정보학연구 제5권 제2호 2002년 6월(pp.81-94)
The Journal of Information Technology, Vol.5, No.2

원자력발전소용 경보원인추적시스템 설계 및 구현

김정택 · 이정운 · 박재창 · 권기춘* · 류승필**

요 약

원자력발전소에서 경보가 발생하면 운전원이 세부원인을 파악하고 그에 맞는 조치를 취해야한다. 세부원인을 파악하기 위해서는 운전원의 경험, 경보절차서, 경보논리도면 등을 참조하는데, 경보절차서에는 관련원인이 여러 개 있을 수 있으므로 이런 경우에 논리도면을 참조하게된다. 그러나 수백 페이지이상 되는 논리도면을 보면서 경보논리의 상태 및 원인을 추적하는 것은 매우 힘들다.
이 연구에서는 논리도면을 전산화하여 온라인으로 경보발생 시 해당경보의 상세원인을 추적하여 화면에 표시하는 시스템을 제안한다.

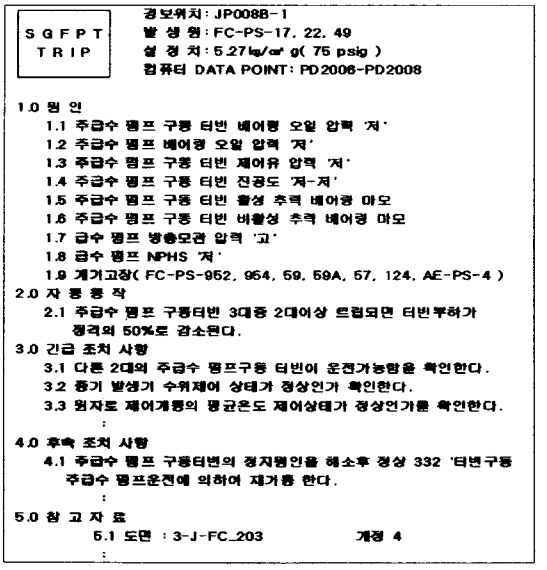
Ⅰ. 서론

전자 및 컴퓨터산업의 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 시스템의 복잡성 때문에 자동화 이전보다도 해야할 작업이 더욱 어려워질 수 있다. 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. 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[1-10] 경보의 원인을 진단하여 운전원에게 알려주는 방법[1-4,11-15] 등이 많이 연구되고 있다. 이들 연구는 상위레벨의 경보를 알려주고, 불필요한 경보를 제거하거나, 이상상태를 진단하여, 이상상태에 해당하는 경보를 찾을 수 있도록 운전원에게 도움을 주도록 하고 있다. 그러나, 이들 방법에 의해 경보창에 있는 많은 경보중 주원인이 되는 경보를 찾았다하더라도 하나의 경보창은 다시 여러 개의 경보입력신호가 할당되거나, 여러 개의 경보 원인에 의해 경보가 발생하는 경우가 있어 하드카피된 논리도면을 통해 해당경보의 상세원인을 다시 찾아야 하는 부담이 존재하게 된다. 그것은 경보원인이 될 수 있는 발전소 공정신호는 수천에서 수만개가 되어 이들 공정신호 각각에 대해 각각

* 세명대학교 컴퓨터과학과
** 한국원자력연구소

경보창을 할당하기에는 제어반이 공간적, 시각적으로 한계가 있으므로 대부분의 원자력발전소는 천여 개 정도의 경보창에 이들 신호들을 조합하여 표시하고 있다. 그림 1.은 고리원전 3,4호기 주급수계통 'SGFPT TRIP' 경보에 대한 절차서의 일부인데, 하나의 경보내에 여러개의 원인이 존재할 수 있음을 보여주는 예이다.

이와 같은 경보가 발생하면 운전원은 여러 원인들 중 직접원인을 찾기 위해서 운전원의 경험적인 지식, 경보논리도면, 현장 기기의 상태, 공정변수의 값 등을 참조하는데, 이상상태가 발생하면 신속한 조치가 요구되는 만큼, 현장 기기의 상태 또는 공정변수를 확인하거나, 수백 페이지에 이르는 경보논리도면을 참조하여 세부원인을 추적하는 것은 운전원에게 있어서 매우 부담되는 일이고, 자칫 오류를 범할 가능성도 있다.


S G F P T
T R I P

경보위치 : JP008B-1
발 생 원 : FC-PS-17, 22, 49
설 정 치 : 5.27kg/㎠ g(75 psig)
컴퓨터 DATA POINT : PD2006-PD2008

1.0 원 인
1.1 주급수 펌프 구동 터빈 베어링 오일 압력 '저'
1.2 주급수 펌프 베어링 오일 압력 '저'
1.3 주급수 펌프 구동 터빈 제어유 압력 '저'
1.4 주급수 펌프 구동 터빈 진공도 '저-저'
1.5 주급수 펌프 구동 터빈 활성 추력 베어링 마모
1.6 주급수 펌프 구동 터빈 비활성 추력 베어링 마모
1.7 급수 펌프 방출모관 압력 '고'
1.8 급수 펌프 NPHS '저'
1.9 계기고장(FC-PS-952, 954, 59, 59A, 57, 124, AE-PS-4)
2.0 자 동 동 작
2.1 주급수 펌프 구동터빈 3대중 2대이상 트립되면 터빈부하가 정격의 50%로 감소된다.
3.0 긴급 조치 사항
3.1 다른 2대의 주급수 펌프구동 터빈이 운전가능함을 확인한다.
3.2 증기 발생기 수위제어 상태가 정상인가 확인한다.
3.3 원자로 제어계통의 평균온도 제어상태가 정상인가를 확인한다.
:
4.0 후속 조치 사항
4.1 주급수 펌프 구동터빈의 정지원인을 해소후 정상 332 '터빈구동 주급수 펌프운전에 의하여 재가동 한다.
:
5.0 참 고 자 료
5.1 도면 : 3-J-FC_203 계정 4
:

