성된 증상 트리는 식 (1)과 같은 방법으로 production rule로 표현되었고, 공정 변수의 frame에 저장되었다.

```
(setq sympmap (make-hash-table))
(setf (gethash 'hfi sympmap) '((fis+ min+)
                                (cv+ vps-) (tkl mv+) (lc-) (lsp-))
      (gethash 'lfi sympmap) '((fis-) (lc+) (lsp-))
      (mv-) (lc+) (lsp-))
      (gethash 'hvp sympmap) '((vps+ cv+) (lc+)
                                (vps+ tkl) (mv+ min-))
      (gethash 'lvp sympmap) '((vps- cv-) (lc-)
                                (mv+ min-))) (1)
```

식 (1)에서 보면 원인 후보들이 괄호로 묶여져 표현되었는데 이것은 원인 후보에 대한 증상 형태(symptom pattern)를 산출할 때 사용되었는데 자세한 것은 3-5절에 설명되었다.

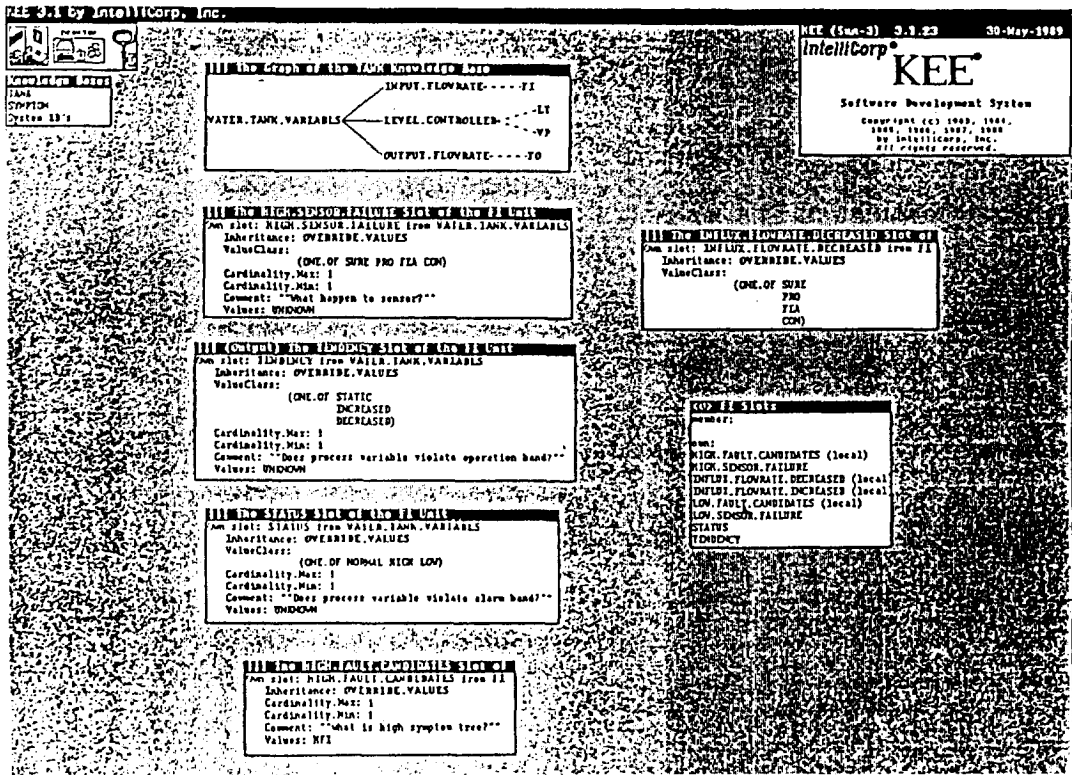
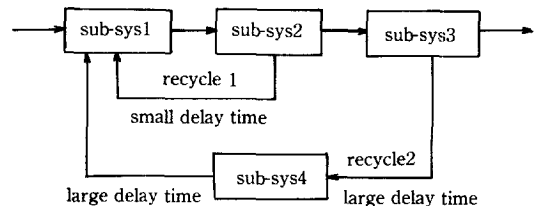


Fig. 6. Frames structure for water tank process.

### 3-3. 큰 공정에 대한 clustering

Clustering이란 증상 트리 모델을 큰 공정에 대해서 적용할 경우 트리를 작성함에 있어서 공정 자체가 크므로 전 공정을 하나로 고려한다면 트리가 매우 커지게 되는데, 이것을 방지하기 위해 전체 공정을 적절한 기준에 의해서 분류함을 말한다. 공정을 여러 개의 cluster로 나누는 기준은 공정의 sub-system간의 지연 시간(delay time)과 상호 작용(interaction)이다.

예를 들어 Fig.7과 같이 4개의 sub-system으로 된 공정에 대해서 이 기준을 적용해 보면, 2개의 recycle에 의해서 전체 공정은 심한 interaction이 생기게 된다. 그러나 recycle 1은 시간 지연이 적으므로 sub-system 1과 sub-system 2를 나눌 수 있는 상태 변수가 없게 된다. 그러므로 sub-system 1과 sub-system 2는 하나의 cluster로 묶여져야 한다. 그러나 sub-system 4를 거치는 recycle 2는 큰 시간 지연을 가지므로 서로의 상호 작용을 감소시킬 수 있으므로 sub-system 4를 하나의 cluster로 분류할 수 있다. 이와 같이 전체 공정을 여러 개의 cluster로 분류하여 각 cluster에서 증상 트리를 작성하게 되는데, 이 과정에서 연결된 각 cluster간의



Cluster 1: sub-sys1, sub-sys2

Cluster 2: sub-sys3

Cluster 3: sub-sys4

Fig. 7. Clustering of large process.

