2003 춘계학술발표회 논문집
한국원자력학회

소프트웨어 요구명세 안전성 분석을 위한 HAZOP 방안

HAZOP Method for Safety Analysis of Software Requirements Specification

이장수, 천세우, 손한성, 차경호, 김장열, 이영준, 권기춘
한국원자력연구소
대전광역시 유성구 덕진동 150

요 약

원전 계측제어계통의 디지털화 추세에 따라 컴퓨터 소프트웨어의 안전성 확보가 중요한 현안으로 부각되고 있다. 이에 따라 소프트웨어 개발공정 각 단계별 안전성 분석 수행을 인허가 기준으로 의무화하고 있다. 소프트웨어 개발공정 단계 중 요구사항 명세 단계에서 안전성 분석을 수행하기 위한 HAZOP (Hazard and Operability) 방안을 제시하였다. HAZOP은 화학공장과 같은 산업에서 시스템 안전성 분석을 위해 성공적으로 사용되었던 기법이다. 이 HAZOP 개념을 소프트웨어 요구사항 명세의 안전성 분석에 활용할 수 있도록 하기 위해 지침구문과 체크리스트를 개발하고 이를 이용한 소프트웨어 HAZOP 절차를 개발하였다.

Abstract

The digitalization of the instrumentation and control system of nuclear power plant makes the safety of computer software be the most important issue. Recently, licensing criteria requires the safety analysis on the product from each phase of the lifecycle. A HAZOP (Hazard and Operability) method for safety analysis of software requirements phase has been suggested. HAZOP is a powerful hazard analysis technique which has a long history in process industries. As the use of digital systems for nuclear power plant becomes more common, it is clear that there is a need for a HAZOP method which can be used effectively with such systems. This paper describes several attempts to derive the guide phrases, checklist and the procedure for software HAZOP.

1. 서 론

원전 계측제어계통의 디지털화 추세에 따라 컴퓨터 소프트웨어의 안전성 확보가 중요한 현안으로 부각되고 있다. 이에 따라 인허가 기관이나 산업표준 기구 등에서

안전성 심사 및 개발기준이 되는 인허가 요건과 국제표준 정립 노력을 기울이고 있다. 본 논문에서는 소프트웨어 개발 공정 단계 중 요구사항 명세 단계에서 안전성 분석을 수행하기 위한 HAZOP 방안을 제시하였다. HAZOP은 Hazard and Operability의 약어로서 화학공장과 같은 산업에서 시스템 안전성 분석을 위해 사용하던 기법이다. 이 HAZOP 개념을 소프트웨어 요구사항 명세의 안전성 분석에 적용하기 위한 방안을 제시한다.

