

2002 추계학술발표회 논문집

한국원자력학회

논리도면추적에 의한 경보원인 추적 Tracking Alarm Causes by Logic Diagram

김정택*, 이명수**, 홍진혁**, 류승필**⁸, 박중필***, 김은주***

*한국원자력연구소

**한국전력연구원

***세명대학교

요 약

원자력발전소에서 경보가 발생하면 그 영향을 줄이기 위해서 운전원은 신속하게 원인을 파악하고 그에 맞는 조치를 취해야한다. 경보원인 추적을 위한 많은 연구가 있었으나, 이들 연구들은 대부분 지식이나, 인과관계 정보를 이용하였다. 이러한 지식과 정보들을 구현하고 검증하는 데에는 전문가들의 많은 노력이 소요되며, 특히 인간의 경험으로부터 생성된 지식이나 정보는 충분한 검증을 거치지 않으면 잘못된 결과를 유도할 수 있다. 이 논문에서는 이러한 불확실한 지식이나 정보를 이용하지 않고 현장에서 사용되는 검증된 논리도면과 경보 relay 등과 같은 전기적 논리 등 정확한 논리적 정보만을 이용하여 경보논리를 추적하는 방법을 제안한다. 제안한 방법은 논리도면, 기계적 경보 relay logic 및 경보절차서를 객체화시키고 시각화시킴으로써 논리의 편집과 확인을 용이하게 하였다.

Abstract

An Operator should provide the correct and fast actions on a cause of alarms and failure for reducing the effect of failure. There are a lot of study. But most of those studies may use a physical knowledges or causal relationships. Most of those studies impose on high level information like the physical knowledges or causal relationships of failure rather than the logical states or process signals as the detail causes of failure. It is very difficult that the physical knowledges or causal relationships are to be implemented and verified. This paper proposes a methodology for tracking alarm causes by logic of alarms. This methodology uses the proper logical knowledges on the proven logic and alarm diagram or electrical alarm relay logic than the uncertain physical knowledges or causal relationships. This system is to display the highlighted alarm procedure related to the causes. The system can be used for operator to identify the detail causes of alarm without checking all candidates for causes in alarm response procedure and the logical states of alarms with alarm logic diagrams provided on CRT

이 연구는 과학기술부 원자력 계측제어계통 개발 과제인 “감시 및 운전지원 기술개발”의 지원을 받았음

dynamically.

1. 서 론

원자력발전소의 경우 이상상태가 되면 경보가 발생하는데 이들 경보는 원자력발전소의 각각의 시스템들이 서로 물리적으로 연결되어 있어, 최초 원인인 이상상태가 여러 기기 또는 시스템에 영향을 주고, 그 결과로 하나의 원인에 의한 이상상태가 많은 경보를 동시에 발생시키는 경우가 많다[1,4,7]. 동시에 발생하는 수많은 경보는 운전원이 직접적인 원인을 파악하기 어려울 뿐만 아니라, 그로 인해 경보에 대한 적절한 조치를 하기 어려워진다. 최근에는 이러한 문제를 해결하기 위해 동시에 발생하는 많은 경보를 축약하거나[6,7] 경보의 원인을 진단하여 운전원에게 알려주는 방법[1,8,9] 등이 많이 연구되고 있다. 경보원인을 찾기 위해 원인-결과의 관계 정보나, 시스템의 물리적 연관관계 등을 지식 또는 여러 가지 형태의 표현을 이용하여 원인을 추적하는 방법들을 제시하고 있다[8,9,10]. 그러나, 원인결과나 물리적인 연관관계를 정보화하는 것은 쉽지 않으며, 또한 정보를 구축하였다 하더라도 그 정보의 정확성과 신뢰성을 확보하는 데에도 많은 작업이 필요하다. 이에 따라, 이들 정보와 지식들의 신뢰성을 확보하고 이들을 구축하기 위한 작업을 줄이기 위한 연구도 활발히 진행되고 있다[2,8,9,12].