관계를 나타낼 수 있는 방법이 제시되어야 한다. 그러므로 본 연구에서는 서로 연결된 cluster간의 관계를 Fig.8에서 보는 바와 같이 cluster 1과 cluster 2간에 가상적인 NODE를 설정하여 NODE에 포함되어 있는 상태 변수를 점검함으로써 각 cluster간의 관계를 표현할 수 있다.

증상 트리 모델을 이용한 전체적인 진단 전략은 가설과 검증(hypothesis and test)에 의해 행해졌는데, 이 방법은 어떤 공정 변수에 위반(violation)이 발생

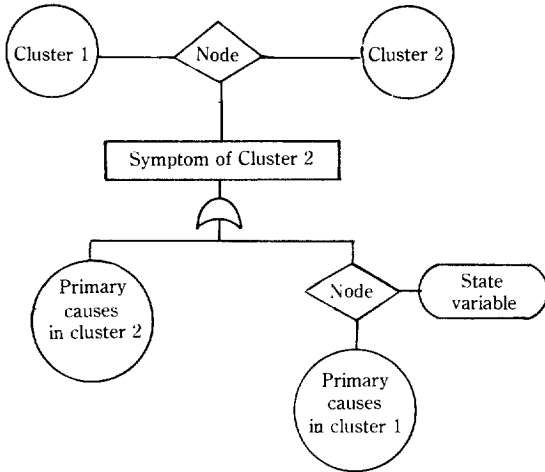


Fig. 8. Example of relationship between clusters.

하였을 때 이것으로부터 이 위반을 일으키는 원인 후보(fault candidate)들을 가정한 다음, 제시된 원인 후보에 대해서 원인 타당성 검토(fault simulation)에 의하여 가정된 원인들이 현재 공정 변수의 변화에 부합되는지를 판단하여 원인 후보에 대해서 상태를 나타내는 값으로 산출함으로써 행해지는 방법이다[19]. 원인 후보 제시 전략은 위반이 된 공정 변수들이 나타나게 되면 위반된 공정 변수들이 top event로 설정되어 있는 증상 트리에 해당하는 production rule들을 찾아내게 된다. 그리고 이 rule들을 작용시켜 공정 변수 위반에 대해서 원인 후보군을 제시하는 방법은 다음과 같다.

$$HPV1=[A, B, C, D, E]$$

$$HPV2=[B, D, F, G]$$

$$LPV3=[A, B, D, F, H, I]$$

$$\alpha_1 < PV1 < \beta_1, \alpha_2 < PV2 < \beta_2, \alpha_3 < PV3 < \beta_3 \quad (2)$$

식 (2)에서 HPV1과 HPV2는 PV1과 PV2의 공정 변수가 높은 상태일 때의 이상 원인들을 포함하고 있는 증상 트리를 나타내고 LPV3는 PV3가 낮은 상태일 때의 이상 원인들을 포함하는 증상 트리를 나타낸다. 또  $\alpha$ 와  $\beta$ 는 각각의 변수에 대한 alarm band를 표시한다. 만약에 PV1과 PV2가  $\beta_1$ 과  $\beta_2$ 보다 높은 값을 갖게 되고 PV3가  $\alpha_3$ 보다 낮은 값을 갖게 될 때, 이상 원인 후보군은 다음과 같이 구해진다.

$$HPV1 \cap HPV2 \cap HPV3 = [B, D] \quad (3)$$

식 (3)에서 보듯이 위반된 공정 변수에 대한 원인들의 교집합(intersection)을 구함으로써 HPV1, HPV2와 LPV3가 나타날 때에 원인 후보군으로 B와 D를

제시하게 된다.

### 3-5. 원인 후보 타당성 검토(fault simulation) 전략

앞 절에서 설명한 바와 같이 이상 진단이 가설과 검증으로 행해졌는데 원인 후보 타당성 검토는 검증에 해당하는 부분으로 원인 후보군이 제시된 후에 각 원인 후보가 현재 공정 변수들의 교란(disturbance)에 부합하는지를 판단하는 것을 말한다. 다른 말로 표현한다면 원인 후보를 제시하는 전략은 경보 해석(alarm analysis)이라 할 수 있고 원인 후보에 대한 타당성 검토는 교란 해석(disturbance analysis)이라 할 수 있다[6]. 이 타당성 검토를 행하기 위해서는 각 원인 후보에 대한 증상 형태(symptom pattern)가 필요하게 되는데 증상 트리 모델은 이상 원인을 기초로 하는 모델이 아니라 공정 변수 위반에 대한 연역적인 모델이므로 증상 형태가 원인에 대해서 표현되어 있지 않다. 그래서 증상 트리 모델을 가지고 이상 원인에 대한 증상 형태를 산출할 수 있는 방법이 필요하다. 증상 형태를 산출하는 방법은 원인 후보군으로 제시된 각각의 원인에 대하여 그 원인이 어떤 증상 트리에 포함되어 있는지를 판단하여 포함되어 있는 각 증상 트리에 대해서 설정되어 있는 top event를 가져오게 된다. 이 top event들은 그 원인이 일으킬 수 있는 증상들을 포함하고 있다. 이렇게 증상 형태를 산출하면 이상 원인에 대해서 나타나는 증상들의 순서(sequence)에 대한 정보를 잃어버리게 되는데 이것을 보완하기 위해서 식 (4)와 같이 각 증상에 대해 원인들이 미치는 순서를 괄호로써 분리하였다.