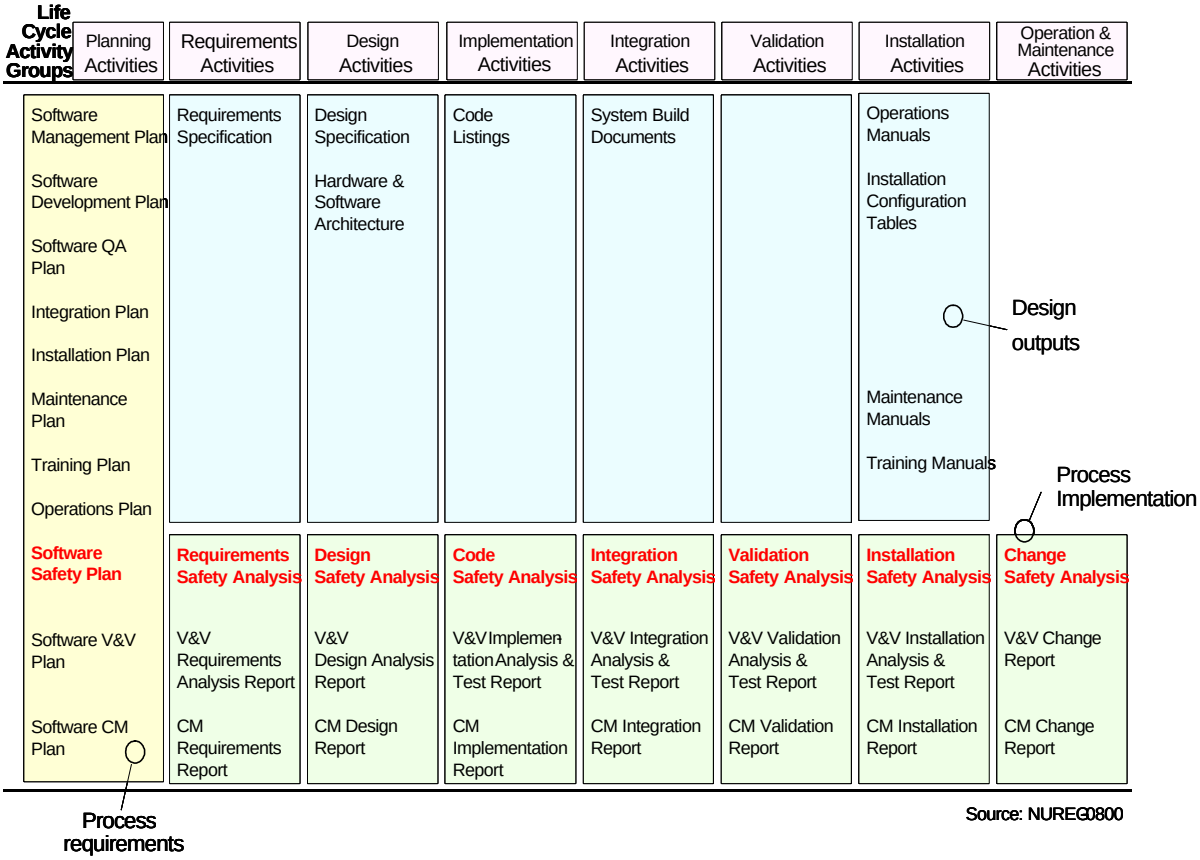

그림 1. 소프트웨어 안전성 분석 허가 기준

현재 한국원자력연구소에서는 디지털 계측제어 소프트웨어 안전성 분석을 위해 일련의 필수안전 소프트웨어 평가 절차서를 개발하고 있다. 본 논문에서 제시한 소프트웨어 HAZOP 방안은 소프트웨어 요구사항 명세 안전성 분석 절차서 개발에 사용하였다. 이러한 절차서는 새로 개정된 미국 표준 검토 계획(Standard Review Plan)에서의 그림 1 과 같은 소프트웨어 안전성 분석 허가 기준을 만족시키기 위한 절차서 이며, IEEE 1228-1994 소프트웨어 안전성 계획 표준에서 요구하는 기준에 따라 그림 2 와 같이 소프트웨어 개발공정 단계별로 안전성 분석을 수행하기 위한 절차이다. 이 절차서는 현재 개발되고 있는 KNICS 원전 디지털 계측제어 소프트웨어 안전성 분석에 사용되고 있다.