그림 1. 'SGFWP TURBINE TRIP' 경보절차서

본 논문에서는 경보발생이후 해당경보의 세부원인을 추적하기 위해서 경보관련 논리도면들을 전산화하고, 경보의 세부원인을 추적하여 보여줌으로써 운전원이 빠른 조치를 취할 수 있도록 하는 운전원 지원시스템을 제안하였다. 본 논문에서는 이러한 운전원 지원시스템의 개발을 위해서 경보 논리도면을 도형적으로 편집하여 논리적인 연결을 하고, 이들 논리상태를 실시간으로 갱신 및 화면표시하며, 경보발생 시, 전산화된 경보논리를 이용하여 경보의 원인을 추적하고 그 결과를 화면에 표시하는 방법을 제시하였다. 제안된 시스템은 고리원자력발전소 논리도면의 일부를 입력하여 논리도면의 실시간 갱신, 논리추적등을 한국원자력연구소 고리원전용 시뮬레이터에 실험하였다.

II. 경보원인 추적시스템

2.1. 시스템 구조

경보원인추적시스템은 경보가 발생하면 경보의 세부원인을 추적하여 그 결과를 화면에 출력하면서, 관련정보를 동시에 보여줌으로써 운전원이 신속하게 경보에 대한 조치를 할 수 있도록 하는 것이 그 목적이라 할 수 있다. 그러기 위해서는 경보 세부원인은 물론, 운전원이 세부원인을 추적, 참조 및 확인이 필요한 경보논리도 및 논리상태에 대한 정보도 필요하다. 그리고 경보추적 경로를 확인하여 원인을 자세히 파악할 수 있도록 다음 세 가지의 기능이 구현한다.

(1) 운전원이 익숙해 있는 논리도면의 형태의 그래픽 화면을 화면에 표시한다.
(2) 화면에 표시된 논리의 상태를 주기적으로 갱신하여 표시한다.

(3) 논리상태 또는 경보의 원인을 추적하여 그 원인과 추적경로를 화면에 출력한다.

제안된 시스템의 기능들은 그림 2.와 같은 모듈들로 구현되며, 이들 주요 모듈들에 의해 생성되는 자료들은, 도형정보, 논리연결, 논리순서리스트, 논리상태파일, 추적경로리스트 등이 있다.

경보원인추적시스템은 외부로부터 발전소 공정신호를 받아서 내부에 구현되는 경보논리를 온라인으로 수행되며, 경보논리에 의해 경보도 발생시킨다. 그러나 이 시스템은 원자력발전소에 이미 설치되어 있는 경보감시시스템과는 완전히 독립적으로 설치 및 수행된다.

