

## 증상 트리 모델을 이용한 이상 진단 전문가 시스템

김 철진\*, 윤 인섭

서울대학교 공과대학 화학공학과

Fault 트리를 기초로 하는 증상 트리 모델을 나프타 열분해로에 대한 지식 모델로 적용하여 이상 진단 전문가 시스템의 지식 베이스로 사용하였다. 이로서 전문가 시스템을 구현하여 실제 공정의 사고 사례와 가상 사고 사례에 적용한 결과 적절한 원인군의 신속한 제시와 각 원인들의 가능성 여부를 판단하는데 적은 시간이 소요되므로 실시간 이상 진단에 적합한 모델임을 확인하였다.

## 1. 서 론

대부분의 화학 공정에 있어서 공정의 정보가 작동했을 때 조업자에 의하여 그 대책이 강구되어 진다. 이때 고장 원인을 탐색하기 위하여 조업자가 사용하고 있는 정보는 경보의 종류, 상호 관련된 공정 변수들의 값, 조업자가 인식하고 있는 대상 공정의 정신적인 모델(mental model)등 이다. 이러한 조업하에서 조업자들은 각각 나름대로의 경험 법칙(heuristics)을 가지고 공정의 경보에 대해 그 원인을 추론하고 대책을 수립하게 된다. 그러나 이러한 조업에서는 작업 교대, 조업자의 변동, 휴가등의 이유로 조업자마다 추론을 달리할 수 있고 스트레스, 피로, 태만등과 불완전한 기억력으로 인하여 부정확한 추론을 할 수 있다. 또한 조업자들은 많은 양의 데이터, 특히 수치적 데이터를 다루는 데에 많은 어려움이 있으므로 컴퓨터를 이용한 이상 진단이 필요하게 되었다. 이상 진단을 하는 시스템을 구현하기 위해서는 수치 데이터 뿐만 아니라 공정에 대한 지식 데이터가 필요하고 이것을 처리 하기 위해서는 추론을 할 수 있는 능력도 필요하게 된다. 그러므로 이상 진단을 위해서 인공 지능의 한 분야인 전문가 시스템에 대한 연구가 시작되었다.

이상 진단 전문가 시스템을 구축하기 위해서는 지식 베이스(knowledge base)가 필요한데 이것을 구성하는 방법은 다양하게 존재한다. 이 지식 베이스는 전문가 나 운전자의 경험에서 얻는 얇은 지식(shallow knowledge)으로 구성될 수 있으나 추론의 효율화를 기하기 위해 공정에 대한 지식 모델로서 표현하는 방법이 연구 되었다.

지식 모델로는 Venkatasubramanian등[8]이 단순한 경험 법칙을 트리 형태로 표시하여 Yes, No의 질문으로 이상 원인 (failure cause) 를 찾아 가는 방법, Kramer등[6]이 공정 변수 상호간에 관계를 digraph로 표시하여 이로 부터 법칙(rule)을 표현한 SDG(Signed Directed Graph),

Davis등[9]이 공정의 경험 법칙을 계층(hierarchy) 구조로 표현한 모델, Himmelbrau등[4]이 Neural Network를 이용하여 지식을 표현하는 모델과 Yoon등[5]이 공정 변수의 변화와 원인들간에 인과 관계를 표현한 STM(Symptom Tree Model) 등이 있다.

본 연구에서는 STM을 공정 지식 표현 모델로 사용했으나 기존의 STM 을 표현하는 방법에 있어서 순서적인(sequence) 의미를 부여하고 sub-tree형태로 표현하지 않고 변수에 대한 트리 전체를 지식으로 표현하는 방법을 제시하였다.

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

증상 트리 모델은 fault 트리를 기본으로 하여 top event를 공정 변수의 변화로 설정하고 이 변화에 영향을 줄 수 있는 변수와 원인들을 연결시키고 변수에 대해서는 원인들로만 연결되는 단계까지 논리적으로 표현한 그래프 모델(graph model)이다. 이 모델을 표현하는 방법은 논리 부호(logic gate)와 사건 부호(event symbol)가 근간을 이루고 있다. 작성하는 방법은 크게 다섯 단계로 분류할 수 있는데, 첫 단계는 대상 공정의 영역을 확정하여 무한정 트리가 확장되는것을 막는 것이고, 두 번째 단계는 top event로 설정할 변수를 선정하는 것이며, 세 번째 단계는 변수 상호간의 인과 관계를 표현하는 것이며, 네 번째 단계는 고려되어야할 원인들을 확정하여 너무 세부적인 원인들은 무시할 수 있는 범위를 정하는 것이며, 다섯 번째 단계는 변수와 변수 또는 원인들을 논리적으로 연결하는 것이다.