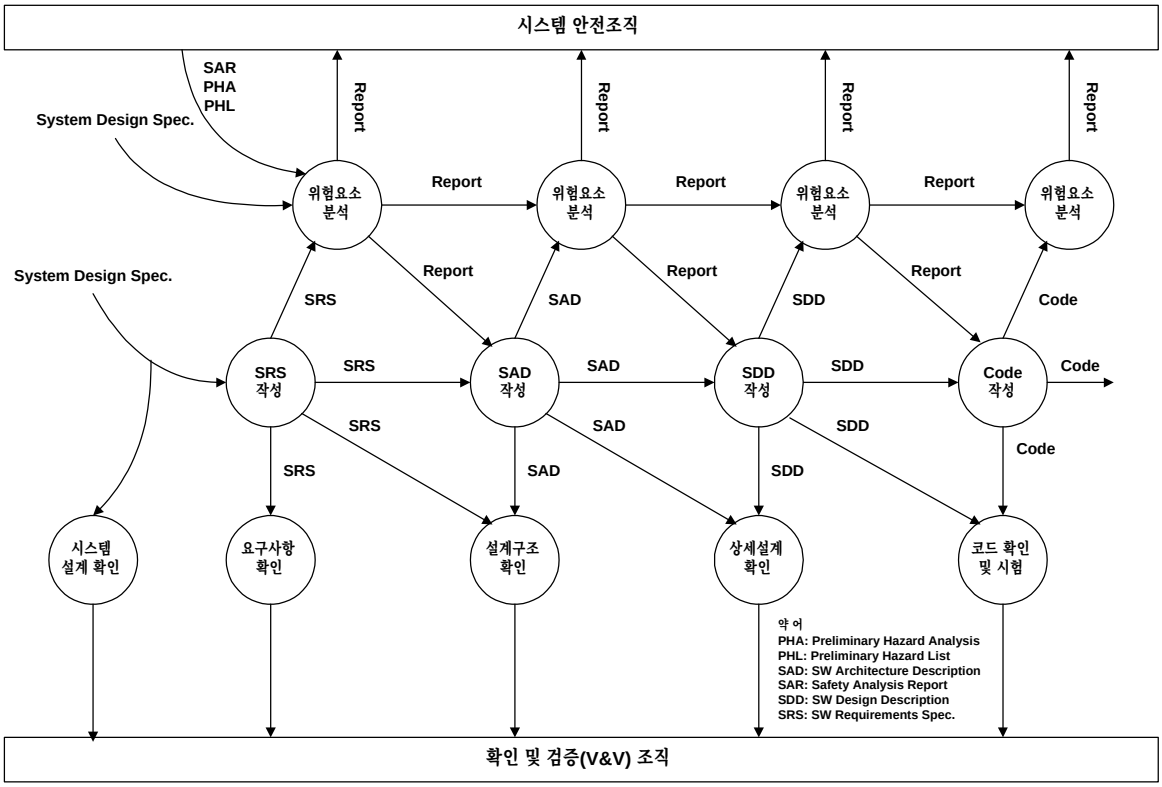

그림 2. 소프트웨어 개발공정 단계별 안전성 분석

소프트웨어 안전계획은 시스템 안전현안이 소프트웨어 개발 중에 적절히 고려되도록 하는 기본 문서이다. 만약 소프트웨어 안전계획이 없다면 안전현안들이 적절히 고려되지 않거나, 또는 해결되지 않을 수 있다. 왜냐하면 어떤 문제점을 놓고 사람마다 서로 다른 방식으로 해결할 가능성이 크며, 경우에 따라 무시해 버릴 수 있기 때문이다. 따라서 안전관련 소프트웨어 개발사업에는 소프트웨어 안전계획을 세워야 한다. 안전계획서는 분석수행 조직, 안전성 분석책임, 소프트웨어 안전성 활동관리, 그리고 위험요소와 비정상조건이나 사건 등을 다루기 위해 각 단계마다 수행되어야 할 분석 사항들을 반영하여야 한다. 안전계획의 규모, 범위 및 내용은 소프트웨어의 규모, 복잡성과 그 시스템이 고장날 때 생길 수 있는 리스크(risk) 수준에 대비하여 적절히 작성되고, 형상관리체계 하에서 관리되어야 한다.

2. 소프트웨어 요구명세 안전성 분석 기술

소프트웨어 요구사항 안전성 분석 기술에는 본 논문에서 제시한 Checklist **와** Hazard and Operability (HAZOP)**에 의한 방법 이외에도** Failure Mode Effect Analysis (FMEA), Fault Tree Analysis (FTA) **기법이 사용될 수 있다.** FTA **는** 1960 **년대부터 시스템 신뢰도 평가를 위해 사용되어 왔으며 현재는 표준화된 방법으로 정착되어 안전성과 신뢰도 공학분**

야에서 가장 실용적인 방법으로 평가 받고 있다. 그러나 소프트웨어 안전성 분석을 위한 FTA 기법의 적용은 최근에 와서야 소프트웨어 코드 단계에서 적용을 시도하고 있다.

소프트웨어 안전성 검증에 대한 연구는 주로 시스템 안전성 분석에서 사용되던 개념을 소프트웨어 개발 단계에 확대 적용하거나 다른 소프트웨어 안전성 분석 기법인 소프트웨어 HAZOP, 소프트웨어 FMEA 등과 FTA 를 통합하여 안전성 검증 효과를 높여 보고자 하는 연구들이 시도되고 있다. 그 동안 시스템 혹은 하드웨어에 적용되는 FTA 는 확률론적 접근이며 시스템이나 하드웨어의 구조적 분해에 의해 fault tree 를 작성하였다. 그러나 소프트웨어는 논리적이고 결정론적이기 때문에 확률론적으로 접근하는 것은 부적절하다. 소프트웨어는 입출력과 이에 따른 상태변화로 표현되며 구조적 인과관계를 갖는 것이 아니라 입출력 인과관계와 상태 인과관계를 갖기 때문에, 이에 근거하여 fault tree 를 작성하여야 한다.

소프트웨어 요구분석 단계에서의 FTA 를 사용한 안전성 분석 노력은 시스템 안전성 분석과 별 차이 없는 결과만을 생산할 것이라는 생각으로 인해 연구가 미미한 실정이었다. 그러나 안전성 분석을 해야 하는 원자력 발전소 제어 시스템, 미사일, 인공위성 등 실시간 특성을 갖는 안전계통의 개발 필요성은 점차 늘어나고 있다. 이러한 시스템의 소프트웨어를 개발하고자 할 때는 요구분석 단계에서 안전성 분석이 매우 중요하며 관련 인허가 요건에서도 의무사항으로 요구하고 있다.

