2004 춘계학술발표회 논문집
한국원자력학회

# 안전- 필수 소프트웨어에 관한 안전성분석 체계구성에 대한 연구

A Study of Software Safety Analysis System for Safety-Critical Software

장훈선, 신현국, 장영우, 정재천, 김재학, 한희환
Hoon-Seon Chang, Hyun-Kook Shin, Young-Woo Chang ,
Jae-Cheon Jung, Jae-Hack Kim, Hee-Hwan Han
한국전력기술주식회사

손한성
Han Seong Son
한국원자력연구소

요약

소프트웨어 수명주기에 걸쳐 안전성 분석이 수행되는 안전-필수 소프트웨어에 대하여 안전성 관련 중요 인자 및 요건을 추적하고, 단계별로 안전성 분석을 수행하는 절차 방법론에 대해 기술하였다. 개념단계에서는 안전성분석 체계를 구축하기 위한 계획, 조직 등의 점검사항 및 시스템 관련 고장모드 및 영향분석(Failure Modes and Effects Analysis)을 수행하였다. 요건단계에서는 HAZOP(Hazard and Operability ) 기법을 이용하여 안전성에 관련한 파라미터 및 구현하여야 할 소프트웨어 항목 도출 및 체크리스트 작성에 의한 워크스루(Walk-Through)를 통하여 타당성 평가 및 수행여부를 점검하였다. 설계단계에서는 시스템을 안전하게 구동하도록 설계된 소프트웨어 모듈의 알고리즘을 안전성 측면에서 고장수목(FTA: Fault Tree Analysis) 기법을 이용하여 위험요소 등을 분석하였다. 적용 사례연구는 원자력발전소 계측제어 소프트웨어 중 원자로보호계통의 가압기 저압력 트립 알고리즘을 대상으로 실시하였다. 시험단계에서는 개념단계의 고장모드 및 영향분석(FMEA)을 기반으로 제공된 시험항목에 대한 시험 수행여부를 점검하였다. 소프트웨어 수명주기 동안에 수행되는 안전성 관련 업무에 대하여 체계적인 표준을 작성하여 업무를 수행함으로써 중요 안전요인을 집중적으로 확인할 수 있었으며, 아울러 요건추적 기반의 CASE 도구를 사용하여 개념단계부터 시험단계까지의 안전성 관련 중요 항목들을 추적함으로써 효율성을 높일 수 있었다.

Abstract

The core factors and requirements for the safety-critical software traced and the methodology adopted in each stage of software life cycle are presented. In concept phase, Failure Modes and Effects Analysis (FMEA) for the system has been performed. The feasibility evaluation of selected safety parameter was performed and Preliminary Hazards Analysis list was prepared using HAZOP(Hazard and Operability) technique. And the

check list for management control has been produced via walk-through technique. Based on the evaluation of the check list, activities to be performed in requirement phase have been determined. In the design phase, hazard analysis has been performed to check the safety capability of the system with regard to safety software algorithm using Fault Tree Analysis (FTA). In the test phase, the test items based on FMEA have been checked for fitness guided by an accident scenario. The pressurizer low pressure trip algorithm has been selected to apply FTA method to software safety analysis as a sample. By applying CASE tool, the requirements traceability of safety critical system has been enhanced during all of software life cycle phases.

## 1. 서 론

원전 계측시스템의 안전계통에 사용되는 안전-필수 소프트웨어는 소프트웨어 안전계획서 (Software Safety Plan) [1]를 작성하게 되어있으며 계통이나 운전에 영향을 주는 정도에 따라 소프트웨어의 등급이 정해진다. 소프트웨어의 확인 및 검증 공정과 안전성 분석 공정은 수행관점과 적용하는 기법에 있어서 서로 다른 면에서 출발하고 있다. 그러나 실제로는 동일한 계통에 대하여 업무를 수행하기 때문에 독립적으로 수행하기보다는 그림 1 에서 보는 바와 같이 검증 팀에서 안전성관련 독립 검토자를 두어서 서로 유기적으로 업무를 보완하여 수명주기 동안에 업무를 수행하는 것이 효율적이다.

소프트웨어 개발은 공식적으로 정의된 소프트웨어 수명주기(Software Life Cycle)에 따라 수행된다. 이와 아울러 원자력발전소에 적용되는 규제요건 및 기술기준 등을 준수하여야 한다. [2]