$$HPV1=[(A, B), (C), (D)]$$

$$HPV2=[(C), (D, E, F), (A)]$$

$$LPV3=[(E, F), (A, B, C), (G, H)]$$

$$HPV4=[(A), (B, D), (E, F), (C, I)] \quad (4)$$

식 (4)와 같은 공정 변수 위반이 주어졌을 때 원인 후보군은 [A, C]이고 A와 C에 대한 증상 형태는 식 (5)와 같다.

$$A=[(HPV1, HPV4), (LPV3), (HPV2)]$$

$$C=[(HPV2), (HPV1, LPV3), ( ), (HPV4)] \quad (5)$$

식 (5)에서 보면 A의 이상 원인이 각 공정 변수 위반에 대한 증상 트리가 나타내고 있는 list 구조상의 어떤 위치의 괄호에 나타나는지를 판단하여 증상 형태를 산출하게 된다. 이와 같은 방법으로 산출된 증상 형태를 가지고 현재 공정 변수의 교란과 비교하는 pat-

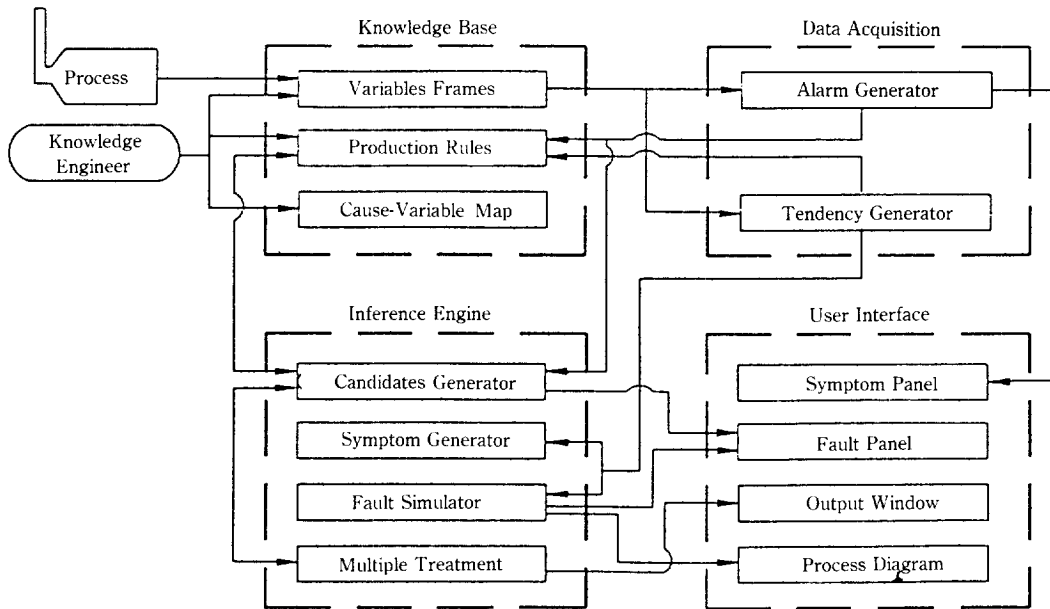


Fig. 9. Overall structure of EXFAST.

tern matching[14,16]을 행하게 된다. 이 pattern matching을 행하는 방법은 다음과 같다. 우선 각 원인들에 대하여 꼭 나타나야 할 증상들을 산출하고 현재 공정의 상태와 비교한다. 그리고 각 원인 후보에 대하여 나타날 증상들의 list를 산출하여, 이로부터 나타나지는 안되는 증상들을 산출하게 된다. 이것을 현재 공정의 변수들의 상태와 비교하여 위반되는지를 검토한다. 만약 현재 공정 변수들의 상태가 하나의 원인에 대해서 산출한 첫번째 증상과 증상 형태에 대해서, 첫번째 증상이 위반이 있고 증상 형태에 위반이 있는 경우에 그 원인의 상태는 CON(conflict), 첫번째 증상이 위반이 있고 증상 형태에 위반이 없는 경우에는 PRO(proposed), 첫번째 증상이 위반이 없고 증상 형태에 위반이 있는 경우에는 FEA(feasible), 첫번째 증상이 위반이 없고 증상 형태에도 위반이 없는 경우 SURE(assured)의 값을 갖게 하여 pattern matching을 행하였다.

#### 4. EXFAST의 구현

EXFAST는 2장에서 설명한 STM을 이용하고 3장에서 설명한 진단 전략을 사용하여 나프타 분해로(naphtha furnace)에 대한 실시간(real time) 이상 진단을 수행하기 위해 개발된 전문가 시스템이다. EXFAST를 구현하기 위해서 Intellicorp사에서 개발된 KEE라는 tool을 사용하였고 UNIX 운영 체제(operating system)

에서 작동하는 SUN-3/260이 사용되었다.

##### 4-1. 나프타 분해로에 대한 증상 트리 작성

나프타 분해로는 열 효율을 높이기 위해 여러 스트림이 얹혀져, 서로간에 상호 작용(interaction)이 심하기 때문에 여러 개의 cluster로 나눌 수 없는 공정이다. 또한 제어 구조도 DCS와 DMC logic이 동시에 부착되어 있기 때문에 공정 변수들간에 상관 관계가 다른 어떤 공정보다 복잡하게 된다. 그러므로 증상 트리 자체도 복잡하고 커지게 된다. 나프타 분해로에 대한 STM은 측정기로부터 컴퓨터로 오는 정보를 이용하여 표현되었으며 대부분 OR gate로 표시되어 하나의 원인이 top event로 선정된 공정 변수에 영향을 미치는 성질(one-event minimal cut sets)[18]을 가진다. 나프타 분해로에서 컴퓨터로 입력되는 공정 변수는 75개인데 증상 트리로 표현된 공정 변수는 제어기상에서 계산되는 변수 38개를 합친 113개이다. 그러므로 113개의 공정 변수에 대해 높고 낮은 두 상태에 대응하는 두 개의 트리를 가지므로 총 226개의 트리로서 분해로의 모든 변수와 원인들의 관계를 표시하였다.