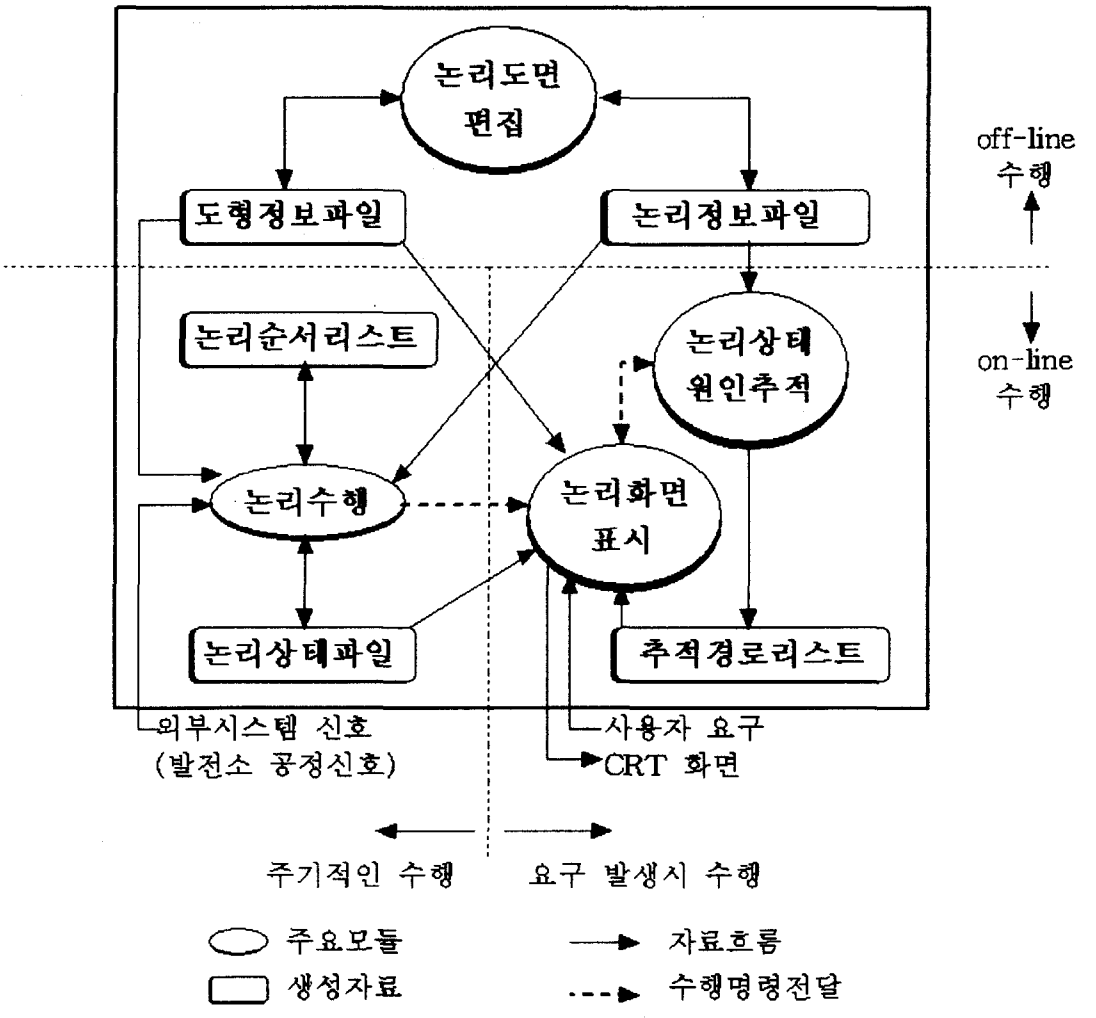


그림 2. 시스템 구성도

2.2. 경보논리의 표현 및 편집

2.2.1. 경보논리의 표현요소

화면에 표시되는 논리도면의 표현은 사용자(운전원)에게 익숙한 원자력발전소에서 사용되고 있는 경보논리도면의 표현방법과 같게 하였으며, 수백 페이지에 이르는 논리도면에 대해서 빠른 접근과 개괄적인 논리 및 세부적인 논리를 모두 참조할 수 있는 계층 구조적 논리도면의 표현이 되도록 하였다.

논리도면을 구성하는 논리요소는 도형적으로, 논리적으로 모양과 기능에 있어서 원자력발전소마다 큰 차이가 없으며[16,17], 이들 논리요소는 크게 논리소자, 도면연결표시, 신호선, 블록, 문자 등으로 나눌 수 있다.

1) 논리소자

논리소자(logic element)는 AND, OR, DELAY 등을 나타내는 논리를 수행하는 기본적인 요소로 이들의 형태에 있어서, 크기는 대체로 비슷하며, 모양(그림 4.의 'F'로 표시된 사각형부분)은 원, 사각형 등 여러 가지 종류가 있다. 한편, 구조적으로는 논리소자에서 신호를 받는 입력부분은 왼쪽, 신호를 보내는 출력부분은 오른쪽이 기본적인 구조이나, 경우에 따라서, feedback 신호선(신호의 흐름이 오른쪽에서 왼쪽 방향)상의 논리는 좌우가 바뀔 수 있다. 다음 그림 3.는 논리도면에 사용되는 논리소자의 도형적인 구조이다.

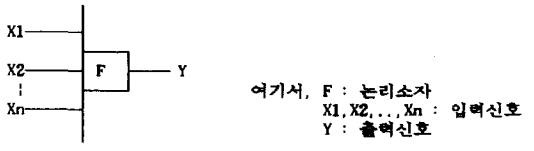


그림 3. 논리소자의 기본적인 형태

그리고 논리소자는 좌측에 입력 신호선을 통하여 다른 소자들과 논리적으로 연결되며, 논리수행 시 좌측에 연결된 논리소자의 출력상태가 입력으로 전달된다. 그리고, 우측에도 신호선을 통하여 다른 소자와 연결될 수 있으며, 이 소자의 출력값이 다른 소자로 전달된다.

2) 도면연결표시

도면연결표시는 도형적으로 연결이 불가능한 논리들을 연결할 때 사용하는 일종의 논리요소(논리소자, 도면연결표시, 블록 등을 통칭)로서, 서로 다른 도면에 있는 논리요소들을 연결하는 도면연결표시(외부 도면연결표시)와 같은 도면 내에 있는 논리들을 연결하는 도면연결표시(내부 도면연결표시)가 있다. 외부 도면연결표시는 연결하고자하는 논리의 도면번호와 논리요소의 위치를 나타내는 번호를 가지고 있고, 내부 도면연결표시는 연결하고자하는 논리요소의 위치를 나타내는 번호만 가지고 있다. 편집시에 이러한 연결표시에 대한 정보는 사용자의 입력에 의해 생성된다.

3) 신호선

신호선은 논리요소의 입력 또는 출력신호의 흐름을 표시하는 선분으로 수평, 수직선 또는 이들의 교차 등으로 이루어져 있다. 신호선은 반드시 논리요소의 출력부에서 다른 논리요소의 입력부에 연결되어야하며 그렇지 못한 신호선은 모두 무시된다. 신호선은 편집시 논리요소들간의 논리적인 연결정보를 준다.