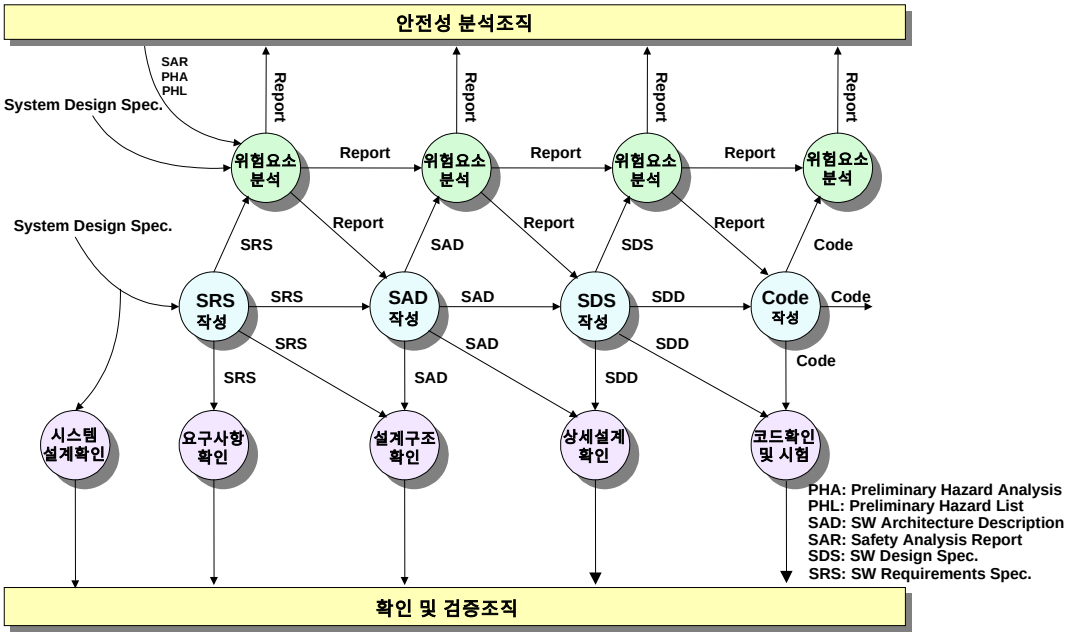


그림 1. 안전성 관련 조직 구성도

일반적인 소프트웨어 안전성 분석 절차는 소프트웨어 개발관리 책임, 소프트웨어 위해도 분석, 소프트웨어 안전성 요구사항 확립 및 설계개념 정립, 검토, 코드확인 및 검증, 위험도 평가 순으로

구성된다.[3]

본 논문에서는 개념단계의 고장모드 및 영향 분석과의 연계성, 설계단계 중요 위험인자의 도출 및 분석, 시험단계에서 시험항목의 연계하는 방법에 대한 체계에 대해 중점적으로 고찰하였다.

## 2. 본론

### 2.1 소프트웨어 수명주기 단계별 안전성 분석

위해도 분석은 특정한 계통의 위해요소들을 정의하고 평가하여 그 위해요소를 제거하거나 허용 가능한 정도로 위험을 감소시킬 수 있는 방안을 제시 또는 제안하는 과정이다. 종래에는 위해도 분석은 소프트웨어를 고려할 필요가 없었다. 그러나 소프트웨어가 점점 더 많은 계통에 포함되고 그 고장이 계통에 영향을 주는 사례가 많아지면서 소프트웨어 위해요소를 정의하는 것이 중요하게 되었다. 계통에서 요구되는 기능을 구현하기 위하여는 하드웨어에 연계된 신호들이 소프트웨어에 의해 이용하게 되므로, 계통과 소프트웨어를 분리하여서 안전성을 분석하는 것은 어려운 일이다. 소프트웨어 위해도 분석은 소프트웨어 안전성 분석의 일부분이다. 소프트웨어 안전성을 위한 공정은 크게 소프트웨어 위해요소들 또는 소프트웨어와 계통사이의 연계(Interface)와 관련된 위해 요소들을 제거하거나 제어하도록 제안하고 있다. 소프트웨어 위해도 분석은 요구사항, 설계, 코드, 사용자 연계 및 변경에 대한 분석을 포함한다. 소프트웨어 위해요소는 소프트웨어가 부적절하게 개발되거나, 부정확하게 정보를 처리하거나, 또는 정보를 전달해야 하지만 전달하지 못할 때 발생한다.

일반적으로 제안되는 소프트웨어 안전성 프로그램은 표 1 에서 보는 바와 같이 개념단계부터 시험단계까지 단계별로 위험요소를 분석하는 것이 필요하다.[3] 즉 영국 국방성에서 제안하는 Mod 00-56[4]과 미국 원자력 안전위원회의 NUREG- 6430[5]에서 요구하는 프로그램의 전 영역을 분석단계로 설정하는 것이 요구된다.

표 1. 소프트웨어 안전성 프로그램 항목

| 번호 | 계통위해도 분석 (Mod 00 –56) | 소프트웨어 위해도 분석(NUREG 6430) |
|---|---|---|
| 1 | 예비위해도 확인 | |
| 2 | 예비위해도 분석 | |
| 3 | 안전성 검토 | |
| 4 | 계통 위해도 분석 | 소프트웨어 요건 위해도 분석 |
| 5 | 계통 위험(Risk) 평가 | 구조설계 위해도 분석 |
| 6 | 독립 안전 감사 | 상세설계 위해도 분석 |
| 7 | | 코드 위해도 분석 |
| 8 | | 소프트웨어 안전성 테스팅 |
| 9 | | 소프트웨어 변경 위해도 분석 |

### 2.2 소프트웨어 안전성 분석 절차

소프트웨어 안전계획은 계통 안전 현안이 소프트웨어 개발 중에 적절히 고려되도록 하는 기본 문서이다. 만약 소프트웨어 안전계획이 없다면 안전 현안들이 적절히 고려되지 않거나, 또는 해결되지않을 수 있다. 따라서 안전계획서에는 분석수행조직, 안전성분석책임, 소프트웨어 안전성

활동관리, 그리고 위험요소와 비정상조건이나 사건 등을 다루기 위해 각 단계마다 수행되어야 할 분석 사항들을 반영하여야 한다.