## (1) 사용 부호

증상 트리를 작성할 때 몇 가지 부호들이 사용되었는데 그것을 크게 두가지로 나눌 수 있다.

### (가) 논리 부호(logic symbol)

논리 부호란 top event로 정의된 공정 변수에 대해 영향을 미치는 변수와 원인들을 연결할 경우에 사용되는 부호로서 주로 AND gate 와 OR gate가 사용된다.

### (나) 사건 부호(event symbol)

변수의 상태나 원인들을 표시하기 위해 사용되는 부호로서 주로 RECTANGLE, CIRCLE, DIAMOND로 표시되었다.

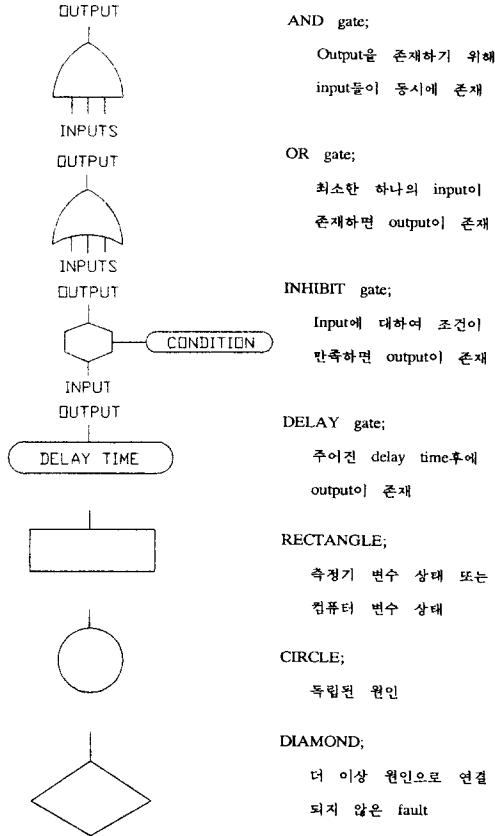


그림 1. 논리 부호와 사건 부호

### (2) 증상 트리의 작성

그림 2와 같은 공정을 예를 들어 증상 트리 작성 단계와 방법을 설명해 보자. 이 공정은  $F_1$ ,  $L_1$ ,  $F_2$ 를 측정하여  $F_1$ 을 조절함으로서  $F_1$ ,  $L_1$ ,  $F_2$ 를 제어하기 위한 cascade 루프가 포함된 공정이다. 여기에서  $L_1$ 과  $F_2$ 를 측정하여  $F_1$  제어를 위한 setpoint를 결정하고  $F_1$  제어기에 내려짐으로서  $F_1$  제어기는 direct 제어와 cascade 제어를 동시에 수행하게 된다. 이 공정의 영역은 펌프에서 유출 유량 측정기까지로 제한하고 top event  $L_1$  감소에 대해 증상 트리를 작성해 보자.

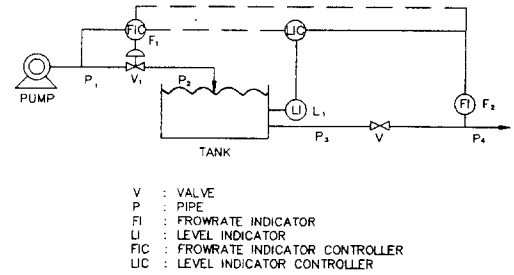


그림 2. 제어 루프를 포함한 액위 조

액위 감소에 영향을 미치는 원인들은  $F_1$ 의 감소와  $F_2$ 의 증가와  $L_1$  측정기 실패(-), 조 누출등이 있다. 그러므로 트리는 다음과 같이 된다.