4) 블록

블록은 단순한 검은 사각형 모양 위에 내부를 설명하는 문자가 있는 형태로, 계층적 표현을 가능하게 하는 논리표시이다(그림 4 참조). 자세

한 논리를 도면상에 표시하기 어려운 경우, 이를 생략하여 세부논리를 대표하는 박스만을 화면에 표시하고(그림 4 (b)), 실제의 세부도면은 다른 논리도면에(그림 4(c)) 표현할 수 있도록 하는 기능을 가지고 있다. 따라서 블록을 포함하는 논리도면은 블록내부의 상세논리도면을 논리적으로 포함하는 관계이고, 구조적으로 상위계층이 된다. 블록내부의 상세논리도면에는 또 다른 블록을 포함할 수 있기 때문에 이들은 논리적으로 계층구조적 형태를 가진다.

편집시 블록은 도형적으로 사각형 내부에, 내부논리와 입출력신호를 연결할 입출력포트 (그림 3.의 'A','B','C','D'와 연결된 5각형 도형)들로 만들어 진다(그림 4(a)). 또한, 편집시 블록내부로 전달되는 입력신호연결정보, 블록내부 논리도면을 호출할 수 있는 내부논리도면 호출정보, 호출후 내부논리도면 출력결과가 저장되는 출력신호저장정보 및 블록 위에 쓰여질 문자정보 등도 대화상자를 통해 입력된다.

입출력포트 중에는 그림 4 (a),(b)의 'D'와 같이 내부논리도면(그림 4 (c))과 연결되지 않은 입출력포트가 있는데, 이를 더미포트(dummy port)라 하며, 더미포트는 신호선을 통해 단순히 도형적으로 다른 블록들과 연결할 수 있도록하는 입출력포트로서, 이 연결은 블록간의 제어순서를 정하는데 이용된다.

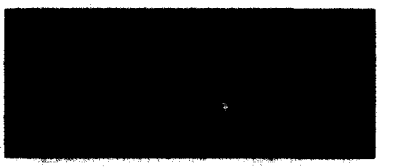
5) 문자

논리도면에 사용되는 문자는 영문자로, 도면을 설명하는데에 사용되는 단순자료이며, 논리적인 역할은 수행하지 않는다.

2.2.2. 도면편집

논리편집은 편집기상에서 이루어지는데 논리편집기는 앞서 설명한 5가지 도형적요소를 객체화하여 객체와 객체간의 연결로 논리적인 연결이 이루어지도록 하였다. 예를 들면, 두 개의 논리요소를 신호선으로 연결하면, 신호선 객체는

(a) 편집시 블록의 내부구조



(b) 애뮬레이션시 블록의 화면표시

(c) 내부논리도면

그림 4. 블록의 구조

두 논리요소의 정보를 상호 교환하여 전달하고, 두 논리요소는 그 정보에 의해서 논리적으로 연결된다.이렇게 연결된 논리정보는 논리정보파일에 저장된다. 각각의 논리요소들은 심볼테이블(Symbol Table)에 표시하였고(그림 5. 참조), 도면편집은 심볼테이블로부터 해당하는 논리요소를 원하는 화면의 위치에 놓음으로써 편집이 가능하다.

2.3. 논리의 순서화 및 논리의 수행

2.3.1. 논리의 순서화

논리도면상에 2차원 방향성 그래프(directed graph)형태로 연결되어 있는 논리요소들은 주기적으로 일정한 시간 내에 적어도 한번이상 수행되어야 한다. 이와같이 2차원 배열로 되어있는 논리요소의 수행방법으로 각 논리요소의 입력신

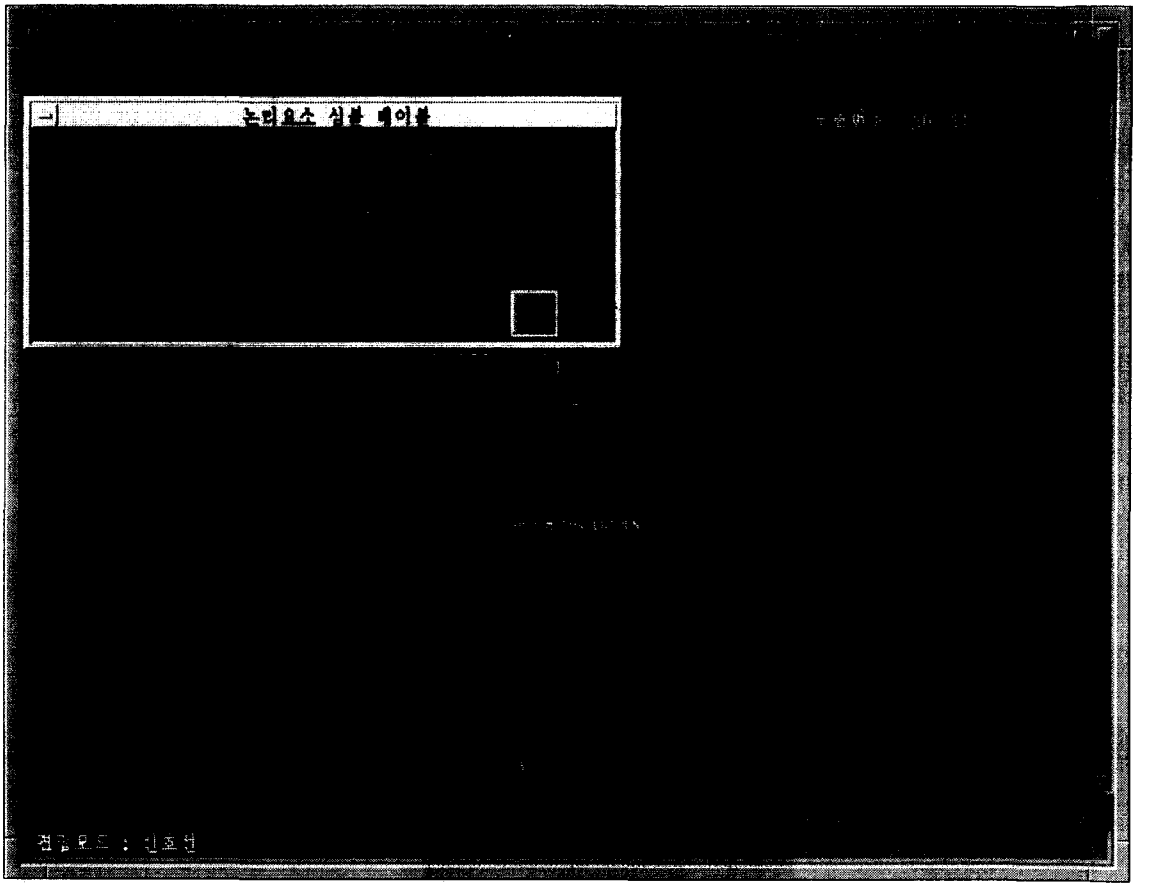
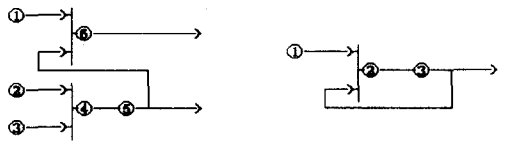