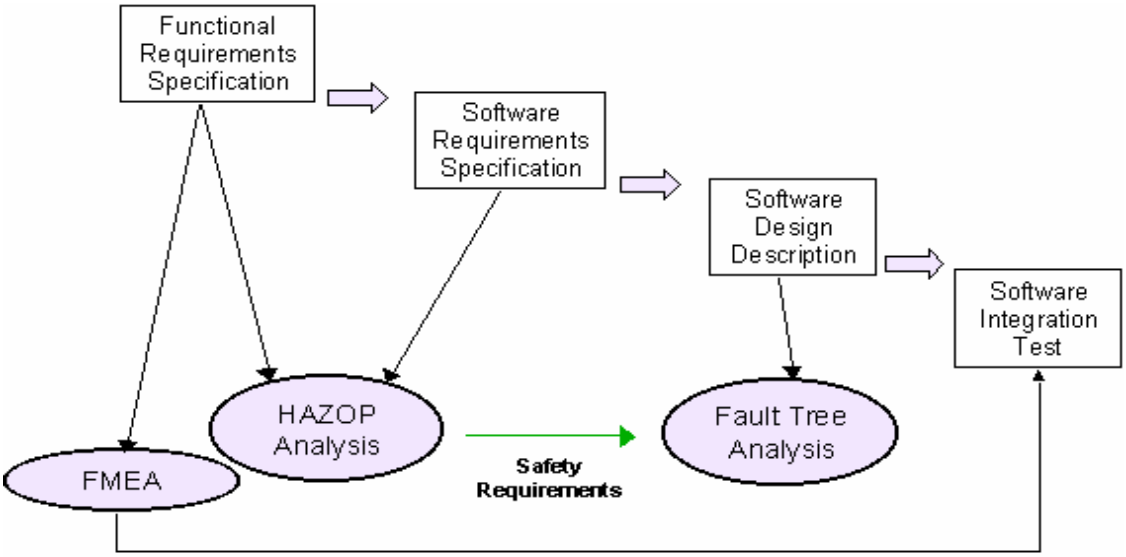


그림 2. 소프트웨어 안전성분석 절차

그림 2 에 제안한 안전성 분석절차는 개념단계에서 발생할 수 있는 소프트웨어 관리절차, HAZOP 방법에 의해서 도출된 안전성 관련 항목처리, FTA 에 의한 모의실험 및 시험단계까지의 안전요건을 효율적으로 관리할 수 있으며 하나의 요건에 부여되는 종속의 충족 요건들을 확인할 수 있다는 장점을 가지고 있다. 본 연구를 통해 제안된 안전성 분석절차의 특징은 다음과 같다.

- 요건단계에 대한 HAZOP 분석을 통해 안전성 관련 중요 인자 및 안전성 요구사항 도출
- 설계단계에서의 고장수목분석 적용
- FMEA 결과를 시험단계 안전성 분석에 활용

2.3 소프트웨어 안전성분석 조직 및 공정확인

소프트웨어 요구사항 안전성 분석은 소프트웨어의 어떤 측면을 표현하는 다양한 그룹의 요구사항을 안전성 측면에서 분석하는 것이다. 소프트웨어 안전성 분석 동안 고려할 기능특성은 정확도, 기능성, 신뢰성, 강인성, 안전성, 보안성 등에 대한 요구사항 들이며 이러한 특성에 대하여는 표 2 와 같은 체크리스트를 작성하여 수행하는 것이 효율적이다.

표 2. 체크리스트에 의한 수행조직 및 책임 검토표

| 공정특성 | 점검 항목 | 부합여부 (Y/N) | 위험영향 (결과) | 설계권고사항 | 비고 |
|---|---|---|---|---|---|
| 완전성 | 소프트웨어 위해도 목록이 존재하는가? | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 일관성 | 안전성 분석자는 안전관련 소프트웨어 요구사항이 계통명세와 안전성분석보고서 내용과 일치하는가 | | | | |
| 정확성 | 안전성 분석자는 안전관련 소프트웨어 요구사항이 정확한지를 확인하였는가? | | | | |
| 검증성 | 안전성 분석자는 안전관련 계획서 등의 내용을 확인하였는가? | | | | |
| 추적성 | 안전성 분석자는 안전관련 소프트웨어 요구사항의 반영사항을 확인하였는가? | | | | |

2.4 HAZOP 에 의한 안전성 분석

HAZOP[6]에 의한 안전성 분석은 사고가 설계 또는 운용상 의도한 것에서 벗어났을 때 발생하는 것을 가정하고 설계에서 예상한 운용을 하였을 경우 일어날 수 있는 모든 가능한 불일치(deviation) 상황과 그와 관련된 모든 위해 요소를 찾으려고 하는 것이다. 이에 따라 시스템을 구성하는 각 실체에 대하여 그림 3 의 절차에 따라 위험(Hazard) 발생의 각 질문에 대한 원인, 공통원인을 기록하고, 이에 따라 초래될 수 있는 결과의 위험영향, 위험도 등을 기록하고 권고사항을 기록한다.