##### 4-2. EXFAST의 구조

EXFAST의 전체적인 구조는 Fig.9에서 보듯이 크게 네 부분, 지식 기반(knowledge base), 데이터 습득(data acquisition), 추론 엔진(inference engine)과 사

용자 환경(user interface)으로 나눌 수 있다.

#### 4-2-1. 지식 기반(knowledge base)

EXFAST의 지식 기반은 공정 변수를 계층적 구조로 표현하여 frames으로 구성된 Variables Frames, 증상 트리를 rules로 표현한 Production Rules과 원인과 공정 변수와의 관계를 표현하는 Cause-Variable Map으로 구성되어 있다.

#### 4-2-2. 데이터 습득(data acquisition)

데이터를 EXFAST에서 얻는 방법은 on-line real time으로 SUN-3/260과 IBM PC의 연결로 이루어져 있다. IBM PC에서 30초마다 하나의 data file을 SUN-3/260에 보내지게 되면 alarm band와 operation band를 벗어나게 되는 공정 변수들을 발견하여 Variables Frames에 있는 공정 변수에 해당하는 unit으로 보내지게 된다. 이 unit에는 STATUS slot과 TENDENCY slot이 있고 이 각각의 slot들은 자신의 값을 방법론으로 가지고 있다. 공정 변수에 이상이 있는 경우 이들 slot에 있는 방법론이 자동적으로 작동하는 OOP(Object Oriented Programming) 기법으로 alarm band를 위반한 공정 변수들의 list(symptom list)를 산출하고 operation band를 위반하는 공정 변수들의 list(tendency list)를 산출하게 된다.

#### 4-2-3. 추론 엔진(inference engine)

추론 엔진은 표현된 지식을 가지고 이상 진단을 행하는 핵심적인 부분으로 원인 후보군을 제시하는 Candidates Generator, 제시된 원인 후보에 대한 증상 형태를 산출하는 Symptom Generator, 타당성 검토를 행하는 Fault Simulator, 독립된 다중 이상을 처리하는 Multiple Treatment로 구성되어 있다.

#### 4-2-4. 사용자 환경(user interface)

본 연구에서 개발된 EXFAST의 사용자 환경은 symptom panel, fault panel과 공정도가 있다. Alarm band를 넘어서는 공정 변수가 나타나게 되면 OOP기법으로 자동적으로 변수의 상태를 나타내는 image가 symptom panel에 표시되고, 이 공정 변수 위반에 대해서 구해진 원인 후보군이 fault panel에 표시된다. 그리고 앞 절에서 설명한 각 원인 후보에 대한 타당성 검토가 행해져 원인 후보의 상태가 결정되면 이것이 자동적인 image로 fault panel에 나타난다. 또 원인 후보의 타당성 검토를 행한 결과 SURE 값을 갖게 되는 원인에 대해서는 원인이 공정에서 위치하는 부분을 공정도에 표시한다.

### 4-3. EXFAST의 전반적 진단 전략 흐름도

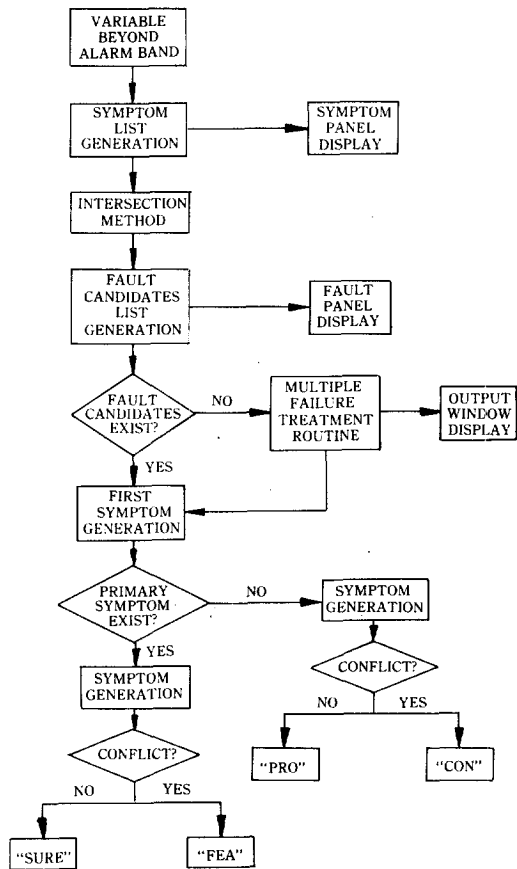


Fig. 10. Flowchart for strategy of EXFAST.

EXFAST의 모든 진단 전략을 전체적인 흐름도로 나타내면 Fig. 10과 같다.

## 5. 사례 연구

본 연구에서 나프타 분해로의 이상 진단을 위하여 개발된 EXFAST를 이용하여 여러 사고 사례와 가상 사고 사례에 적용하여 그 특성을 살펴봄으로서 증상 트리 모델을 이용한 이상 진단 전문가 시스템의 효용성을 살펴보았다.