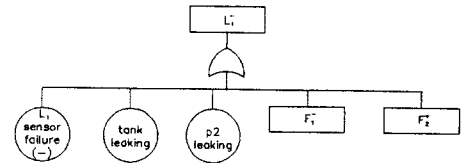


그림 3. 액위 감소에 대한 증상 sub 트리

여기에서  $F_1^-$ 와  $F_2^+$ 는 sub 증상으로 표현되어 있으므로 이것에 대한 트리가 필요하게 된다. 그러므로  $F_1^-$ ,  $F_2^+$ 에 대해서 트리를 작성해 보자.

$F_1^-$ 에 영향을 주는 원인들을  $L_1$ 의 증가와  $F_2$ 의 감소, 펌프 고장,  $F_1$  측정기 실패(-),  $P_1$  누출 등이 있다.  $F_2^+$ 에 영향을 주는 원인은  $F_2$  측정기 실패(+),  $V_2$  열림,  $L_1$  증가가 있다. 그러므로  $F_1^-$ 와  $F_2^+$ 에 대한 트리는 다음과 같다.

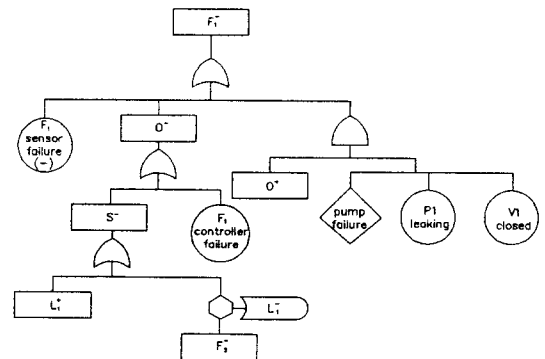


그림 4. 유입 유량 감소에 대한 증상 트리

그림 3, 그림 4, 그림 5를 연결하여  $L_1$  감소에 대한 트리를 완성해 보자. 우선  $F_1$  감소 트리를  $L_1$  감소에 연결할때 setpoint(S) 감소 밑에  $L_1$  증가는 top event  $L_1$  감소에 위배되므로 sub 사상에서 제외 시킨다. 그리고  $F_2$  감소도  $L_1$ 이 감소하지 않은 조건 gate가 있으므로 top

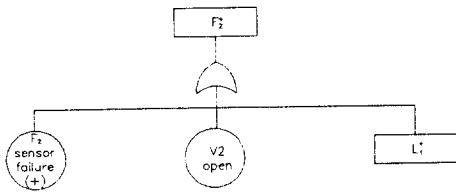


그림 5. 유출 유량 증가에 대한 증상 트리

event에 위배된다. 그러나 여기에서 고려되어야 할 것은  $F_1$ 의 감소는 측정기 정보로서의 감소와 실제량 감소가 있다.  $L_1$  감소에 영향을 주는 것은 실제량 감소이므로  $F_1$  측정기 실패(-)는 제외되며  $F_1$  측정기 실패(+)가 제어기 output(O)의 감소 아래 사상으로 연결되어야 할 것이다.  $F_2$  증가에 대해서도  $F_2$  측정기 실패(+)와  $L_1$  증가가 제외되고  $F_2$  측정기 실패(-)가  $F_1$  감소의 sub 사상인 setpoint 감소에 연결되어야 한다. 위에서 언급한 모든것을 고려하여 트리를 작성하면 다음과 같이  $L_1$  감소에 대한 트리가 완성된다.

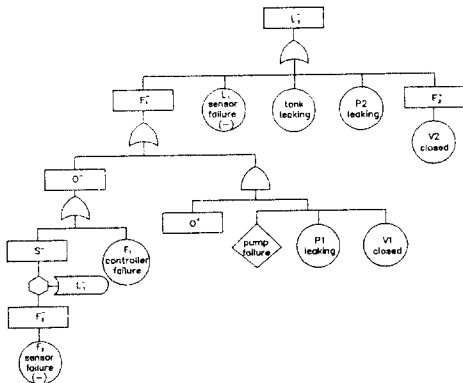


그림 6. 액위 감소에 대한 증상 트리

### (3) 나프타 분해로의 증상 트리

#### (가) 나프타 분해로(Naphtha Furnace)

나프타 분해로 공정은 물질 흐름상 크게 세 부분, 나프타와 회석 스팀(dilution steam)이 섞여 고온에서 열 분해되어 에틸렌등의 생성물을 만드는 원료-생성물 흐름 부분, 연료 가스(fuel gas)를 연소시켜 나프타 분해의 열 원으로 공급되는 부분과 에틸렌등의 생성물을 급냉하기 위해서 BFW(Boiling Feed Water)를 공급하는 부분이다. 물질을 공급하는 부분들은 각각의 흐름을 제어하기 위한 DCS(Distributed Control System)이 부착되어 있고 제어 변수와 중요 변수들의 관계를 모델링하여 각 제어기의 setpoint를 내려주는 DMC(Dynamic Matrix Control) logic이 연결되어 있다. 장치상으로 분해로 공정을 나누면 분해로, 나프타 펌프, ID FAN, BFW 펌프, 스팀 드럼(steam