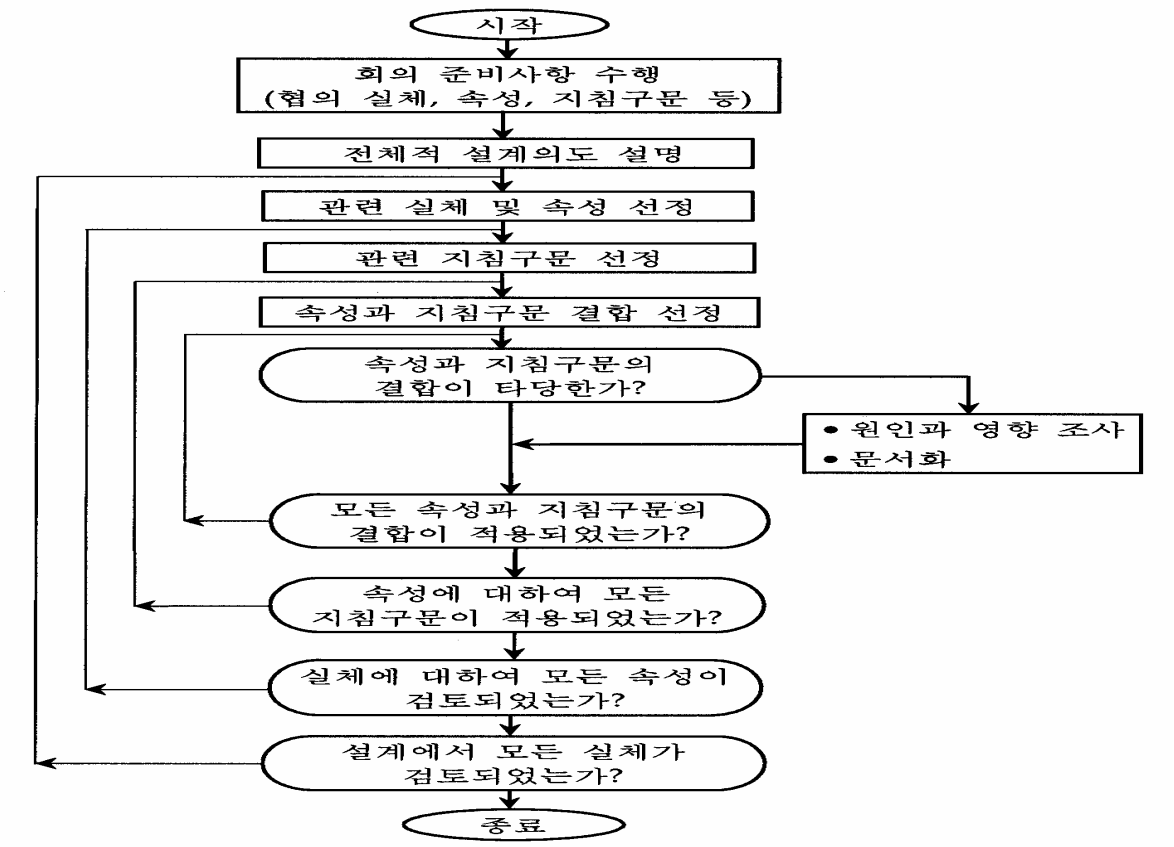


그림 3. HAZOP 절차 개요도

표 3 의 지침 구문 (guide phrase)[5]를 사용하여 의도된 동작에서 불일치가 일어났을 때 발생 가능한 모든 위해도에 대하여 팀이 구성되어 설계에 대한 의문을 제기하는 형식으로 수행된다. [6]