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 쉽게하고, 구축된 논리도면내에 있는 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 원인추적시 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. 경보원인 추적시스템 구성 및 기능

제안된 시스템은 경보논리처리, 경보원인추적, 사용자인터페이스 크게 3가지 기능을 경보논리 입력처리기, 논리소자, 경보원인추적기, 경보절차서, 사용자인터페이스 등 5개의 주요 클래스로부터 생성된 객체들에 의해 수행된다. 그림 1은 클래스들 간의 메시지의 흐름을 보여주고 있다. 그림 1에서 경보논리 입력처리기는 외부로부터 주어지는 공정신호를 감시하고 신호의 변화가 생기면 즉시 해당 논리소자 객체에게 메시지를 보내는 역할을 한다. 메시지를 받은 논리소자 객체는 입력상태를 연산하고, 그 결과 자신의 상태가 변하면 자신과 연결된 다른 객체에게 메시지를 전달하는 방식으로 논리가 수행된다.

한편, 경보원인추적은 경보발생시 사용자 인터페이스로부터 추적해야할 경보 정보를 받아 그 정보와 연결된 논리소자객체들에게 상태원인 추적메시지를 보낸다. 메시지를 받은 논리소자객체는 자신의 상태와 입력상태를 비교 조사하여 원인이 되는 입력 객체를 경보원인추적기에 보낸다. 경보원인추적기는 더 이상 추적할 객체가 없을 때까지 추적하고 그 결과, 경로추적트리는 사용자인

터페이스로, 정보와 원인들은 정보절차서로 보낸다. 그리고, 정보절차서는 추적된 정보원인(논리소자 객체)와 정보절차서를 연결하여 사용자 인터페이스에 보내어 화면에 표시한다. 여기서 정보원인(논리소자 객체)과 정보절차서간의 연결은 정보절차서내의 정보원인 후보를 나타내는 목록과 이에 해당하는 논리소자간의 논리적 연결로 사용자인터페이스를 통하여 이루어진다.