drum), 제어기등으로 구성될 수 있으며, 분해로 자체는 여러개의 열교환기와 같은 역할을 하며 생성물의 열원을 제거하기 위한 것은 BFW 공급과 스팀 드럼 부분이 담당한다. 나프타 분해로 공정 자체가 여러개의 열교환기로 모델링될 수 있고 제어 구조도 DCS와 DMC logic이 같이 부착되어 있기 때문에 변수들의 관계가 어떤 다른 공정정보보다 복잡하게 된다.

#### (나) 나프타 분해로의 증상 트리 작성

나프타 분해로에 대한 SIM은 측정기(sensor)로 부터 컴퓨터로 오는 정보를 이용하여 표현되었으며 대부분 OR gate로 표시되어 하나의 원인이 top event로 선정된 공정 변수에 영향을 미치는 성질(One-Event Minimal Cut Sets)을 가지며 증상 부 트리 모델(symptom sub-tree model)로 구성되어 있다. 나프타 분해로에서 컴퓨터로 입력되는 공정 변수는 75개 인데 증상 트리로 표현된 변수는 제어기 상에서 계산되는 변수 38개를 합친 113개이다. 그러므로 113개의 변수에 대해 높고 낮은 두 상태에 대해 두개의 트리를 가지므로 총 226개의 트리로서 분해로의 모든 변수와 원인들의 관계를 표시하였다.

나프타 분해로의 증상 트리는 나프타, 회석 스팀, BFW, 연료 가스가 공급되기전 공정을 제외시켜으며, top event로는 공정 변수와 제어기상의 변수로 제한하였으며, 나프타 분해로에 대하여 작성한 SDG를 기초로하여 변수 상호간에 인과 관계를 규명하고 조업자들과의 많은 대화를 통해 발생 가능한 원인들을 고려하여 작성되었다.

#### (4) 증상 트리에 포함된 가정

증상 트리를 작성함에 있어서 트리 자체가 공정 전체에 대한 모든 상황을 고려할 수 없으므로 많은 가정들이 사용되었다. 우선 트리를 구성하기 전에 큰 두 가정을 하였다. 첫째로 진단할 원인들이 top event로 정의한 공정 변수 변화에 대해서 주어진 트리 상에 존재한다라는 것이고, 둘째로 단지 하나의 원인이 공정변수의 변화를 가져 오게 한다라는 가정(Single Fault Cause Assumption)이다. 트리를 작성하는 과정에서 제어 루프가 있는 변수에서 action이 되돌아 오는 것을 표시하지 않고 어떤 상태 인지 트리위에서 결정되면 트리미의 사상에서 나타나는 모순된 변수 상태는 제외시켰으로써 제어 루프를 끊었다. 한 변수에 대한 sub 사상에서 대부분 많은 것이 고려되어야 하나 영향이 적은 원인들은 제외시켰다. 그리고 독립된 원인으로 선택된 것들은 세부적인 원인으로 발전시키지 않고 공정 영역 외부에서 발생되어 영역에 들어오는 원인들은 DIAMOND 사건 부호로 나타냄으로써 영역 외부로 트리가 확장되는 것을 막았다.