### 5-1. 나프타 펌프 고장

이 사례는 실제 일어났던 사고 사례로서 나프타 분해로에 공급하는 펌프 고장이 나서 공정이 shutdown된 사고를 고려한 것이다. 이 때 변화된 공정 변수는 사고 일지를 조사해 본 결과, 펌프가 고장이 남에 따라 각 스트림의 나프타의 유량이 떨어지게 되고 이에 따라



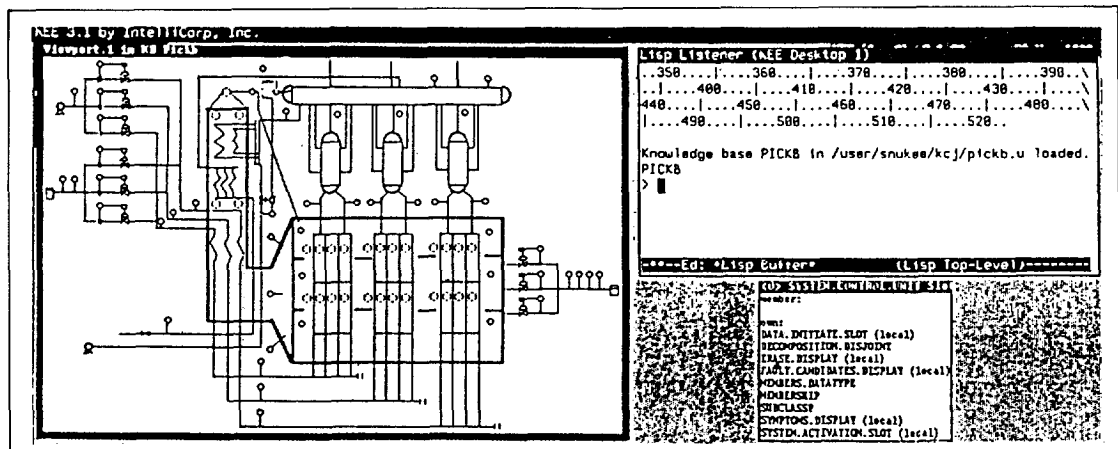


Fig. 11. Result for naphtha pump failure (time=0).

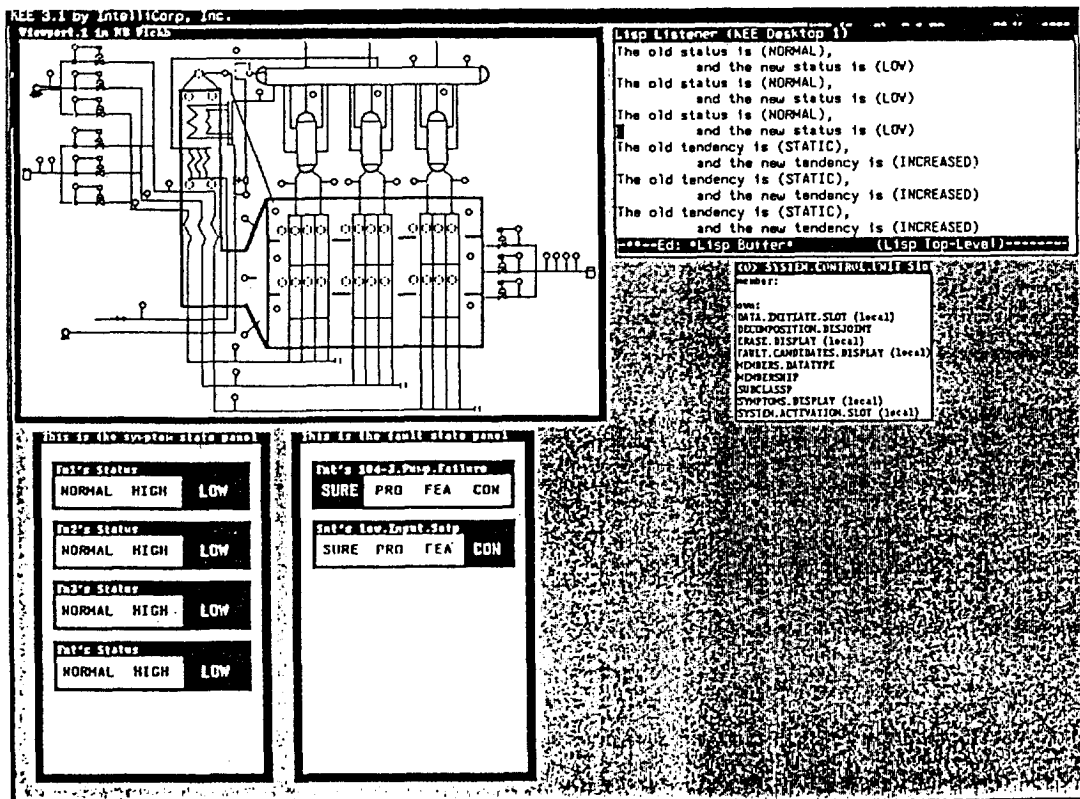


Fig. 12. Result for naphtha pump failure (time=1).

나프타 유량 제어기의 출력(output)이 급격히 증가하였으며 연료 가스의 압력이 조절되지 않아 대류 부분의 출구 온도와 복사 부분에 각 zone의 출구 온도가 증가하였다[20-22]. 이와 같은 현상을 포함하는 입력 데이터를 만들어 EXFAST를 수행하였다. EXFAST는

30초마다 실시간으로 진단이 수행되므로 각 30초마다 진단 결과를 출력한다. 아래에서 time으로 설정되어 있는 숫자의 시간 간격은 30초이다.