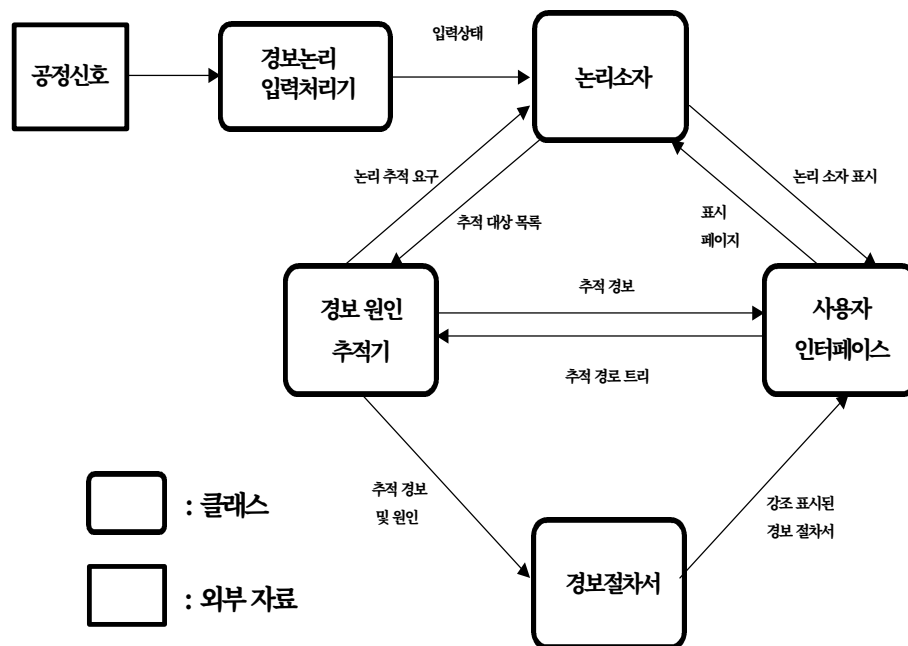


그림 1. 주요 클래스간 메시지 흐름

3. 정보논리의 객체화와 논리처리

3.1 논리소자의 객체화

정보 및 제어논리는 수백 페이지에 이르는 복잡한 논리도면들로 구성되어 있어서, 논리구현 및 논리확인 기능과 논리 추적 속도는 시스템의 성능을 결정하는 중요한 요소가 된다. 이 논문에서는 Lyu[]의 논리표현 방법을 이용하여 논리도면을 그래픽 객체화된 논리소자들로 표현함으로써, 논리도면을 도형적 편집과 도면내의 논리소자들의 동작상태를 시각적으로 확인할 수 있도록 하였다.

논리소자 클래스는 그림 2 와 같은 구조를 갖는다.

LogicElement(논리소자) 객체는 위치, ID, Description, 현재상태, 논리상태 변경시각, 입력객체포인터, 출력객체포인터 등의 자료멤버와 입력객체포인터 정보를 이용하여 입력상태읽기 함수와 상태가 변경되면 출력객체포인터의 정보에 따라 해당객체에게 메시지를 보내는 상태변경메시지보내기 함수를 가지며, 도면상에 논리적으로 연결되어 있지 않은 논리들간의 연결을 위하여 논리연결 함수는 대화상자를 제공한다. 예를들면, 논리절차서의 세부목록과 논리소자간의 연결, 정보창과 제어논리와의 연결 등이다. 그리고 각각의 객체는 자신의 현재 상태를 유지하도록하는 입력신호를 찾아서 해당 입력 객체의 포인터를 반환하는 함수를 가진다. 이함수는 정보원인추적을 위하여 사용된다.

한편, 하위클래스의 논리소자는 논리소자의 종류에 따라 입력상태를 조사하여 자신의 상태를 갱신

하는 상태연산 함수와 상태변화에 따라 자신을 다시 그리는 Draw함수를 가진다. 그림 3은 객체화

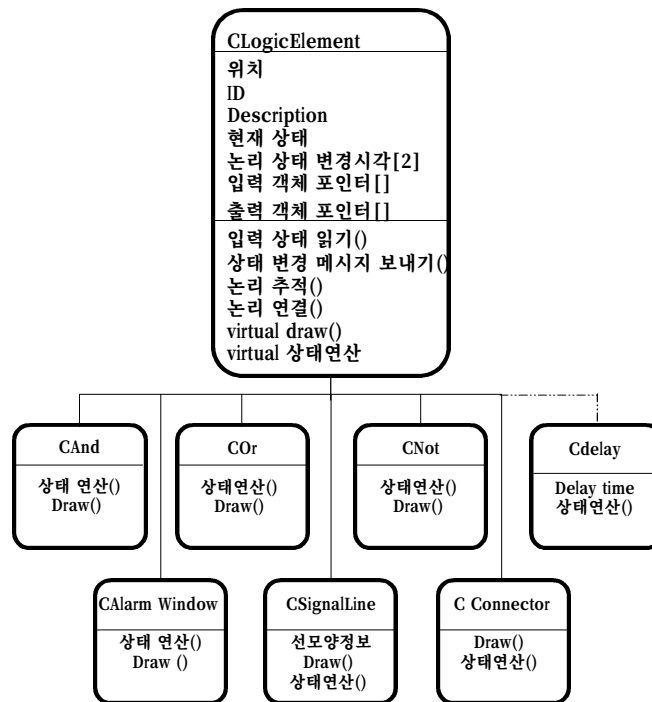


그림 2. 논리소자 클래스 상속 계층구조도

된 논리도면을 보여주고 있다.