### 3. 전문가 시스템의 구현

이상 진단 전문가 시스템은 Intelicorp사의 KEE (Knowledge Engineering Environment)로 SUN-3/260에서

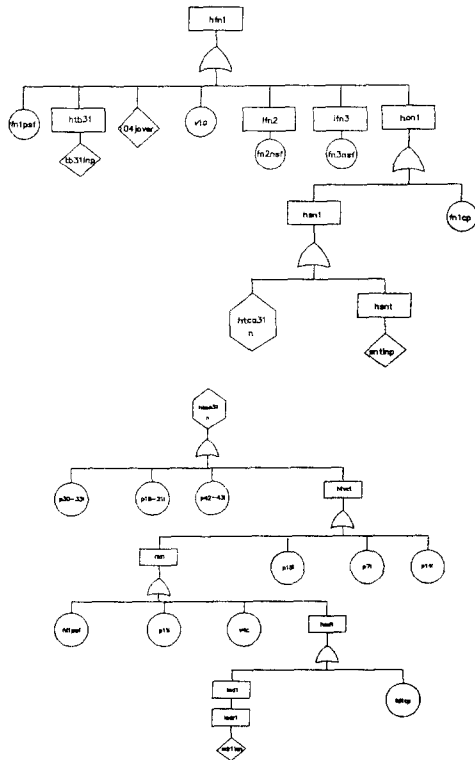


그림 7. 나프타 입력 유량 증가에 대한 중상 트리

구현되었다. 전체적인 시스템 구조는 Frame 형식으로 구성되었으며 STM을 이용하여 지식 모델을 표현하였다.

### (1) 지식 표현

지식 표현 방법은 크게 3가지로 나누어 표현되었는데, 시스템 전체 구조를 나타내고 나프타 분해로상의 변수들의 상관 관계를 표현할 수 있게 하고 inheritance와 OOP (Object Oriented Programming) 기법을 활용할 수 있는 Frame 형식, STM을 LISP 언어로 표현하여 증상과 원인들의 관계를 표현하는 방법과 Frame 상에서 증상과 원인들의 관계를 mapping 하는 방법이다.

Frame 형식의 표현 방법은 나프타 분해로 공정을 서로 관계가 있는 부분으로 나누고 여기에 STM에 top event로 표시된 변수를 연결하여 전체적으로 계층 구조를 갖게 하였다. Frame은 여러개의 unit들이 계층적 구조를 이루고 있고, 각 unit들은 여러개의 slot들로 구성되어 있다. 그림 8은 나프타 분해로 Frame의 구조를 보여주고 있다.

STM을 LISP으로 표현하는 방법은 top event로 선정된 변수에 영향을 미치는 원인들을 열거하였다. 이 과정에서 STM을 OR gate로만 논리 표현하고 여기에 대응하는 지식 표현 방법으로 모든 부 증상(sub-symptom)들을

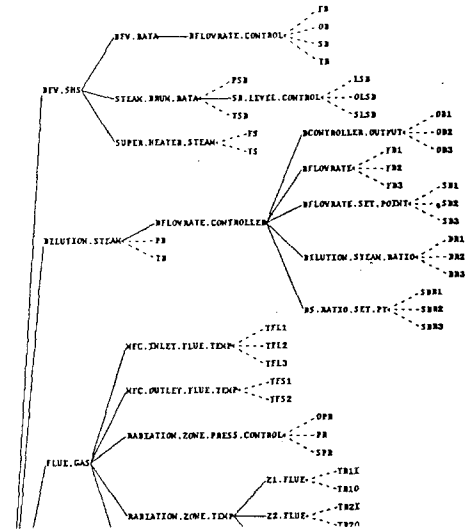


그림 8. 나프타 분해로 공정의 Frame 구조

제거하고 top event에 원인 후보를 직접적으로 연결된 것부터 부 증상에 연결된 원인 후보를 차례로 열거하고 순서적 의미를 부여하기 위해 부 증상에 연결되는 순서를 괄호로서 분리함으로써 부 증상을 제거하는 것을 보안하고 있다. 그림 9는 그림 7의 증상 트리를 LISP으로 표현한 예를 보여주고 있다.

```
(setq sympmap (make-hash-table))

(setf (gethash 'hin sympmap) '((tnpsf niuh))
      (gethash 'lin sympmap) '((tnnsf niul))
      (gethash 'hfnl sympmap) '(((fn1psf 104jover v1o)
                                     (fn1cp fn2nsf fn3nsf)
                                     (sntinp tb3linp p18l
                                     p19l p20l p21l p30l
                                     p31l p32l p33l p42l
                                     p43l) (p7l p14l p15l)
                                     (fdlpsf p11l v4c)
                                     (fd1cp) (sdr1inn)))
```