#### 5-1-1. time=0

시간이 0이라는 의미는 정상 조업 상태를 나타내는

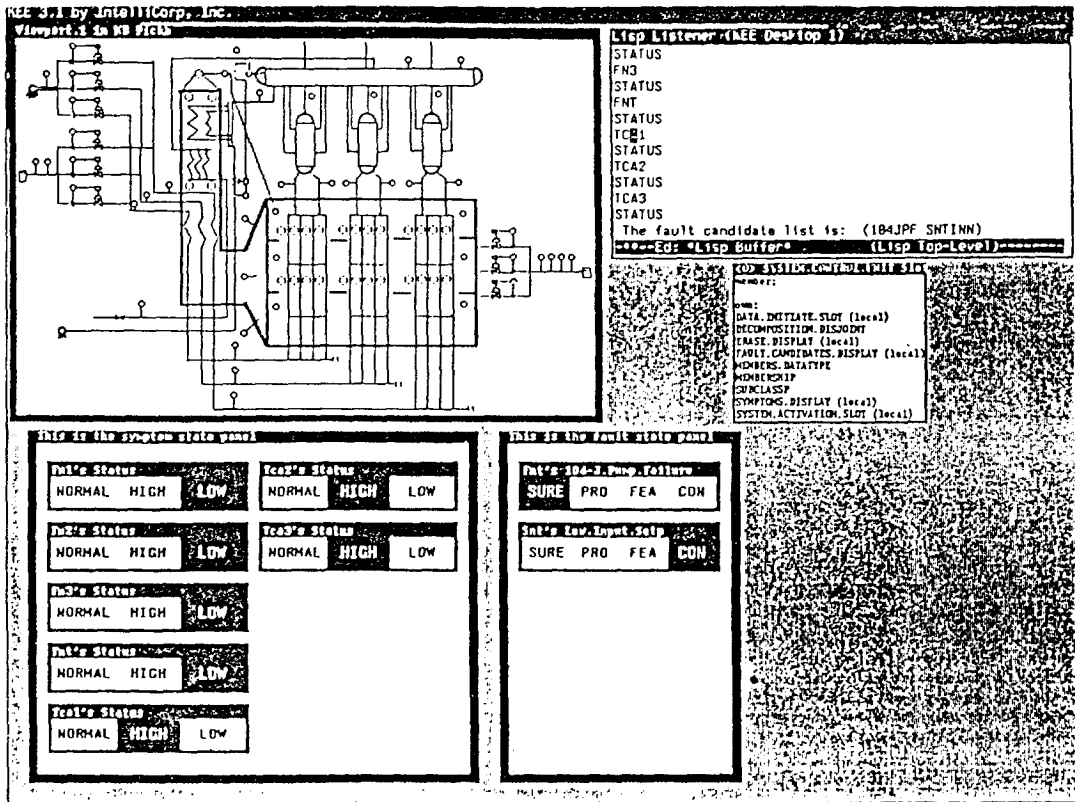


Fig. 13. Result for naphtha pump failure (time=2).

것으로 정상에서 나타나는 출력 형태는 Fig. 11에 나타나 있다. 이 그림에서 좌측 상단에 있는 것은 나프타 분해로의 공정도를 나타내고, 우측 상단에 있는 Lisp Listener는 EXFAST가 KEE라는 Lisp 언어로 개발된 tool 체제하에서 개발되었고, EXFAST 자체도 Lisp 언어로 쓰여졌기 때문에 Lisp와 interface할 수 있는 window이다. 그리고 Lisp Listener에서 현재 모든 지식 베이스가 load 되었음을 나타내고 있다. Lisp Listener 아래에 있는 window는 EXFAST의 모든 데이터를 초기화할 수 있는 DATA.INITIALIZE.SLOT, symptom panel과 fault panel을 화면에서 제거하기 위한 ERASE.DISPLAY, 제시된 원인 후보군을 fault panel에 표시하는 FAULT.CANDIDATES.DISPLAY, 현재 alarm band를 위한 공정 변수 상태를 나타내는 SYMPTOMS.DISPLAY, 진단 시스템을 가동시키는 SYSTEM.ACTIVATION.SLOT 등의 slot들을 가지고 EXFAST를 전체적으로 관리하는 SYSTEM.CONTROL.UNIT을 나타내는 것이다. 앞에서 언급한 모든 slot들은 OOP 기법으로 이루어져 있기에 message를 보냄에 따라 그

자체가 가지고 있는 방법론을 작동시켜 자동적으로 모든 절차를 수행하게 된다.

#### 5-1-2. time=1

정상적인 조업 중에 있던 공정이 펌프의 이상으로 인하여 공정 변수들의 변화를 가져오고 이것을 인식한 EXFAST가 처음으로 작동되는 시간으로, 그 결과는 Fig. 12에 나타나 있다. 나프타 펌프의 이상으로 각 스트림의 나프타 유량에 정보가 올리고, 총 나프타 유량에도 정보가 올랐다. 이로 인해 진단 시스템이 작동되어 symptom panel에 공정 변수의 상태를 나타내고, 원인 후보를 제시하는 시스템이 작동하여 fault panel에 2개의 원인 후보, 나프타 펌프 이상과 나프타 총 유량 제어기의 setpoint를 낮게 설정한 것을 제시하고 있다. 그리고 operation band에 위반이 있는 공정 변수들을 가지고 원인 후보 타당성 검토가 이루어져, 나프타 펌프 이상은 SURE, setpoint의 낮은 설정은 CON의 값을 주고 있다. 이로서 EXFAST는 나프타 펌프 이상이 유력한 원인임을 제시하기 위하여 공정도에서 나프타 펌프에 해당하는 부분을 초기의 그림보다 두꺼운 선

