

# 논리도면과 경보절차서를 이용한 경보원인추적 시스템

박중팔\*, 김은주\*, 류승필\*

\*세명대학교 컴퓨터학과

e-mail : orion08@ailab.semyung.ac.kr

## An Alarm-Cause Tracking System Using Logic Diagram and Alarm Response Procedure

Joong-Pal Park\*, Eun-Ju Kim\*, Sung-Pil Lyu\*

\*Dept of Computer Science, Semyung University

### 요 약

원자력 발전소 운전시 이상 현상으로 인한 발전소의 정지 등 기타 여러 사고시 그 원인을 신속히 파악하여 조치하는 일은 발전소의 안전과 효율적인 측면에서 매우 중요한 부분이다.

경보원인추적시스템(ACTS:Alarm-Cause Tracking System)은 발전소 현장에서 자주 사용하는 경보 논리도면을 자동화하여, 경보의 원인을 보다 상세하게 추적하여 경보절차서에 표시되는 원인들 중 실제로 발생한 원인과 그에 해당하는 절차 부분을 강조하여 표시함으로써 운전원이 쉽게 상세원인을 파악할 수 있도록 한다. 또한 이 시스템은 운전원이 경보의 발생경로와 논리상태를 그래픽으로 전산화된 논리도면상에서 확인하여 경보에 대한 빠른 조치를 취할 수 있도록 인터페이스를 제공한다.

### 1. 서론

최근 컴퓨터 산업과 전자 산업의 발달로 각 산업의 공정처리에 있어서 자동화는 매우 빠른 속도로 진행되고 있다. 이에 따라 원자력발전소의 공정제어도 점차 자동화되고, 자동화가 진행됨에 따라 기계의 수행범위가 점차 넓어지면서, 사람의 개입은 점차 줄어들고 있다. 그러나 시스템의 이상이 발생하면 사람의 개입이 필요한 때가 있는데, 이때 자동화로 인한 시스템 내부의 복잡성 때문에 자동화 이전보다도 해야 할 작업이 더욱 어려워질 수 있다. 특히 원자력발전소의 경우, 다른 산업분야에 비해 비교적 논리가 복잡한 편이므로 자동화가 진행 될 수록 시스템이상 발생 시, 그 원인을 사람이 직접 추적해나가는 것이 점점 더 어려워진다.

그래서 이상상태가 발생하면, 경보가 발생하고 주요 경보에 대해서는 그 원인을 알려주는 운전원 보조 시스템의 개발이 연구되고 있지만, 그 원인이 구체적이지 못한 경우 자세한 원인을 파악하기 위해 운전원

은 경보절차서, 경보논리도면 및 제어반의 기기상태를 참조하여 경보의 원인을 직접 추적하게 된다. 그러나, 발전소에 사용되는 경보논리도면은 수백 페이지에 이르고, 해당논리를 찾아서 제어반으로부터 해당 신호를 일일이 확인하면서 도면상의 경보논리를 신속, 정확하게 따라간다는 것은 운전원에게 있어서 매우 부담되는 일이며, 자칫 오류를 범할 가능성도 있다.

또한, 고장의 원인은 하나이나, 하나의 고장이 원인으로 인해 매우 많은 다중 경보들이 발생하게 되는 것이 일반적이다. 이럴 경우 다중 경보 중에서 원인이 되는 경보를 찾는 작업은 매우 중요한 부분이 된다.

이 논문에서는 경보와 원인간의 관계를 발전소에서 이미 사용하고 있어 논리적인 측면에서 검증이 필요하지 않는 논리도면(제어논리 및 경보논리)과 경보절차서에 있는 정보를 이용하여 경보원인을 추적하는 방법을 제시한다. 제안된 방법은 논리도면과 경보절차서를 객체지향적인 그래픽 형태로 전산화함으로써, 논리구축과 확인을 쉽게 하고, 구축된 논리도면 내에 있

는 논리소자들의 입출력관계를 이용하여 논리의 상태의 원인을 추적할 수 있도록 한다. 실제 발전소 경보창과 같이 전산화된 논리를 통하여 경보가 발생하며, 경보는 경보절차서와 논리도면과 연계되어 그 원인 추적이 이루어진다. 한편, 도면내의 논리소자는 객체화되어 있어서 최근 상태 변경시각에 대한 정보를 가지므로 원인 추적시 경보원인의 논리적, 시간적인 배열 및 표시가 가능하다. 그리고 논리추론의 과정을 추적경로로 표시하고, 경보절차서에 해당되는 경보원인과 해당조치를 강조하여 표시함으로써 운전원의 경보에 대한 조치에 도움을 주는 방안을 제시하였다.