표 3. 소프트웨어 HAZOP 을 위한 지침구문

| 품질 | 관점 | 단계 | 지침 문구 |
|---|---|---|---|
| 정확도 (Accuracy) | 센서 및 작동기 (Actuator) | RADC | 모든 영점 및 1 점에서 고착(stuck) |
| | | RADC | 최소 범위 이하 또는 이상의 수치 |
| | | RADC | 잘못된 데이터 유형이나 데이터 크기 |
| | 운전원 입력 및출력 | RA | 허용범위 이하 또는 이상의 수치 |
| | | RA | 잘못된 데이터 유형이나 데이터 크기 |
| | 계산 | RDC | 계산결과가 허용오차 한계를 벗어남(너무 낮음) |
| | | RDC | 잘못된 데이터 유형이나 데이터 크기 |
| 신뢰성 (Reliability) | | R | 소프트웨어가 내부(in-service) 테스트에 실패 |
| | | A | 소프트웨어가 복원 실패 |
| 강인성 (Robustness) | | RA | 예상치 못한 입력에 소프트웨어 실패 |
| | | RA | 메시지 손실 소프트웨어 실패 |
| 안전성 (Safety) | | RA | 필요 시 원자로정지를 위한 소프트웨어 실패 |
| | | RA | 원자로 위험상태 인지하는 소프트웨어 실패 |
| 안전 (Security) | | RA | 허가받지 못한 소프트웨어 변경 |
| | | RA | 허가받지 못한 변경소프트웨어 적용 |
| 용량 (Capacity) | 메시지 | RADC | 메시지 내용 부정확 |
| | | RADC | 메시지 전달 비율 부정확 |
| | 시간성 (Timing) | RADC | 입력신호 지연 |
| | | RADC | 프로세싱이 부정확 순서로 작동 |
| 기능성 | | RA | 특정기능이 각 운전모드에 명세된 대로 수행하지 못함. |
| | | RA | 시작조건이 만족되지않았는데 특정기능을 수행함. |
| | | RA | 필요한 행동, 계산 사건들을 완수하기 전에 특정기능을 종료함. |
| | | RA | 특정기능이 부정확한 출력신호를 발생함 |
| | | RA | 특정기능이 부정확한 입력신호를 사용함 |

R: Requirements, A: Architectural Design D : Detailed design, C: Coding

2.5 계통 관련 안전성 분석 절차

시스템 관련 안전성 분석은 조직적인 부분과 계통관련 부분으로 크게 분류될 수 있다. 조직적 측면은 조직의 구성을 위한 행정적인 면과 함께 수명주기 초기단계에 수행되어야 할 문서들이 포함된다. 이러한 문서들은 과제계획서, 품질보증계획서, 확인 및 검증계획서 등이 포함되며 특히 안전에 관련된 사항들을 점검하게 된다. 또 다른 하나는 예비 위해요소 목록(Preliminary Hazard List)에 의한 고장모드 및 영향분석표 (Failure Modes and Effects Analysis)를 작성하는 것이다. 표 4 는 기본적으로 하드웨어와 소프트웨어 사이의 연계, 소프트웨어의 오동작으로 계통에 미치는 영향, 소프트웨어 명령과 응답을 제어하기 위한 계통설계기준 및 소프트웨어 고장안전(Fail Safe) 설계고려사항 등이 포함되며, 계통의 기능 또는 기능상실이 위해요소를 초래하도록 하는 구성요소를 규명하고, 영향을 주는가를 검토하게 된다.

표 4. 고장모드 및 영향분석표

| No | Name | Hazard Description | Hazard Detection | Method Detection | Potential Consequence | Safety Hazard Mitigation | Verification Method |
|---|---|---|---|---|---|---|---|
| 1 | Processor | Numerical Value below or above acceptable range | Entry errors Or Hardware read error | Range limit check | Channel Trip | Channel redundancy | Software testing |
| 2 | Processor | Function is not initialized | Programming error | Interchannel Comparison failure | No trip when it is required or inadvertent trip | Channel redundancy | Software testing & Code inspection |
| 3 | Communication Module | Module stalls/halts status data flow to/from processor | Software error or Hardware error | Trouble Alarm | Loss of in channel communication | Redundant Communication channel | Software testing & Validation test |
| 4 | I/O module | I/O or processor module error | System error | Trouble Alarm | Trip or half Trip, Channel trip | Error flags are monitored by channel redundancy by application | Software testing & Validation test |

2.6 소프트웨어 설계사항 분석

소프트웨어 요구사항에 대하여 개발자는 설계를 수행하고 그 결과를 소프트웨어 설계명세로 표현한다. 일반적으로 소프트웨어 설계 중에서 안전성 관련 인자로 크게 다음과 같이 구분할 수 있다.

- 트립 관련 인자
- 하드웨어 관련 인자 (아날로그 입출력 점검, 디지털 입출력 점검)
- 시스템 관련 인자 (Heart beat 점검, Watchdog Timer)
- 메모리 관련 인자 (Cyclic Redundancy Check, Checksum Check)
- 프로그램 수행 관련 인자 (시퀀스 점검, 응용 블록다이어그램 정상점검)

소프트웨어 위해요소를 규명해내는 작업은 요구사항 검토 후에 시작되며 소프트웨어 코딩이 시작되기 전에 대부분 완성되어야 한다. 이 단계에서는 설계문서 및 시험계획 등을 분석하며 이 결과로 설계문서에 대한 변경이 이루어지며 안전성 요구사항이 소프트웨어 시험에 통합된다. 아울러 코드단계에서는 소스코드와 계통연계 등을 분석한다.

2.7 Simulation 및 시험

요구사항이 추가되거나 변경되었을 경우 요건과 관련한 인자에 대한 반영 및 위해요소 영향 등을 분석하게 된다. 이에 따라 소프트웨어 설계가 재구성되어진다. 소프트웨어 설계명세서는 기능블록다이어그램(Function Block Diagram) 형태로 작성되는 것을 가정하였다. 이러한 기능블록다이어그램을 바탕으로 고장수목을 전개한다. 이 때 상위 사건에 대한 인과관계를 고려할 뿐 아니라 기능 블록다이어그램 상에서 설계자가 범할 수 있는 다양한 형태를 고려하면서 고장 수목을 전개

한다. 그림 4 는 가압기저압력 트립에 대한 설정치 및 변화를 표시하고 있으며 트립기능에 대한 요구사항은 다음과 같이 요약할 수 있다.