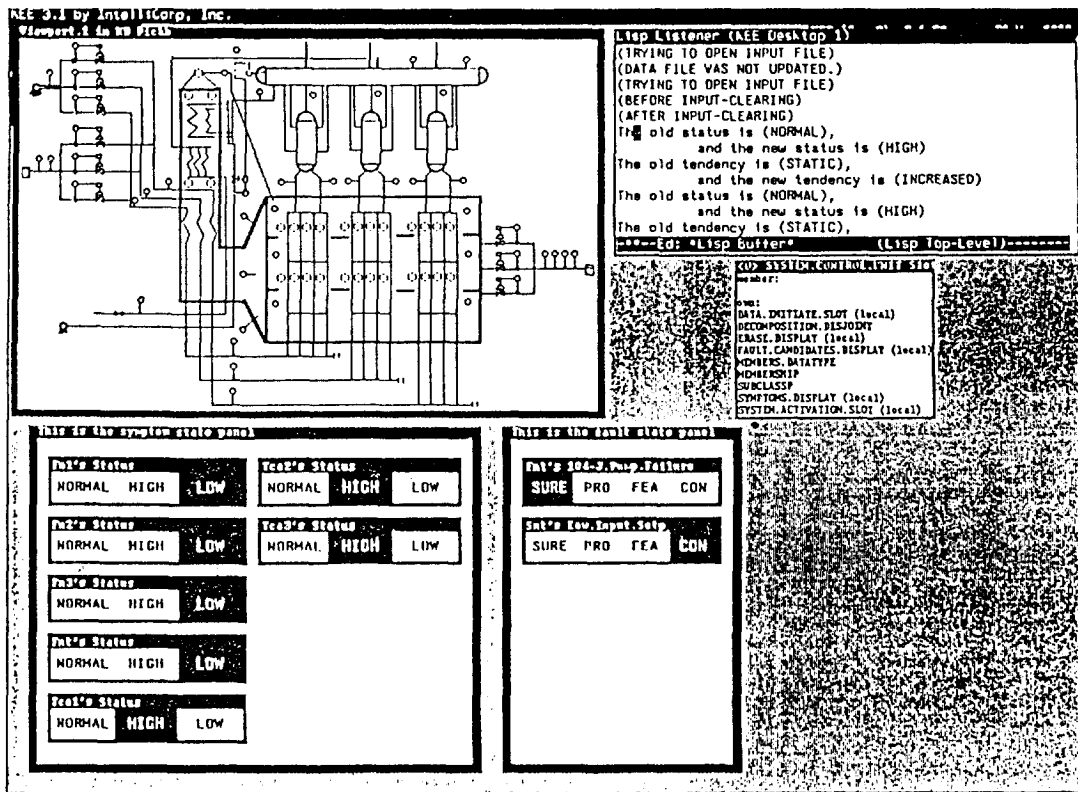


Fig. 14. Result for naphtha pump failure (time=3).

으로 표시하고 있다.

#### 5-1-3. time=2

이상이 조치가 취해지지 않고 계속 발전하는 단계로 Fig. 13에 출력이 나타나 있다. 출력 형태는 전 단계와 동일하나 단지 더 많은 공정 변수에 정보가 올려 symptom panel에 나타난 공정 변수의 위반이 전 단계보다 많아졌다. 그러나 타당성 검토 결과는 전 단계와 같다.

#### 5-1-4. time=3

시간이 더 경과하였으나 그 결과는 전단계와 같다. 이것은 Fig. 14에서 보여지고 있다.

## 6. 결과 검토 및 제언

증상 트리를 지식으로 표현하는 방법과 이를 이용하여 이상 진단을 행할 수 있는 전략에 대해 연구하여 실제 공정에 대해 적용하였다. 실제 공정은 나프타 분해 공정에서 가장 중요한 부분이며 수학적 모델을 세우기가 어려운 나프타 분해로이다. 이 공정에 대하여 많은 조업자의 의견을 수렴하여 증상 트리가 작성되었으며,

이것을 이용하여 지식 기반을 구축하고 적절한 진단 전략을 포함하는 이상 진단 전문가 시스템인 EXFAST가 개발되었다. 이 EXFAST의 진단 효율성을 알아보기 위하여 많은 실제 사고 사례와 가상 사례에 적용하였다.

위와 같은 연구로 다음과 같은 토론과 제안을 할 수 있었다.

1. 증상 트리는 공정의 정성적인 관계로부터 이루어지고, 이로부터 production rule 형태의 지식을 직접적으로 얻을 수 있기 때문에 공정 지식을 표현하기 쉬운 모델이다.
2. 증상 트리의 지식 표현이 룰(rule) 형태로 되어 있기 때문에 실제 공정에서 조업자들이 가지는 경험 법칙(heuristics)을 쉽게 포함시킬 수 있다.
3. 증상 트리를 큰 공정에 적용하기 위해서 공정을 여러 개의 cluster로 나누고 cluster 내에서 증상 트리를 작성하기 때문에 큰 공정에서도 적용 가능하다.
4. 정보가 올린 공정 변수들에 대한 원인 후보군의 제시는, 공정 변수의 위반에 대해서 증상 트리에 포함

되어 있는 원인들의 교집합을 구하는 간단한 구조로 되어 있기에 신속한 원인 후보 제시를 행할 수 있었다.