그림 5. 논리도면 편집의 예

호가 모두 준비될 때(갱신되었을 때) 그 논리요소를 수행하는 방법이 있다[18]. 입력신호는 도형구조상 대부분 왼쪽에서 오른쪽으로 흐르므로 이에 따라 논리요소도 왼쪽에서 오른쪽으로 수행하는 것이 자연스럽다. 그러나 경우에 따라 그렇지 않은 경우도 생기는데, 이런 경우에 이들에 대한 순서를 조정할 필요가 있다. 그림 6 (a)의 경우는 논리요소 ⑥은 ⑤보다 왼쪽에 있지만, 신호의 흐름상 ⑤는 ⑥보다 선행되어야 한다. 그러나 그림 (b)의 경우에는 ②와 ③이 서로 신호를 주고받는 loop를 형성하고 있다. 이 경우에 ②와 ③중에서 어느 것을 먼저 수행하는가에 따라 그 결과가 달라질 수 있는데[16], 대체로 ②를 먼저 수행하고 ③을 다음에 수행하게 된다. 이때 논리요소 ②의 입력신호인 ③의 출력값은 이전 cycle에 수행되었던 값을 사용한다.



(a) 피드백이 없는 논리도 (b)피드백이 있는 논리도

그림 6. 순서화된 논리요소들의 예

따라서, 본 방법에서는 위의 내용을 반영하여 아래의 규칙을 정하고, 이에 따라 모든 논리요소들의 순서를 정하였다.

<논리요소 수행 우선순위 규칙>

(1) 같은 도면내의 논리요소는 반드시 그 입력신호들을 발생하는 논리요소들의 수행이 끝난 후에 수행된다.

(2) 같은 도면내에서 서로 출력에 영향을 주지 않는 논리요소들은 상위/좌측우선(상위우선이고, 같은 높이에서는 좌측우선)으로 한다.

(3) 만약 논리요소의 출력신호가 피드백(feedback)되어, 폐회로(closed loop)가 되면 좌측/상위(좌측우선이고, 같은 좌측의 위치이면 상위우선)에 있는 논리요소에게 우선순위를 주고, 나머지에 대해서 위의 규칙들을 적용한다.

예를 들면, 규칙 (1)은 앞의 그림 6 (a)의 경우 논리요소 ②,③,④ 다음에 ⑤이 수행되고, ⑤의 입력을 받아 ⑥가 수행됨을 의미한다. 그리고 규칙 (2)는 서로 신호를 주고받는 관계가 아닌 논리요소들의 순서를 정하는 규칙으로 그림 6 (a)의 ①, ②, ③가 여기에 해당된다. 따라서, 위의 예 그림 6 (a)의 논리요소 순서는 규칙 (1),(2)에 의해서 번호순서로 정해진다.

한편, 그림 6 (b)에서 ②와 ③의 경우 loop를 형성하므로, 규칙 (3)에의해 ②와 ③중에 좌측에 있는 ②가 우선되고, ③은 규칙 (1)에 의해 ②의 바로 다음으로 결정된다.

위의 논리들의 순서는 단지 같은 도면내에서만 적용되고, 서로 도면이 다른 경우에는 논리적으로 연결이 되었더라도 상관하지 않으며, 도면들의 순서는 다음절에 설명된다.

위와같이 논리요소들의 순서가 정해지면, 이들은 순서대로 논리순서리스트에 기록되고, 논리도면의 시작번호(도면내에서 최초로 시작하는 논리요소의 순번)를 논리도면이름과 함께 논리도면시작 테이블에 저장한다. 그리고, 논리도면의 마지막 논리요소 다음에는 논리도면의 끝을 표시하는 논리요소를 추가하여 저장한다.