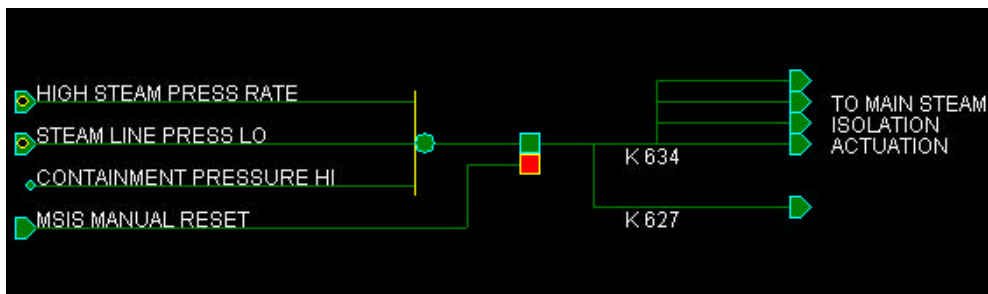
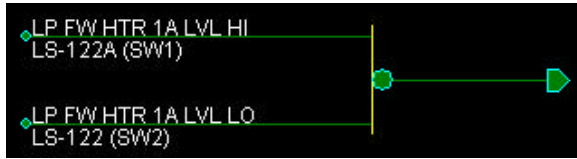


그림 3. 객체화된 논리도면의 예

3.2 도면상의 논리소자들간의 논리적 연결

논리소자들간의 논리적 연결은 논리도면의 신호선과 논리도면 연결객체를 이용한다. 그림 4(a)의 경우 논리소자 객체는 논리연결을 위한 객체로 신호선과 논리연결객체를 이용한다. 신호선은 시작과 끝이 논리소자의 출력부와 입력부에 연결되고 연결정보에 의해 두 객체간의 정보(메시지)를 전달한다. 한편, 논리소자 객체들이 서로 다른 도면이나 연결이 어려운 위치의 논리소자 객체들은 그림 4(b)와 같은 논리연결 객체를 사용한다. 이 객체는 반드시 쌍으로 이루어지며, 각각의 쌍은 상대를 인식하는 ID가 편집 시 주어져야 한다. 그림 4(b)의 경우, 왼쪽 연결자는 AB201 논리도면의 3번 연결자이고, 오른쪽 연결자는 SB204 논리도면의 3번 연결자인데, 각각 상대를 나타내는 ID정보를 표시하고 있다.



(a) 논리소자간 연결



(b) 떨어져 있는 논리소자간 연결

그림 4. 논리소자 객체간의 연결

3.3 경보창 객체와 경보원인 후보 논리와의 논리적 연결

논리객체간의 연결은 앞서 논리도면상의 논리적인 연결뿐만 아니라, 경보창과 경보절차서, 경보창과 경보논리 및 제어논리, 경보창과 경보창, 논리소자객체와 경보절차서와 등과 같은 연결이 필요하다.

(a) 경보창과 경보절차서

- 경보 발생시 해당 경보절차서의 표시 및 추적된 경보원인을 절차서 상에 강조하여 표시하기 위해 사용된다.

(b) 경보창과 경보논리 및 제어논리

- 경보의 원인은 단순히 인터록에 의한 것뿐만 아니라, 기계적인 관계, 역학적인 관계 등에 의해서 시스템적인 원인이 존재할 수 있다. 예를 들면, 'Valve A Open'이라는 경보창이 있다면, 그 경보는 'A Valve open' 경보논리 신호에 의해서 발생하지만, 보다 앞선 원인은 'Valve A Open' 제어신호에 밸브가 기계적인 과정을 거쳐 열리게 된다. 제어논리에 의한 'Valve A Open'과 경보논리에 의한 'Valve A Open' 신호는 기계적인 인과관계가 된다. 경보절차서에서 기술된 경보원인을 이와 같이 기계적인 원인 이외에도 다른 물리적인 원인들을 기술하고 있으므로 이 원인과 관계되는 경보논리나 제어논리 또는 다른 경보논리를 포함하고 있는 경보창을 연결할 필요가 있다.

(c) 경보창과 경보창

- 경보창은 화면상에 동시에 구현하기 어려우므로 시스템그룹별 계층구조적으로 구현하였다. 따라서 상위그룹의 하나의 경보창은 여러개의 하위그룹의 경보창과 논리적으로 연결되어야 한다. 경우에 따라서 (b)와 같이 경보절차서에 있는 경보원인을 추적하기 위해 같은 그룹내의 경보창 또는 다른 그룹내에 있는 경보창과 연결도 필요하다.