- 운전원이 수동으로 설정치 리셋을 개시하면 트립 및 예비트립 설정치가 공정변수 값에 대하여 미리 정해진 값만큼 감소된다.
- 설정치 리셋 개시는 정해진 시간이 경과한 후에 다시 수행할 수 있다.
- 공정변수의 상승 중 다시 하강하게 되면 트립 및 예비트립 설정치는 현재값을 유지한다.
- 공정변수가 상승하여 공정변수와 하한값의 차가 정해진 값보다 커지게 되면, 미리 정해진 값만큼의 차이를 유지하면서 트립 및 예비트립 설정치도 함께 상승한다.
- 트립 비교논리를 통해 트립이 된 이후에는 운전원이 리셋을 시키더라도 트립 및 예비트립 설정치는 현재값을 유지해야한다.
- 운전우회가 있을 경우에는, 공정변수가 운전우회 허용범위 내로 들어오면 운전우회 허용신호가 표시되고, 이 때 운전원이 운전우회를 개시하면 운전우회 상태가 된다.
- 운전우회 허용상태가 정상적으로 운전되어야 하고, 공정변수가 상승하여 운전우회 허용범위를 벗어나면 자동으로 운전우회 상태가 제거된다.
- 운전우회 허용상태가 아닐 경우 주 제어실 및 원격정지실에서 운전우회를 개시하면 운전우회 오류경보가 발생된다.

그림 5 는 가압기저압력 트립 기능의 고장수목 일부를 보여준다. 이는 표 3 에 나타난 지침구문 가운데 “특정기능이 부정확한 출력을 생산하면 어떤 위험을 초래하는가?” 점검표를 사용하여 HAZOP 분석을 수행한 결과로 도출된 시스템의 위해 요소이다.