2. 시스템 구성

본 시스템은 논리도면과 경보절차서를 이용하여 경보원인추적시스템을 구성하였다. 논리도면과 경보절차서는 이미 검증이 되어있는 자료이기 때문에, 이것을 이용하여 시스템을 구성하면 검증하는 작업 중 상당부분을 줄이는 이점을 얻을 수 있다.

경보원인추적시스템은 그림 1처럼 크게 세 부분의 주요 모듈과 여러 DB로 구성이 되어 있다.

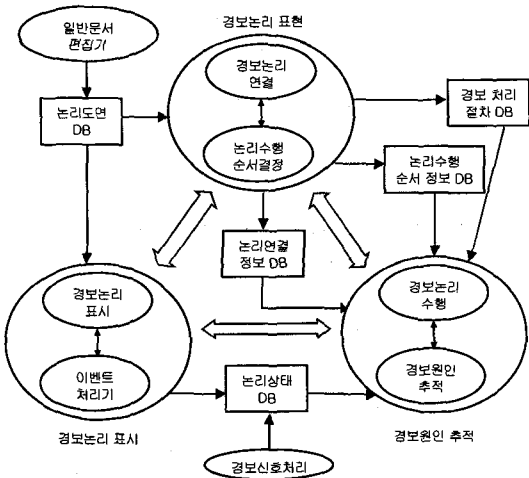


그림 1. 시스템 전체 구성도

시스템의 기본적인 운영은 일반 문서 편집기를 이용하여 논리도면을 작성하고, 작성된 논리도면을 DB에 추가하여 논리도면 DB가 업데이트 된다. 경보논리 표현 모듈은 구축된 논리도면 DB를 이용하여 논리도면내에 있는 각각의 구성요소들의 속성, 위치, 상태, 다른 구성요소들과의 관계 등 여러 가지 정보를 가지는 객체를 생성하게 된다. 이렇게 생성된 객체를 기반으로 경보논리연결 모듈은 논리 구성요소들 간의 연결

관계 정보를 DB화하여 논리연결 정보 DB를 생성하게 되고, 연결된 논리를 바탕으로 논리수행 순서결정 모듈에서 논리 수행 순서를 결정하고, 이를 DB화하여 논리수행 순서 정보 DB를 구축하게 된다. 논리 수행 순서는 기본적으로 위치적으로 상위/좌측 우선으로 순서를 정하게 된다.

논리도면 DB를 이용하여 경보논리 DB 구축을함과 동시에 경보논리 표시 모듈에서는 CRT에 논리도면을 표시하게 된다. 이때 경보논리의 상태를 DB화시켜 논리상태 DB를 구성한다. 또한 이벤트 처리기에 의해서 논리도면의 이동, 연결정보 확인, 입력 등 여러 가지 이벤트를 수행하게 된다.

그림 2는 일반 문서 편집기로 작성된 논리도면의 일부이며, 아랫부분은 작성된 논리도면을 경보논리 DB 구축 모듈과 경보논리 표시 모듈을 이용하여 CRT에 나타난 화면이다.

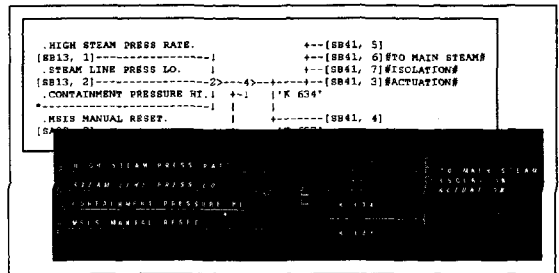


그림 2. 작성된 논리도면과 표시의 예

경보원인 추적 모듈은 논리상태 DB가 경보신호처리 모듈에 의해 변경될 때에 구축된 각종 DB를 바탕으로 경보 발생 유무, 경보논리 상태를 파악한다. 경보논리는 각각의 논리요소(AND, OR, NOT, 블록 등)를 node로 하는 방향성그래프로 표시할 수 있다. 그림 3은 경보논리를 방향성그래프 형태로 표시한 것으로 노드 N은 입력  $I_1, I_2, \dots, I_k (k=1,2,3,\dots)$ 로부터 신호를 받아 논리를 수행하고 자신의 논리상태를 갱신하게 된다. 각각 노드의 상태를  $S(N), S(I_1), S(I_2), \dots, S(I_k)$ 라 두면,  $S(N)$ 은 입력상태에 의해서 결정되는데,  $S(N)$ 을 유지하는데 필요한 입력상태를 찾아서 그것을 출력상태의 원인으로 간주한다. 예를 들어, N이 OR 논리요소이고,  $k=3, S(N)=1, S(I_1)=0, S(I_2)=1, S(I_3)=1$  이면,  $S(I_2)$ 와  $S(I_3)$ 는  $S(N)=1$ 이 되기 위한 필요조건으로써,  $S(N)$ 의 원인으로 간주한다. 원인추적은  $S(I_2), S(I_3)$ 의 입력에 대해서도 계속적으로 수행되며, 더 이상 진행할 그래프의 노드가 존재하지 않을 때 원인 추적을 마치게 된다. 최종 상세 원인은 그래프상의 말단노드

(Terminal)가 되며, 말단노드는 그림 3의 입력신호에 해당된다.