(d) 논리소자객체와 경보절차서

- 경보원인 추적이 이루어지면 그 추적된 원인을 경보절차상에 강조하여 표시하여 운전원이 쉽게 파악하기 위해 두 객체간의 연결이 필요하다.

4. 경보논리처리

경보논리처리를 위해서 경보논리입력처리 객체가 공정신호를 주기적으로 감시하면서, 공정신호에 변화가 발생하면 공정신호목록(공정신호와 논리객체간의 연결정보 목록)을 참조하여 이를 입력으로 하는 논리객체에게 메시지를 발생한다. 메시지를 받은 객체들은 즉시 입력신호를 상태연산 함수를 이용하여 자신의 상태를 갱신하고, 연산결과 그 상태가 변하면 상태에 따른 현재 시각을 저장한다. 만약 논리소자 객체의 상태가 true상태에서 false 상태로 변했다면 상태변경시각[0]에

현재시각이, 반대이면 상태변경시각[1]에 저장된다. 그리고 난 뒤, 변경된 상태값을 입력으로 하는 객체들에게 메시지를 보낸다. 메시지에는 메시지를 발송하는 객체의 ID, 최초의 입력 공정신호 ID, 신호발생 시각, 상태값 등이 전달된다.

5. 경보원인 추적

본 방법에서는 모든 논리소자 객체의 상태가 이진논리로 되어 있으므로, 입력상태를 연산하는 함수 f 와 입력상태 및 출력상태는 아래와 같이 표현될 수 있다.

if A_k then $s_k=f(A_k)$
 where,
 k = k -th object in a diagram
 s_k = Output State of an Object k
 $A_k = \{a_{k1}, a_{k2}, \dots, a_{kn}\},$
 $a_{ki}(i=1,2,\dots,n)$ =the i -th input state of the object k
 $f(A_k)$ = Operation with A_k : 상태연산 함수

이 논문에서는 상태 s_k 를 추론하기 위해서, 즉, 원인을 찾기 위해서, A_k 의 원소 중 객체 k 의 상태 s 를 유지시키는 A_k 의 원소를 찾는 함수가 객체의 상태연산 함수이다. 예를 들어 AND 논리소자의 현 상태값 s 가 true이면, A 의 모든 원소가 추적대상 즉, s 의 원인이 된다. 그리고 만약 $s_k=false$ 이면, A_k 의 원소중 false값을 가지는 원소가 추적대상이 되며 true값을 가지는 원소는 추적대상에서 제외된다. Object k 의논리추적 함수는 추적대상을 경보원인추적 객체에게 메시지로 전달하는 것으로 임무를 종결한다.

한편, 경보원인추적객체는 논리소자 객체로부터 추적대상 논리소자의 정보를 받아 더 이상 추적할 논리소자가 없을 때까지 아래 알고리즘을 수행한다.

알고리즘 1.

```
function TrackingCauses(논리소자)
{
  (1) 해당 논리소자객체의 논리추적메소드 호출
  (2) 만약 반환되는 논리소자가 없으면 return;
  (3) 반환된 논리소자들에 대해 다음을 반복
  {
    (3-1) 논리소자가 말단노드(공정신호)인지 검사
    (3-1-1) 말단노드이면 추적경로트리에 등록 후 다음
    논리소자에대해 (3)을 계속
    (3-2) 논리소자가 추적경로트리에 있는지 검사
    {
      (3-2-1) 있으면 다음 논리소자에 대해 (3)을 계속
      (3-2-2) 없으면 TrackingCauses(해당입력논리요소) 수행하고
      해당 입력논리소자 정보를 추적경로트리에 등록
    }
  }
}
```

6. 사용자인터페이스

사용자인터페이스는 사용자로부터 주어지는 마우스, 키보드 이벤트를 처리하고, 논리소자에게 화면 갱신메시지를 보내거나, 논리연결대화상자 제공, 논리소자들이 화면에 상태를 표시할 수 있도록 하며, 경보원인추적결과를 화면에 출력하는 역할을 한다.