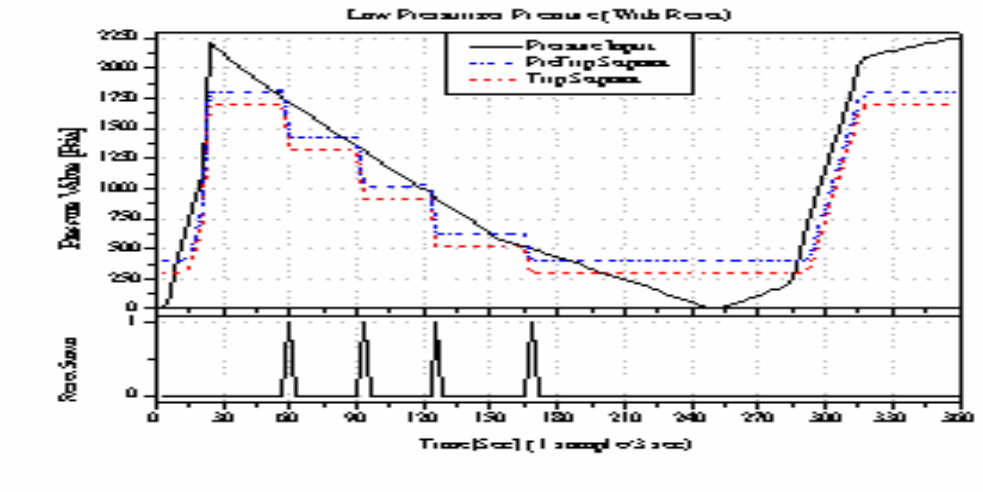

그림 4. 가압기저압력 트립 설정치 변화

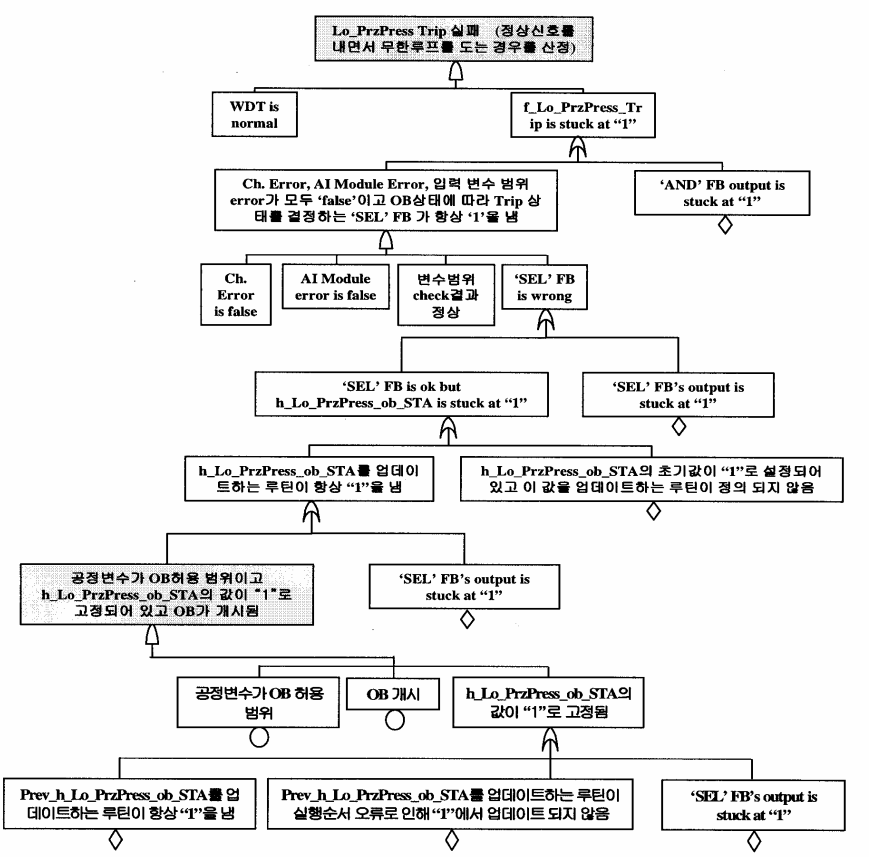


그림 5. 가압기저압력 트립기능의 고장수목도

2.8 소프트웨어 테스트사항 확인

HAZOP 에 의해서 발견된 요건의 변경사항이나 지침구문과 FMEA 를 기반으로 하여 위해도 관련 테스트가 잘 진행되었는지를 점검하게 되며 변경된 연계사항에 대하여 영향 등을 분석하게 된다. 표 5 는 FMEA , HAZOP 지침서 및 테스트의 연계사항을 보여주고 있다.

표 5. FMEA 와 테스트문서 연계

| 번호 | HAZOP 지침서 | FMEA 연계 위험도 번호 | 테스트문서 번호 | 비고 |
|---|---|---|---|---|
| 1 | 허용범위 이하 또는 이상의 수치 | FMEA-4 | VP100-5.1.1 | |
| 2 | 계산결과가 허용오차 한계를 벗어남 (너무 낮음) | FMEA-5 | VP100-5.1.1 | |
| 3 | 특정기능이 각 운전모드에 명세된 대로 수행하지 못함. | FMEA-8 | VP100-5.1.2 | |
| 4 | 시작조건이 만족되지않았는데 특정 기능을 수행함. | FMEA-12 | VP100-5.1.3 | |
| 5 | 필요한 행동, 계산 사건들을 완수하기 전에 특정기능을 종료함. | FMEA-13 | VP100-5.1.4 | |
| 6 | 메시지 전달 비율 부정확 | FMEA-17 | VP100-5.1.5 | |

2.9 CASE 도구를 이용한 안전성 분석 체계 구축

소프트웨어 공학 도구를 이용한 통합 모형은 그림 6 과 같다. 개념단계에서는 안전성체계를 구현하기 위하여 점검되어야 할 항목을 가진 도표, FMEA 도표가 포함되고, 요건단계에서는 위험항목 점검을 위한 지침서가 구성되고, 설계단계에서는 요건이 변경된 인자 또는 안전성 측면에서 구현되는 모듈들의 점검 및 시험단계에서는 FMEA 도표의 항목과 안전성관련 시험항목 등이 연계되고 추적되어 안전성항목을 효율적으로 수행할 수 있다.

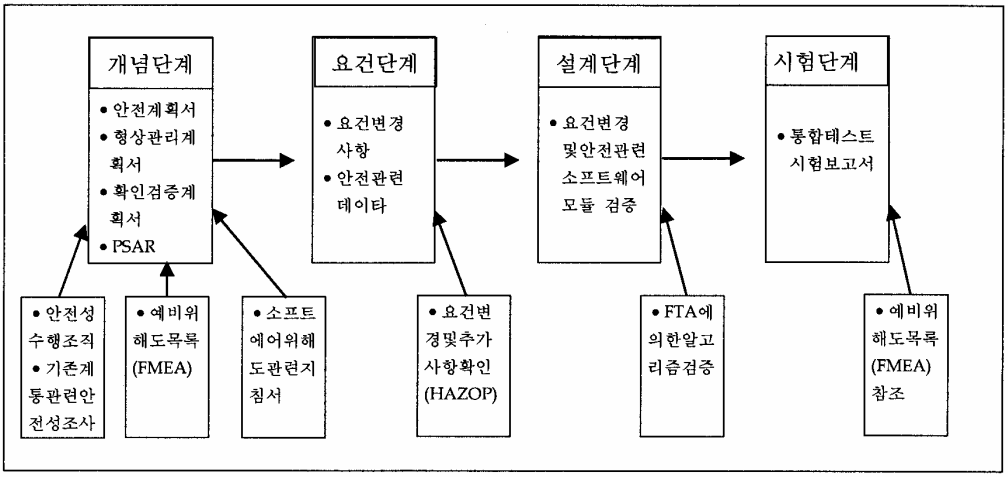


그림 6. 안전성분석 체계 구축 모형도

그림 7 은 계측제어계통의 디지털화에 따라 소프트웨어의 효율적인 생산과 오류를 발견하기 위하여 적용되고 있는 Computer Aided Software Engineering (CASE) 도구에 하나인 요건관리 도구를 이용하여 안전성 인자의 연계성을 구현한 그림을 보여주고 있다.[7] [8]