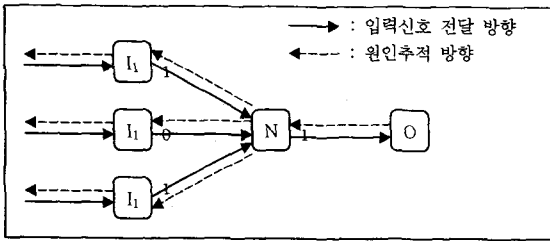


그림 3. 논리상태 원인추적의 예

경보원인을 추적하기 위해서 앞서 기술한 바와 같이 2차원 그래프 상에서 각 노드의 상태를 확인하고, 그 상태의 원인이 되는 입력을 발생시키는 노드에 대해서 재귀적(recursive)으로 원인을 추적한다. 그러나, 논리는 2차원그래프 형태로 같은 노드를 순환적으로 반복할 수 있기 때문에 이에 대한 고려가 필요하다. 따라서 같은 한번 지나간 노드의 방문을 막기 위해 지나간 노드에 대한 추적경로리스트를 작성하고, 리스트에 있는 노드는 다시 방문하지 않도록 하였다. 그림 4는 이에 대한 알고리즘이다.

```

function TrackingCauses(논리요소)
{
    (1) 주어진 논리요소의 상태를 결정하는 데에 필요한 입력신호를 주는 논리요소 찾기
    (2) 만약 해당하는 입력 논리요소가 없으면 return;
    (3) 해당하는 모든 입력 논리요소들에 대해 다음을 반복
    {
        (3-1) 해당 입력 논리요소가 말단노드인지 검사
        (3-1-1) 말단노드이면 경보원인리스트에 등록 후 다음 입력 논리요소에 대해 (3)을 계속
        (3-2) 해당 입력 논리요소가 추적경로리스트에 있는지 검사
        {
            (3-2-1) 있으면 다음 입력 논리요소에 대해 (3)을 계속
            (3-2-2) 없으면 TrackingCauses(해당입력논리요소) 수행하고 해당 입력논리요소를 추적경로리스트에 등록
        }
    }
}
    
```

그림 4. 경보원인추적 알고리즘

그림 4의 알고리즘 (3-1-1)의 수행결과로 경보원인리스트가 생성되며, 이것이 경보의 세부원인들의 목록이 된다. 그리고 (3-2-2)의 수행결과로 추적경로리스트가 생성되는데, 이것은 경보원인을 추적하는 경로 상에 존재하는 논리요소들에 대한 정보를 가지며, 경보논리표시모듈이 추적경로를 화면에 나타내기 위해서 사용된다. 추적된 경보원인을 표시할 때에 해당되

는 경보절차서, 경보원인의 추적 경로 및 각종 정보를 표시한 추적트리도 함께 표시된다.

경보원인 추적에 있어서 경보논리 각각마다 자신의 상태, 변경된 상태, 변경된 시간 등의 정보를 가지고 있기 때문에 이 정보들을 이용하여 경보논리간의 선행경보(precedent alarm)와 후행경보(following alarm)를 가릴 수 있다.

### 3. 실험 및 고찰

본 시스템의 실험을 위해 고리 원자력발전소 3&4 호기 2차 시스템의 경보 중 101개 경보 및 관련 경보 절차서를 IBM PC(Pentium 733Mhz)에 구현하여, 총 250페이지의 논리도면과 연결하였으며, 고리 연수원 시뮬레이터와 연계하여 기능테스트도 실시하였다.

실험은 기능 실험과 속도 실험으로 나누어 진행하였다. 기능 실험의 내용은 계통 내의 선행 경보 추적 및 표시, 경보원인추적 경로를 논리도면에 표시, 원인 관련 경보절차서 표시, 경보절차서 내의 해당 항목 강조 표시 등을 실시하였다.