6.1 경보창 및 경보논리도면 출력

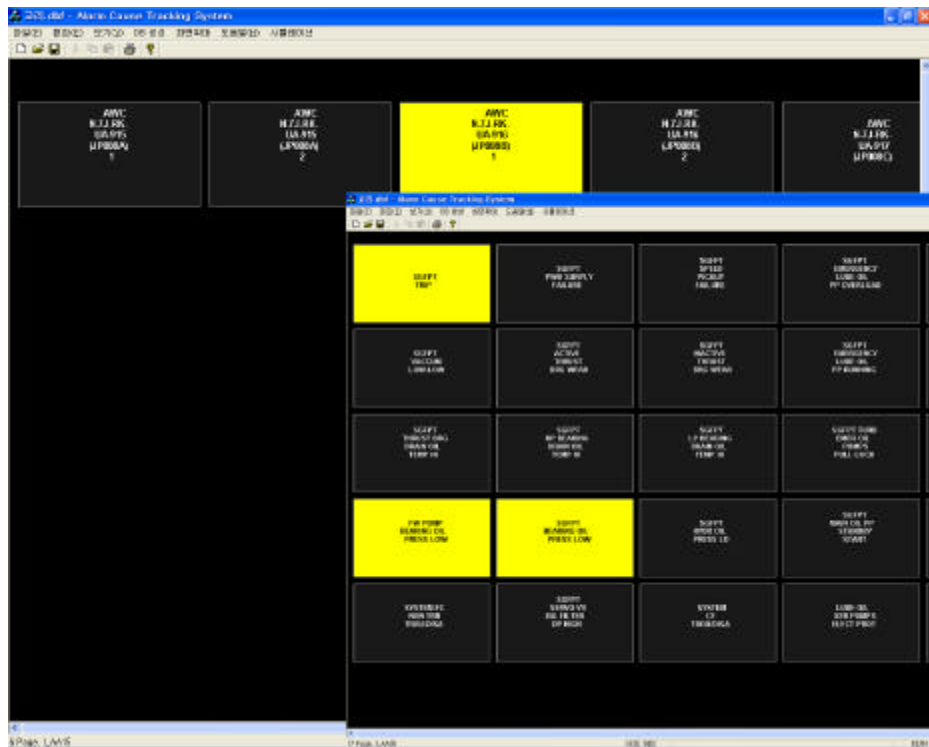
이 논문에서 제안하는 논리객체는 앞서 언급한 바와 같이 모두 그래픽 객체이므로 화면은 모두 실제 경보창, 논리도면의 형태와 유사하다. 단지 논리도면은 도면내의 논리객체의 출력상태에 따라서 적색 또는 녹색을 가지게 되며, 신호선도 전달되는 신호상태에 따라 적색 또는 녹색을 표시하게된다.

논리도면에서 논리연결객체를 클릭하면 객체내의 연결정보를 이용하여 그 객체와 연결된 논리객체가 있는 논리도면으로 이동한다.

6.2 경보원인추적

경보발생시 경보창을 클릭하면 경보원인추적을 시작한다. 경보원인추적은 논리연결경로를 따라 수행되므로, 경보원인추적이 끝난 후에 추적된 경로를 논리도면상에 강조하여 그리게 된다(그림 5 참조).

그리고, 경보절차서-논리객체연결정보를 이용하여 경보절차서에 경보원인에 해당하는 부분을 강조하여 표시한다. 경보원인 추적 중 생성된 추적경로트리를 각 객체별 상태 발생시각과 최초의 원인 등이 출력된다.