3.3.2. 논리의 수행

논리의 수행은 앞서 언급한 바와 같이 논리요소들의 순서에 따라 수행되는데, 각각의 논리요

소가 수행하는 작업(함수)은 종류(AND, OR, DELAY 등)에 따라 제안시스템 내부에 미리 정해져 있다. 다만 블록은 논리 연산을 수행하는 대신에, 다른 도면을 호출하는 수행할 수 있도록하는 프로그램 제어하는 함수를 가진다. 논리요소의 수행중 블록의 차례가 되면 수행순서는 블록이 가지고 있는 논리도면의 정보를 논리정보파일로 부터가져와 해당하는 논리도면을 수행하는데 수행하기 전에, 블록의 바로 다음에 수행될 논리요소정보를 Stack에 저장하고 블록이 지정하는 논리도면으로 제어를 이동하여 그 논리도면내의 논리요소를 논리순서리스트에 따라 수행하게된다. 이 논리도면의 논리수행이 끝나면 Stack으로부터 이전 도면의 블록 다음의 논리요소를 계속해서 수행하게된다.

블록을 제외한 모든 논리요소들은 수행순서가 되면, 논리정보파일과 논리상태파일에 참조하여, 입력신호값을 읽어온 다음, 해당함수를 수행하고, 그 결과를 논리상태 파일에 저장한다.

2.4. 경보 원인추적

경보논리는 각각의 논리요소(AND, OR, NOT, 블록 등)를 node로 하는 방향성그래프로 표시할 수 있다. 그림2.는 경보논리를 방향성그래프 형태로 표시한 것으로 노드 N은 입력 I1, I2, ...,Ik(k=1,2,3,...)로부터 신호를 받아 논리를 수행하고 자신의 논리상태를 갱신하게된다. 각각 노드의 상태를 S(N), S(I1), S(I2), ..., S(Ik)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(I1)=0, S(I2)=1, S(I3)=1 이면, S(I2)와 S(I3)는 S(N)=1이 되기 위한 필요조건으로써, S(N)의 원인으로 간주한다. 원인추적은 S(I2), S(I3)의 입력에 대해서도 계속적으로 수행되며, 더 이상 진행할 그래프의 노드가 존재하지 않을 때 원인추적을 마치게 된다. 최종 상세 원인은 그래프상의 말단노드(Terminal)가 되며, 말단노드는 그림 7.의 입력신호에 해당된다.

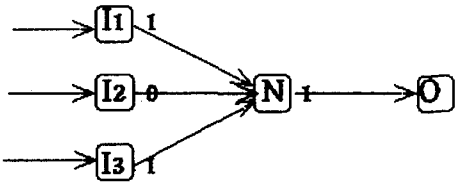


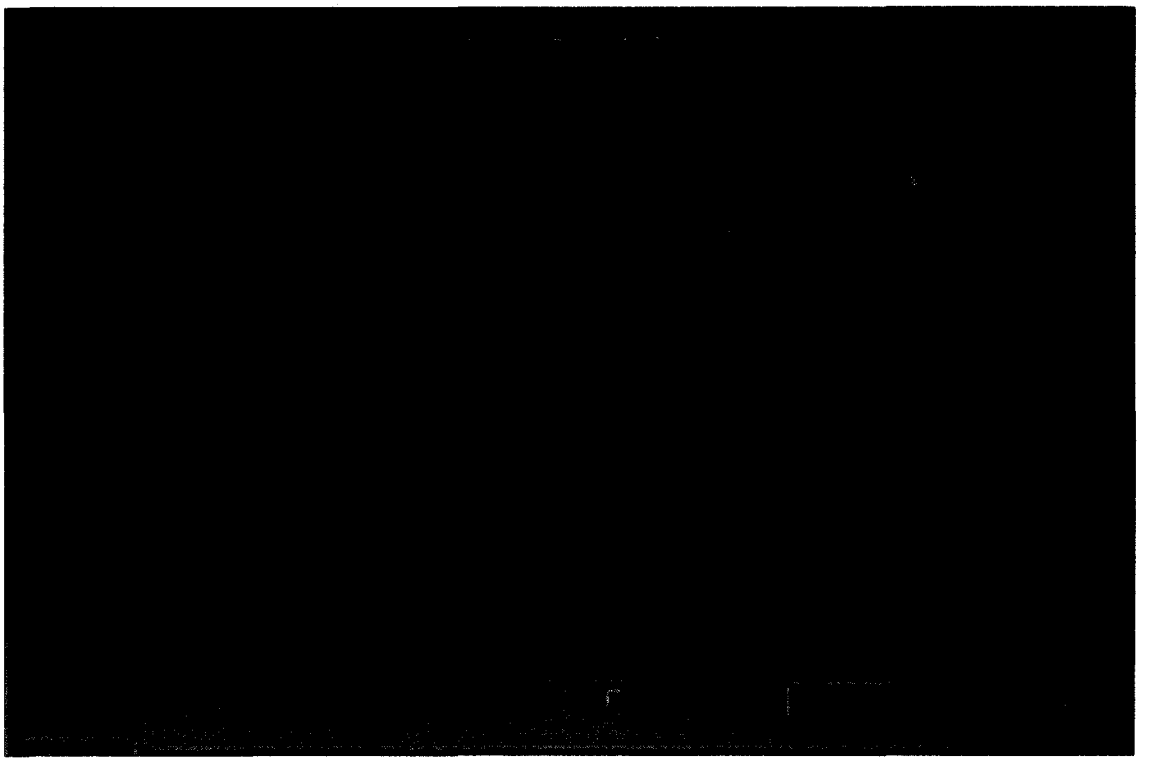
그림 7. 논리상태 원인추적의 예

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. 이 논문에서는 같은 한번 지나간 노드의 방문을 막기 위해 지나간 노드에 대한 추적경로리스트를 작성하고, 리스트에 있는 노드는 다시 방문하지 않도록 하였다. 다음은 이에 대한 알고리즘이다.