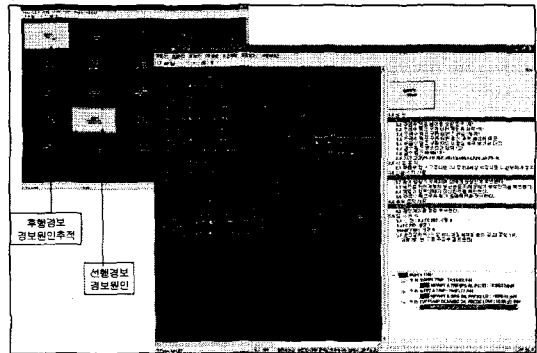


그림 5. 기능 실험 결과 화면

표 1. 속도실험 결과

실험 내용	결과 속도
논리소자 상태 갱신	$1.8 \times 10^{-1}$ sec/논리소자
경보원인추적	$6.1 \times 10^{-2}$ sec/논리소자
추적결과표시	$8.0 \times 10^{-2}$ sec

표 1은 속도 실험에 대한 결과이다. 본 속도 실험을 바탕으로 발전소 전체 Full Scope System에 적용할 경우, 논리도면 수가 2000 페이지 내이고, 논리소자의 수도 페이지 당 평균 최대 87개 이하이므로 논리소자 수는  $2,000 \text{ pages} \times 87 \text{ logic element/page}$

임으로 최대 174,000개 이하가 된다. 최소 논리소자 개수를 가정하면 예상 속도는 표 2와 같이 나타나며, 경보원인추적 및 표시를 하는데 걸리는 시간은 0.17 sec 이내 라는 것을 알 수 있다.

표 2. Full Scope System 예상 속도

실험 내용	결과 속도
논리소자 상태 갱신	$3.1 \times 10^{-2}$ sec/논리소자
경보원인추적	$6.1 \times 10^{-2}$ sec/논리소자
추적결과표시	$8.0 \times 10^{-2}$ sec

#### 4. 결론

경보원인추적시스템은 크게 두 가지의 기능을 가지고 있다. 첫째는 경보원인을 추적하여 운전원에게 경보의 상세한 원인정보를 제공하는 기능이고, 또 하나는 논리도면을 전산화할 수 있는 기능이다.

첫 번째 기능은 운전원 보조시스템의 주요 기능으로 경보 발생 시 원인 추적, 해당 절차서 표시 및 해당 항목 표시 기능을 수행하는 것이다. 또한 다중 경보 발생 시 논리적, 시간적 순차를 이용하여 선행 경보를 표시하는 방법으로 원인 경보의 축약을 할 수 있다. 현재 고리 원자력 교육원 시뮬레이터에 적용하여 시험이 성공적으로 끝났으며, 한국표준원전을 대상으로 경보원인추적시스템을 구현하여 그 기능과 성능을 재확인할 예정이다.

두 번째 기능은 논리도면 전산화 기능으로 현재 문서화되어 있는 논리도면을 전산화하여 경보논리도면을 컴퓨터로 검색하거나 논리를 추적할 수 있는 기능으로, 논리도면의 보관, 관리 등이 매우 용이할 뿐만 아니라, 입력신호만 주어지면 발전소 컴퓨터와 독립적으로 논리의 상태도 바로 확인할 수도 있다. 이러한 기능은 가동중인 원자력발전소 운용 측면뿐만 아니라, 설계중인 원자력발전소의 경보논리를 확인하거나 수정하는 데에도 도움을 줄 것으로 생각한다.

#### 참고문헌

- [1] Nuplex 80+ Advanced Control Complex, ABB CE, 1995.  
 [2] I.S. Kim, Computerized Systems for On-line Management of Failures: A State-of-Art Discussion of Alarm Systems and Diagnostic Systems Applied in the Nuclear Industry, Reliability Engineering and System Safety 44, pp279-295, 1974.

[3] J.H. Park and P.H. Seong, An integrated knowledge base development tool for knowledge acquisition and verification for NPP dynamic alarm processing systems, Annals of Nuclear Energy, Vol. 29, pp447-463, 2002.

[4] K.C. Kwon, et.al, The Real-Time Functional Test Facility for Advanced Instrumentation and Control in Nuclear Power Plants. IEEE Transaction on Nuclear Science, Vol.46, No.2, 1999.

[5] R.P. Leger, W.J. Garland and W.F. Poehlman, Fault detection and diagnosis using statistical control charts and artificial neural networks, Artificial Intelligence in Engineering, Volume 12, Issues 1-2, pp35-47, 1998

[6] J. T. Kim, et al, An Analysis of the Causal Alarm in Alarm and Diagnosis-Integrated Operator Support System(ADIOS), MARCON97, Knoxville, Tennessee, USA, 1997.

[7] S.W. Cheon, et. al, Development strategies of an expert system for multiple alarm processing and diagnosis in nuclear power plants, IEEE Trans. on Nuclear Science, Vol.40, Issue 1, pp21 -30, 1993.