5. 이상 진단을 행하는 전반적인 전략이 가설과 검증(hypothesis and test)에 의한 문제 해결 방법으로 행하여 졌기에, 진단 효율성을 높일 수 있었다.

6. 열 효율을 위하여 스트림이 복잡하게 얽혀있고 DCS 제어 구조와 DMC(Dynamic Matrix Control) logic이 있는 나프타 분해로에 대한 사고 사례와 가상 사고 사례에 대해, 사고 원인을 초기에 찾아낼 수 있었다.

7. 증상 트리는 작성을 하는 사람에 따라 달라질 수 있고, 작성하는 시간이 많이 걸리므로 공정의 정성적인 모델식으로부터 자동적인 작성이 필요하다.

8. 증상 트리는 이상 트리를 기반으로 하는 모델이므로, 트리의 안정성 해석을 통한 정량적인 해석으로 확률적 데이터를 포함하고 있다. 그러므로 확률적 데이터를 이용한 이상 진단을 행하고자 할 때, 이것에 대한 적절한 진단 전략이 필요하다.

위와 같은 검토와 제안을 보완함으로써, 증상 트리 모델은 앞으로 많은 분야에 적용되어질 전문가 시스템의 지식 모델로 널리 사용되어질 것이다. 또한 증상 트리 모델에 많은 연구가 이루어진다면, 대상 영역에 대한 지식을 더욱 정확하게 표현할 수 있을 것이다.

## NOMENCLATURE

CV	: control valve
$F_i$	: flowrate (in)
$F_o$	: flowrate (out)
LC	: level controller
LSP	: level setpoint
$L_t$	: tank level
MV	: manual valve
mIN	: feed in
P2L	: pipe 2 leakage
S	: sensor
TKL	: tank leakage
$V_p$	: valve position

## REFERENCES

1. Kramer, M.A. and Palowitch, B.L.: "A Rule-Based Approach to Fault Diagnosis using the Signed Directed Graph", *AIChE J.*, **33**(7), 1067 (1987).
2. Kramer, M.A.: "Malfunction Diagnosis using Quantitative Models with Non-Boolean Reasoning in Expert Systems", *AIChE J.*, **33**(1), 130 (1987).
3. Rich, S.H. and Venkatasubramanian, V.: "Model-

Based Reasoning in Diagnostic Expert Systems for Chemical Process Plants", *Comput. Chem. Engng.*, **11**(2), 111 (1987).

4. Shum, S.K., Davis, J.F., Punch, W.F. and Chandrasekaran, B.: "An Expert System Approach to Malfunction Diagnosis in Chemical Plants", *Comput. Chem. Engng.*, **12**(1), 27 (1988).
5. Andow, P.K.: "Real-Time Analysis of Process Plant Alarms using a Mini-Computer", *Comput. Chem. Engng.*, **4**, 143 (1980).
6. Lees, F.P.: "Process Computer Alarm and Disturbance Analysis: Review of the State of the Art", *Comput. Chem. Engng.*, **7**(6), 669 (1983).
7. Kumamoto, H., Ikenchi, K. and Inoue, K.: "Application of Expert System Techniques to Fault Diagnosis", *The Chemical Engineering Journal*, **29**, 1 (1984).
8. Buchanan, B.G. and Shortliffe, E.H.: *Rule-Based Expert Systems*, Addison-Wesley, Inc., (1985).
9. Wolfgram, D.D., Dear, T.J. and Galbraith, C.S.: *Expert Systems for the Technical Professional*, John Wiley.
10. Ramesh, T.S., Shum, S.K. and Davis, J.F.: "A Structured Framework for Efficient Problem Solving in Diagnostic Expert Systems", *Comput. Chem. Engng.*, **12**(9, 10), 891 (1988).
11. Hoskins, J.C. and Himmelbrau, D.M.: "Artificial Neural Network Models for Knowledge Representation in Chemical Engineering", *Comput. Chem. Engng.*, **12**(9), 881 (1988).
12. Yoon, E.S., Evans, L.B. and Tooney, T.B.: "Process Diagnostics using the Symptom Sub-Tree Model", Ph.D. Thesis, Massachusetts Institute of Technology, (1982).
13. Venkatasubramanian, V. and Rich, S.H.: "An Object-Oriented Two-Tier Architecture for Integrating Compiled and Deep-Level Knowledge for Process Diagnosis", *Comput. Chem. Engng.*, **12**(9, 10), 903 (1988).
14. Charniak, E. and McDermott, D.: *Introduction to Artificial Intelligence*, Addison-Wesley, Inc., (1985).
15. Nilson, N.J.: *Principles of Artificial Intelligence*, Springer-Verlag, Inc., (1982).
16. Winston, P.H.: *Artificial Intelligence*, Addison-Wesley, Inc., (1982).
17. Han, J.H. and Yoon, E.S.: "A Study on the Knowledge-Based Expert System for Cement Process Control and Diagnosis", M.S. Thesis, S.N.U., (1986).
18. Fussell, J.B.: "Fault Tree Analysis-Concepts and Technique", NATO Adv. Study Inst. on Generic

- Techniques in Systems Reliability Assessment, England, (1973).
19. Oh, Jeon Keun and Yoon, En Sup: "A Real-Time Operation Aiding System of Naphtha Furnaces", AIChE Annual Meeting, San Francisco, (26 C), (1989).
20. 공정 사고 분석, 주식회사 유공, (1987)
21. Unit 사고 분석, 주식회사 유공, (1986)
22. 사고 사례 분석, 주식회사 유공, (1988)