3. 소프트웨어 요구사항 위해도 분석 절차

소프트웨어 요구사항 위해도 분석은 소프트웨어 요구사항 명세로 인한 계통 위해 영향을 조사하는 것이다. 일반적으로 소프트웨어 요구사항은 소프트웨어의 어떤 측면을 표현하는 다양한 그룹의 요구사항으로 구성된다. 이러한 그룹들은 소프트웨어 위해도 분석 동안 고려할 다음과 같은 기능 특성들이다: 정확도, 성능, 기능성, 신뢰성, 강인성, 안전성, 보안성 등에 대한 요구사항들이며 경우에 따라 조금씩 수정 적용되어야 한다. 즉, 요구사항의 종류에 따라 분석해야 할 안전관련 기능 특성 종류의 집합도 달라야 하며, HAZOP 시 사용할 질문의 깊이도 적절히 조정하여 사용하여야 한다.

소프트웨어 요구사항 위해도 분석의 일반적인 의도는 이러한 기능 특성과 그 특성관련 요구사항들을 조사하여 위험에 미치는 영향을 평가하는 것이다. 이외에도 소프트웨어 요구사항명세와 관련된 공정 특성들이 있다. 예를 들어, 완전성, 일치성, 정확성, 추적성, 명확성, 확인가능성 등이 있으며 이들은 요구사항 위해도 분석에서 보다 요구분석

또는 요구사항 검증에서 다루어져야 한다. 물론 성공적인 소프트웨어 요구사항 위해도 분석을 위해서는 이러한 특성들이 미리 만족되어야 한다. 따라서 이러한 공정 특성에 대해서는 HAZOP 을 이용하여 요구사항 상세 안전성 분석을 수행하기 전에 Checklist 수준에서 안전성 검토를 먼저 실시한다.

표 1. 소프트웨어 공정특성 안전성 검토 체크리스트

공정 특성	Checklist	위험 영향	설계 권고	비고
완전성	소프트웨어 위해도 목록이 존재하는가?			
완전성	각 필수-안전 요구사항을 위한 제어 자동화 정도가 결정되어 있는가?			
완전성	각각의 안전관련 요구사항이 각 요구사항에 안전 중요도를 부과하기 위한 자동화 정도와 계통 위해도에 대해 배치 되었는가?			
완전성	각각의 소프트웨어 요구사항이나 전체 소프트웨어 요구사항이 계통 안전 목표나 요구사항에 부합하는가?			
완전성	부합하지 않는다면 그러한 부적합 사항 해소를 위해 완화 기준이 파악되었는가?			
완전성	안전성 분석자가 안전관련 소프트웨어의 요구사항이 완전한지를 보장했는가?			
일관성	안전성 분석자는 안전관련 소프트웨어 요구사항이 계통 설계명세와 계통 안전성 분석보고서 내용과 일치하는지를 확인하였는가?			
일관성	안전성 분석자는 안전관련 소프트웨어 요구사항이 내부적으로 일관성을 유지하는지를 보장하였는가?			
정확성	안전성 분석자는 안전관련 소프트웨어 요구사항이 정확한지를 확인하였는가?			
정확성	안전성 분석자는 안전관련 소프트웨어 요구사항이 애매하지 않음을 확인하였는가?			
정확성	안전성 분석자는 안전관련 소프트웨어 요구사항의 정밀도를 확인하였는가?			

3.1 소프트웨어 요구사항 위해도 분석 입력

소프트웨어 요구사항 위해도 분석을 위해서는 다음과 같은 정보가 주어져야 한다.

- 예비 위해도 목록
- 예비 위해도 분석
- 안전성 분석 보고서

- 안전계통 설계명세
- 소프트웨어 요구사항 명세서

3.2 소프트웨어 요구사항 위해도 분석 결과

소프트웨어 요구사항 위해도 분석의 출력물은 다음과 같다.

- 소프트웨어 요구사항 위해도 목록
- 소프트웨어에 의해 영향을 받을 수 있는 각 위해도의 안전 중요도
- 각 소프트웨어 요구사항의 안전 중요도
- 소프트웨어가 각 요구사항에 대해서 정확 또는 부정확하게 작동할 때의 위해도 영향분석결과

3.3 소프트웨어 요구사항 위해도 분석 절차

소프트웨어 요구사항 위해도 분석을 위해 다음과 같은 단계가 사용될 수 있다. 각 단계는 분석 공정을 구성에 도움을 주기 위한 수단이며 실제로는 각 단계간 시간적 중첩도 허용된다.