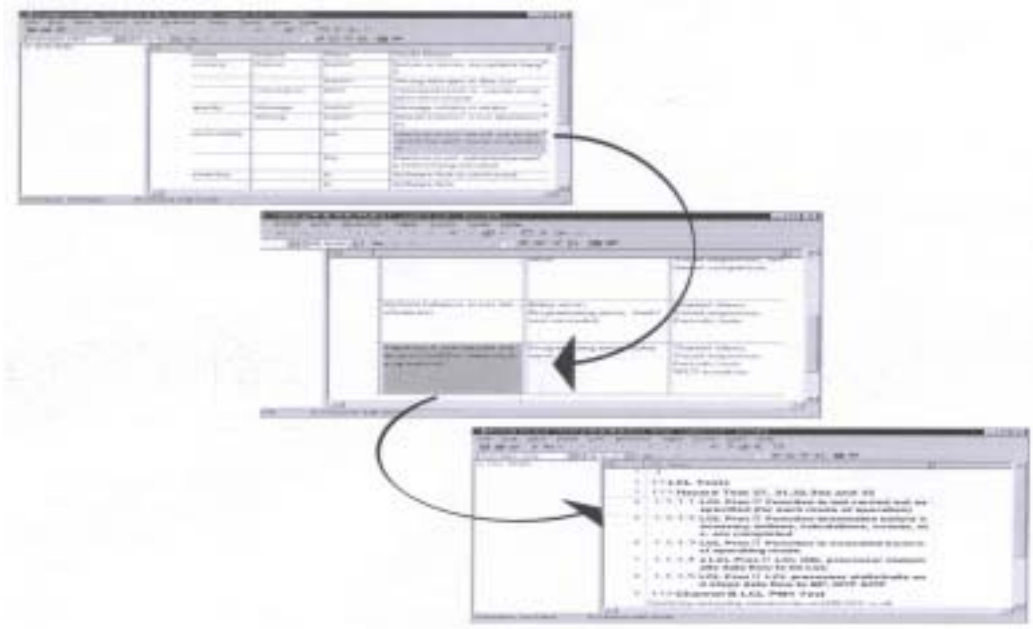

그림 7. CASE 도구를 기반으로 한 안전성 분석 구현

## 3. 결론

원자력발전소 계측제어시스템 소프트웨어 안전성 분석에 대하여 수명주기 동안에 일어나는 안전성 분석 절차를 제안하였다. 안전성 분석과 연계되는 점검표의 작성, 설계 안전성, 안전성 시험 등에 이용되는 여러가지 도표를 이용한 HAZOP 방법 및 소프트웨어 설계 안전성을 점검하기 위하여 고장수목분석(Fault Tree Analysis)을 이용한 안전성분석 방법의 타당성 등이 수행되었으며, 요건 추적 도구를 이용하여 안전에 중요한 요건을 체계적으로 추적하는 방법에 대해 중점적으로 고찰하였다.

이와 같은 안전성 분석 체계의 구축은 복잡한 계측제어 시스템 개발 시 시스템의 안정성을 높이는데 유용하게 사용할 수 있을 것으로 판단된다. 아울러 CASE 도구에 의한 안전성 분석체계를 구축함으로 요건의 변경이나 추가되는 항목에 대하여 효율적으로 분석 수행할 수 있었다.

그러나 좀더 설계변경에 대한 효율적인 타당성을 점검하기 위해서는 여러가지 시나리오를 가정하여 수행할 수 있는 도구의 선정 등에 대한 연구가 향후 진행되어야 할 것으로 판단된다.

## 알리는 말씀

본 논문은 산자부가 지원하는 전력산업연구개발사업의 과제 수행으로 이루어졌습니다. 본 과제를 수행하는데 물심양면으로 지원하고 있는 산자부 및 한국전력연구원(KEPRI) 관계자 여러분께 깊은 감사를 드립니다.

## 참고문헌

1. IEEE Std. 1228-1994, “IEEE Standard for Software Safety Plans”
2. IEEE Std 1074-1995, “IEEE Standard for Developing Software Life Cycle Processes”
3. Nancy G. Leveson, “SAFEWARE: System Safety and Computers,” ISBN 0-201-11972-2, Addison-Wesley, 1995.
4. MoD 00-56, “ Safety Management Requirements for Defense Systems Containing Programmable Electronics”, UK Ministry of Defense (MoD), 1996
5. NUREG/CR-6430, “Software Safety Hazard Analysis’, February 1996
6. Felix Redmill, “System Safety: HAZOP and Software HAZOP”, John Wiley & Sons Ltd, 1999
7. EPRI TR-105989-Vol. 1, “ Software Fault Reducing using Computer-Aided Software Engineering (CASE) Tools, 1995
8. Telelogic AB, “Using DOORS for Requirements Management”, 2003