아래의 알고리즘 (3-1-1)의 수행결과로 경보원인리스트가 생성되며, 이것이 경보의 세부원인들의 목록이 된다. 그리고, (3-2-2)의 수행결과로 추적경로리스트가 생성되는데, 이것은 경보원인을 추적하는 경로상에 존재하는 논리요소들에 대한 정보를 가지며, 경보논리표시모듈이 추적경로를 화면에 나타내기 위해서 사용된다.

```
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
          논리요소에대해 (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 8. 경보원인추적 알고리즘



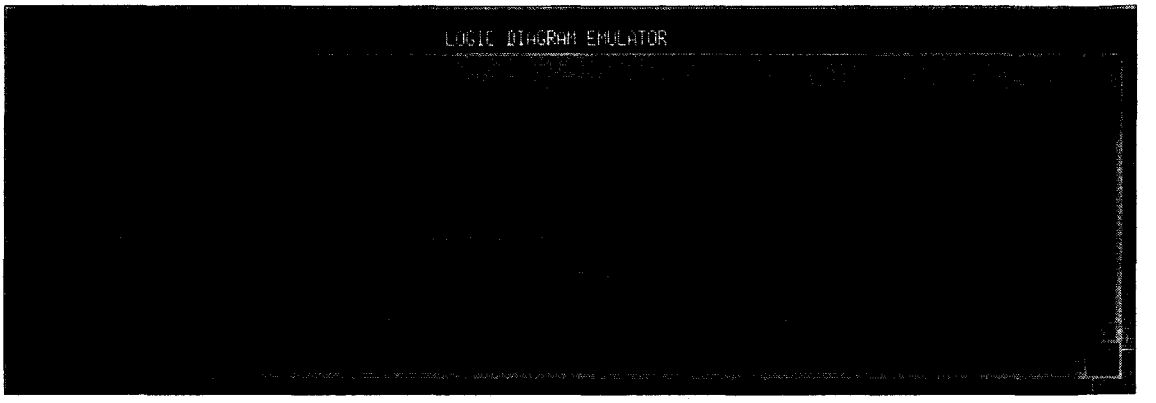


그림 9. 논리상태원인 추적의 예

그림 9.는 논리상태 원인추적의 예를 보인다. 그림 9.에서 맨 위의 그림은 커서위치의 논리요소(SSILS ALARM)를 선택하는 그림이고, 두 번째는 그림은 원인추적결과를 보여준다. 그리고, 맨 아래 그림은 최초의 상태원인이 있는 화면을 나타낸다.

2.5. 논리화면표시

화면에 출력된 논리도면의 논리요소들은 논리상태값에 따라, 논리수행이 끝난 후 갱신된 논리상태파일에 따라 화면에 즉시 반영되는데, 논리화면표시 모듈은 갱신된 논리요소의 상태와 현재 표시되고 있는 논리요소의 상태가 서로 다르면, 화면에 표시된 논리상태를 갱신하게된다. 논리상태는 이진논리를 출력하는 논리요소의 경우 논리요소를 색으로(예를 들어, 출력이 1이면, 적색, 0이면 청색) 표시한다.

그리고, 경보원인추적이 수행되면, 추적결과로 추적경로리스트와 경보원인리스트가 만들어지는데, 추적경로리스트에 있는 논리요소들의 논리적연결관계와 그래픽적 연결관계를 그림 3.의 도형정보 및 논리연결정보를 참조하여 신호선을 강조하여 표시하고(그림 9. 참조), 경보원인리스트에 있는 원인들의 목록을 대화창을 통해 출력한다.

III. 실험 및 토의

본 논문에서 제안된 도구를 실험하기 위해 고리원자력발전소 3,4호기의 보호, 급수, 증기 및 터빈관련 논리도면 156 페이지(논리소자 수 4,862개, 논리요소에서 블록과 도형적인 요소 신호선, 문자 등은 제외됨)를 앞서 설명한 논리도면 편집기로 전산화하고(사용 컴퓨터기종은 HP9000 (712/80) workstation), 한국원자력연구소의 원전 시뮬레이터와 연결하여, 시뮬레이터로부터 발전소 신호를 받아 그 기능 및 성능을 확인하였다. 실험내용은 논리수행확인, 경보추적확인, 논리수행속도 등을 조사하였다. 실험결과, 약 50개의 경보에 대해 여러 가지 원인을 입력하여 테스트한 결과 논리요소의 동작과 경보발생 시 그 원인추적이 100% 정확하게 이루어졌다. 그리고,

논리수행속도는 약 13msec, 156개의 논리도면 내에서 추적경로가 가장 긴(노드수 17개) 논리요소에 대한 논리상태원인 추적속도는, 약 33msec로 측정되었다. 한편 논리요소의 수를 가감하여 논리수행속도를 측정한 결과, 논리수행속도는 논리요소의 수에 비례하였고, 논리상태원인 추적속도는 원인추적경로의 길이에 비례함을 확인하였다.

2차원 그래프의 노드 수가 n개일 때, 임의의 두 노드사이에 이르는 경로의 평균길이는 $l=k\sqrt{n}$(k는 비례상수)로 간주할 수 있다. 만약 발전소에 사용되는 경보관련 논리도면의 수가 1000페이지 이내, 논리요소수 50,000개 이내로[16,17], 모든 논리도면 내의 논리요소가 하나의 방향성 그래프로 연결되었다고 가정하면, 논리소자의 수가 4862개에 비해 경로의 길이가 평균 약 3.2배 정도로 늘어난다고 할 수 있다. 이때, 논리상태추적의 평균속도는 33msec x 3.2 = 약 100msec 이다. 그러나, 실제의 논리도면은 전체가 하나로 연결된 2차원그래프가 아니고, 여러개의 독립된 논리도면으로 구성되어 있으므로 실제의 평균 추적거리는 실험에 사용된 논리도면의 평균 추적거리의 3.2배보다 작을 것으로 사료된다.

IV. 결론

원자력발전소에서 이상상태로 인하여 경보가 발생하면 그 원인을 정확히 파악하고, 이에 대한 조치가 신속하게 이루어져야한다. 이를 위하여, 경보를 축약하거나, 경보의 원인을 진단하여 그 원인을 개괄적으로 제시하는 연구는 많이 이루어졌지만, 경보의 세부원인을 추적하는 연구는 거의 없었다. 경보발생시 운전원은 경험적 지식을 이용하거나, 운전절차서를 통하여 그 원인을 추측하고, 보다 자세한 사항을 위해서는 논리도면을 확인하거나, 기기의 상태를 확인하는데, 이상상태가 발생하면 이에대한 신속한 조치가 요구되는 만큼 이러한 과정은 운전원에게 부담되는 작업이며 자칫 실수를 유발할 수 있다.

본 연구에서는 원자력발전소에서 사용하는 논리도면을 전산화하여 경보가 발생하면 그 경보의 상세원인을 추적하여 그 결과를 화면에 나타내고, 또한 그 추적경로를 표시함으로써 운전원이 원인을 확인하고 그 원인에 해당하는 절차를 신속히 수행할 수 있도록 도움을 주는 시스템을 개발하였다. 한편, 시스템을 한국원자력연구소 원자력발전소 모의장치에 연결하여 테스트한 결과, 경보원인추적이 평균 0.1초 이내에 이루어질 수 있음을 확인하였다.

또한, 제안된 시스템은 경보원인추적 뿐만 아니라, 운전원이 원자력발전소의 논리상태와 원인들을 감시하는데 이용할 수 있고, 원자력발전소와 같이 대규모 논리를 설계한 후에 설계된 논리들의 기능 확인 및 수정하거나, 발전소에 사용되는 수많은 분량의 문서화된 논리도면을 전산화하는 데에 도움이 될 것으로 기대된다.