(a) 경보그룹선택과 세부경보 선택에 의한 경보원인추적

않으므로 총 논리도면의 수를 약 1000페이지, 논리 도면내 논리 소자의 수를 평균 100개라고 가정하고 모든 논리소자가 동시에 상태가 바뀐다고 가정하면 약 1초 정도가 소요된다. 논리추적의 경우는 원인으로 간주되는 논리소자에 한해서 추적이 수행되므로 그 수가 훨씬 적다. 따라서 발전소 전체 시스템에 대한 논리도면에 대해서도 최악의 경우에 1초를 넘지 않는다. 그리고 논리추적결과 표시는 정해진 화면 내에서 이루어지므로 논리추적결과에 상관없이 거의 일정한 시간이 요구된다.

8. 결론

경보원인을 추적하는 여러 방법들이 많이 제안되었지만 경보원인을 추론하기위해서 대부분 발전소 시스템 전문가들이 작성한 지식 또는 인과관계관련 정보를 구축해서 이를 기반으로 하는 연구가 대부분이다. 이들 연구는 상위레벨의 원인을 추적하고 깊은 지식을 이용하여 효율적인 진단과 예방을 가능하게 하는 방법을 제시하였지만, 이들 지식 또는 정보를 구축하고 검증하는 데에 많은 작업과 전문가 노력이 필수적이라고 할 수 있다. 또한 경험적, 확률적 지식을 이용하는 경우 충분한 검증 이 없으면 자칫 오류를 발생시킬 가능성이 있다.

이 논문에서는 전문가의 지식이나 인과관계에 대한 정보대신 발전소 경보 및 제어논리 및 경보 절차서를 이용하여 경보원인을 추적하는 방법을 제시하였으며, 다음과 같은 효과가 기대된다.

- 그래픽 객체형태의 논리도면으로 구현되므로 가시적이며, 발전소에서 사용하는 논리도면 및 경보절차서와 일치하므로 논리확인 및 편집이 용이하다.
- 논리추적은 논리도면정보와 경보절차서의 정보를 이용하므로 애매함이 거의 존재하지 않으며, 추적(추론)경로의 확인이 시각적으로 가능하다.
- 추적 경로와 원인 발생시각을 알 수 있으므로 경보발생의 선후 구분에 이용할 수 있다.
- 탐색속도는 탐색경로의 길이에 비례하므로 실시간처리가 가능하다.

여기서 제안한 방법은 물리적, 통계적 해석이나 경험적인 지식을 이용하기 어려우므로 이러한 것이 이용 가능한 시스템과 연계하면 유효한 정보를 운전원에게 줄 수 있을 것으로 생각된다.

참 고 문 헌

1. 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, 1994.
2. 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.
3. Baro R. Moum, et. al, CASH: The news alarm system in HAMMLAB, HALDEN Report, HWR-480, 1996
4. L. R. Lupton, P. A. Lapointe and K. Q. Guo, Survey of international developments in alarm processing and presentation techniques, NEA/IAEA International Symposium on Nuclear Power Plant Instrumentation and Control, Tokyo, Japan, 1992.
5. 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.

6. J.T. Kim, et. al, An Evaluation Approach for Alarm Processing Improvement. IAEA Specialists Meeting (IWG-NPPCI) on Experience and Improvements in Advanced Alarm Annunciation Systems in Nuclear Power Plants Chalk River, Ontario, Canada, 1996.
7. I.K. Hwang et. al, An Object-Oriented implementation to improve Annunciation IAEA Specialists Meeting(IWG-NPPCI) on Experience and Improvements in Advanced Alarm Annunciation Systems in Nuclear Power Plants, Chalk River, Ontario, Canada, 1996.
8. J. Reifman and Y. C. Wei, PRODIAG: A process-independent transient diagnostic system-I: Theoretical concepts, *Nuclear Science and Engineering*, Vol.131, pp329-347, 1999.
9. 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
10. J.L. Maryak, et. al, Automated System Monitoring and Diagnosis via Singular Value Decomposition, *Automatica*, Volume 33, Issue 11, pp2059-2063, 1997.
11. 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.
12. 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.