그림 9. 증상 트리의 지식 표현

Frame 상의 공정 변수 unit과 진단할 원인들을 mapping하는 방법은 단순히 원인 후보가 어떤 unit에 속해 있으며 어떤 변수와 직접적으로 영향이 있는지를 표시하는 지식으로 common LISP에서 제공하는 make-hash-table function에 의해 작성되었다. 그림 10은 그 예를 보여주고 있다.

```

(setq fimap (make-hash-table))
(setf (gethash 'tnpsf fimap) '(tn high.sensor.failure)
      (gethash 'tnnsf fimap) '(tn low.sensor.failure)
      (gethash 'fn1psf fimap) '(fn1 high.sensor.failure)
      (gethash 'fn1nsf fimap) '(fn1 low.sensor.failure)
      (gethash 'fn2psf fimap) '(fn2 high.sensor.failure)
      (gethash 'fn2nsf fimap) '(fn2 low.sensor.failure)
      (gethash 'fn3psf fimap) '(fn3 high.sensor.failure)
      (gethash 'fn3nsf fimap) '(fn3 low.sensor.failure)
      (gethash 'tdpsf fimap) '(td high.sensor.failure))

```

그림 10. 원인-증상 관계 표현

## (2) 시스템의 작동

이상 진단 전문가 시스템은 공정 변수의 과거 값들을 평균한 평균치를 가지고 산출된 alarm band를 넘어가는 공정 변수가 시스템으로 들어오게 되면 SYMPTOM 지식 베이스에 각 unit의 status slot에 message가 보내지며, OOP 기법으로 alarm band를 벗어난 공정 변수들의 list가 만들어지고 이것으로 LISP로 표현된 STM 모델 지식 베이스에 있는 각 증상에 대한 원인들의 list를 가져오게 되며, 이 원인 후보를 intersection하여 원인 후보를 제시한다. 이와 더불어 ActiveImage로된 증상 panel에 증상들이 표시되며 intersection한 원인 후보에 대해서 원인-증상 관계 표현 지식 베이스에서 얻어진 관계를 가지고 각 원인들에 대해서 fault panel에 표시하게 된다.

## (3) 원인 후보의 진단

원인 후보가 fault panel에 나타나지면 각 원인들에 대해서 타당성을 검토하는 fault 모사(fault simulation)가 시작되는데 그 단계는 다음과 같다.

우선 first-symptom-generator에서 각 원인 후보에 대한 꼭 나타나야하는 증상을 산출하고 나타난 증상이 있는지를 판단한다. 그리고 각 원인 후보에 대해서 symptom-list-generator에서 나타날 증상들의 list를 만들어서 현재 나타난 증상 형태에 위배되는지를 검토한다. 이 모든 진단이 끝나면 각 원인 후보에 대해서 상태를 결정하여 각 slot에 SURE(assured), PRO(proposed), FEA(feasible), CON(conflict)의 값을 간격하여 fault panel에 표시된다.

## (4) 시스템 진단 전략

원인 후보 진단 전략은 앞 절에서 설명한것과 일치하며 시스템 전체적 진단 전략을 살펴보면 다음 흐름도와 같이 표현할 수 있다.