참고문헌

1. AP600 Design Workshop Information Package Instrumentation & Controls/Man-Machine, Westinghouse, 1990.
2. *Nuplex 80+ Advanced Control Complex*, ABB CE, 1995.
3. J.M. OHara, et. al, Human Factors Engi-

neering Guidance for the Review of Advanced Alarm Systems, NUREG/CR -6105, U.S. Nuclear Regulatory Commission, 1994.

4. 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, *pp*279-295, 1974.
5. 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.
6. Baro R. Moum, et. al, CASH: The news alarm system in HAMMLAB, HALDEN Report, HWR-480, 1996
7. 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.
8. 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.
9. 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 PlantsChalk River, Ontario, Canada, 1996.
10. 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.
11. J. Reifman and Y. C. Wei, PRODIAG: A process-independent transient diagnositic system-I: Theoretical concepts, Nuclear Science and Engineering, Vol.131, pp329-347, 1999.
12. 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
13. J.L. Maryak, et. al, Automated System Monitoring and Diagnosis via Singular Value Decomposition, Automatica, Volume 33, Issue 11, pp2059-2063, 1997.
14. 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.
15. 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.

16. *Logic Diagram(KNU 5&6)*, 한국전력공사, 1984.
17. 영광 3,4호기 *Logic Diagram*, 한국전력공사, 1995.
18. R.W. Lewis, Programming industrial control systems using IEC 1131-3, the Institution of Electrical Engineers, pp126-136, 1995

Design and Implementation of a Alarm-Cause Tracking System for Nuclear Power Plant

Jung-Taek Kim · Jung-Woon Lee · Jae-Chang Park · Kee-Choon Kwon* · Sung-Pil Lyu**

Abstract

When a alarm is happened in nuclear power plant, operator tries to identify the direct and specific causes of the alarm and to do proper actions to mitigate the effect of it. To recognize the specific causes of it, the operator uses his experiences, alarm procedures, logic diagrams and so on. But, if the alarm procedure described many causes of the alarm unfortunately, it is very difficult for the operator who has no experience on the alarm to search the causes of it in hundreds of logic diagrams when emergency.

In this study, a system is proposed, which tracks and displays the causes of alarms on-line from computerized logic diagrams.

* Department of MMIS, Korea Atomic Energy Research Institute/Design and Implementation of a Alarm-Cause Tracking System for Nuclear Power Plant)
** Department of Computer Science, Semyung University