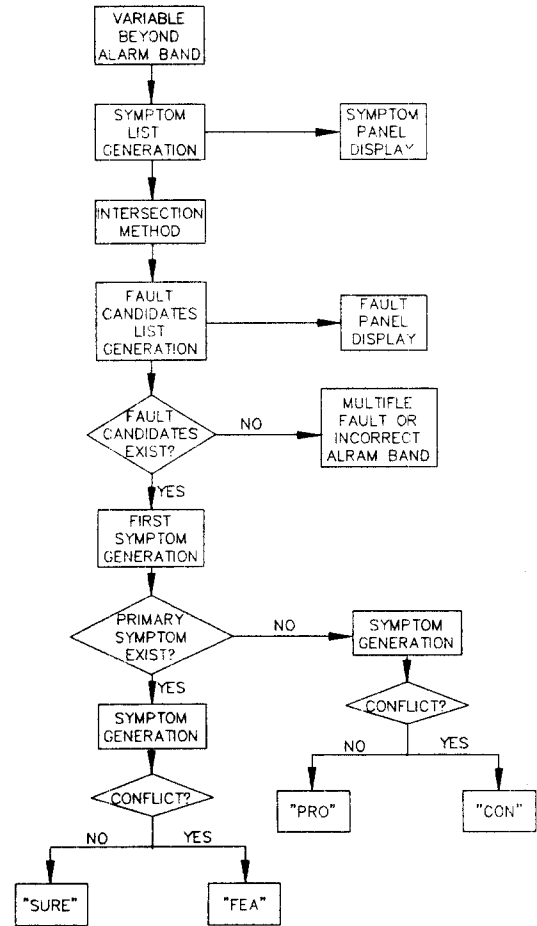


그림 11. 시스템 진단 전략 흐름도

## (5) User Interface

본 연구에서 개발된 이상 진단 전문가 시스템의 user interface는 symptom panel, fault panel과 공정도가 있다. alarm band를 넘어서는 공정 변수가 나타나게 되면 자동적으로 변수의 상태를 나타내는 image가 symptom panel에 표시되고 intersection 방법으로 구한 원인 후보가 fault panel에 표시된다. 그리고 각 원인 후보에 대해 타당성 검토가 행해져 원인 후보의 상태가 결정되면 이것이 image로 fault panel에 나타난다. 또 사용자가 공정도를 원하면 고장 부위를 표시하는 공정도가 나타나게 된다.

## 4. 결론 및 제언

본 연구에서는 STM을 이용하여 지식 베이스를 구현하고 이로 부터 이상 진단 전문가 시스템을 개발하였다. STM을 지식 베이스로 표현하는 방법은 list 구조로 간단히 표현되었으며 제어 구조도 intersection 방법과 증상 산출(symptom generation) 방법에 의하여 행해졌다. 이와

같은 방법은 간단하며 수행 속도가 빨라 실시간 이상 진단 전문가 시스템의 지식 모델로 적합함을 확인하였고, 나프타 분해로에 대한 사고 사례와 가상 사고 사례에 적용하여 검토한 결과, 신속한 원인 후보 제시와 원인 후보의 타당성 검토에 적합한 모델이었다.

또한 STM을 작성하고 표현하는데 INHIBIT gate와 DELAY gate를 포함하여 공정 모델로 사용한다면 더 많은 정보를 포함하게 되어 더욱 빠른 진단을 요하는 전문가 시스템의 지식 모델로 적합할 것이다.

#### 감 사

본 연구의 수행을 위하여 연구비를 지원하여준 (주)유공에 감사드리며, 공정에 대한 지식과 경험을 제공해 준 NCC(Naphtha Cracking Center) hot section을 담당하는 여러분들과 공정에 대한 자료와 제어 구조에 대한 정보를 제공해 준 제어부 여러분들께 감사드립니다.

#### 참고 문헌

1. D. M. Himmelblau, Fault Detection and Diagnosis in Chemical and Petrochemical Processes, Elsevier Scientific Publishing, 1978
2. E. S. Yoon, L. B. Evans and T. B. Tooney, "Application of Fault Tree Analysis to Process Failure Diagnosis", Ph.D. Thesis, Department of Chemical Engineering, Massachusetts Institute of Technology, 1980
3. G. J. Powers and F. C. Tomkins, "Fault Tree Synthesis for Chemical Process", AIChE J., Vol. 20, No.2, 1974
4. J. C. Hoskins and D.M. Himmelblau, "Artificial Neural Network Models of Knowledge Representation in Chemical Engineering", Comput. chem. Engng., Vol.12, No.9, 1988
5. J. H. Han and E. S. Yoon, "A Study on the Knowledge-Based Expert System for Cement Process Control and Diagnosis", M. S. Thesis, Department of Chemical Engineering S.N.U., 1986
6. M. A. Kramer and B. L. Palowitch, "A Rule-Based Approach to Fault Diagnosis Using the Signed Directed Graph", AIChE J., Vol. 33, No. 7, 1987
7. N. J. Nilson, Principles of Artificial Intelligence, Tioga Publishing Co., 1980
8. S. H. Rich and V. Venkatasubramanian, "Model-Based Reasoning in Diagnostic Expert Systems for chemical Process Plants", Comput. Chem. Engng., Vol. 11, No. 2, 1987

9. S. K. Shum, J. F. Davis, W. F. Punch and B. Chandrasekaran, "An Expert System Approach to Malfunction Diagnosis in Chemical Plants", Comput. Chem. Engng., Vol. 12, No. 1, 1988