1. 어떤 형태로던 소프트웨어가 책임져야 하는 위험요소를 파악하라. 이 때 각 위험요소의 발생빈도와 결과의 강도를 나타내는 위험 정도(risk)를 포함한다.
2. 각 위험요소와 제어범주와 관련한 소프트웨어 중요도를 파악한다.
3. 각 요구사항의 중요도를 정하기 위해 소프트웨어 요구사항 명세서 (SRS)에서 필수 안전 요구사항들을 계통 위험요소 및 위험범주와 배치시킨다.
4. 필수 안전 소프트웨어 대해서만 특정 지침을 활용하여 각 요구사항을 분석한다.
5. 요구사항 공정특성에 대한 Checklist 수준의 종합적 안전성 검토를 실시한다.
6. 그림 3 과 같은 절차에 따라 HAZOP 기법을 이용한 상세 안전성 분석을 실시한다.
7. 분석결과 보고서를 작성한다.
8. 분석결과와 안전성 분석에 참여한 조직과 사용한 자원의 안전성 검토를 시행한다.

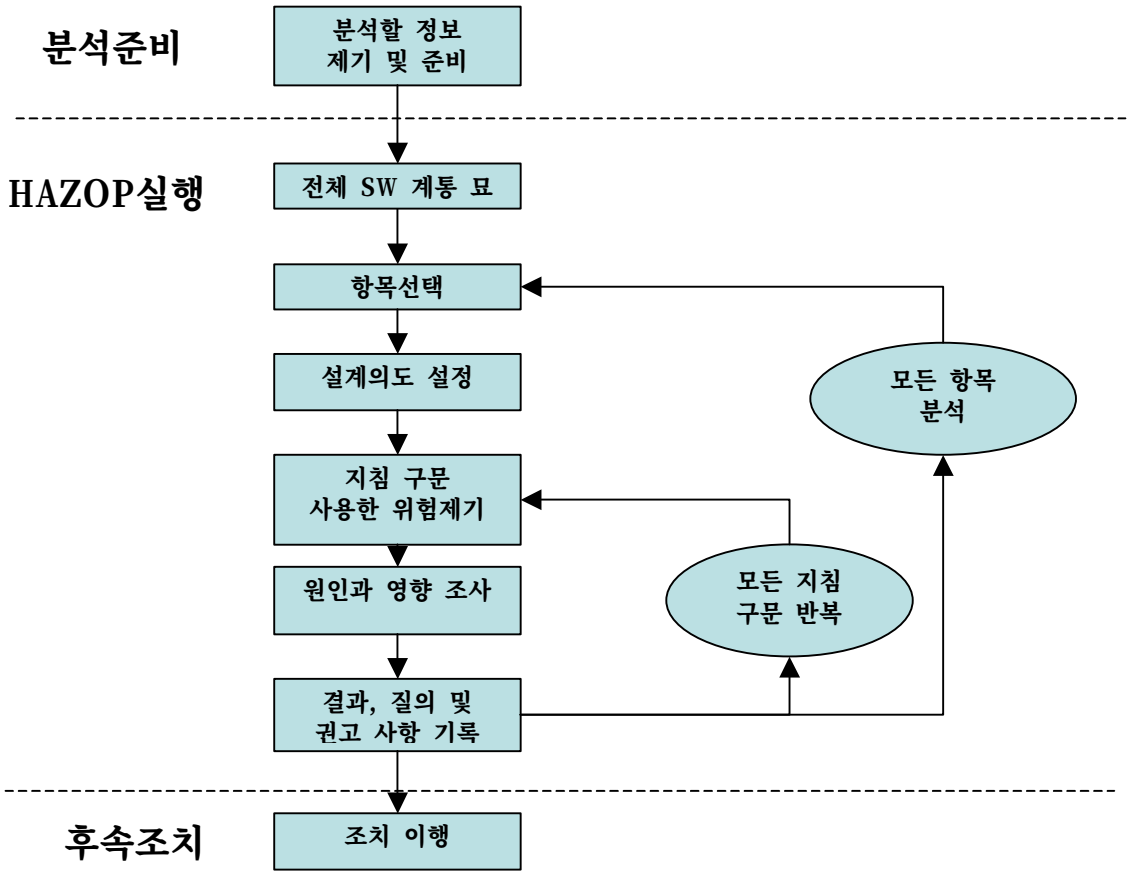

그림 3. 소프트웨어 HAZOP 절차

HAZOP 에 의한 안전성 분석은 사고가 설계 또는 운용상에서 의도한 것에서 벗어났을 때 발생하는 것을 가정하고 설계에서 예상한 운용을 하였을 경우 일어날 수 있는 모든 가능한 일탈(Deviation) 상황과 그와 관련된 모든 위해 요소를 찾으려고 하는 것이다. 이 HAZOP 을 위한 조직은 표 2 와 같이 구성된다.

표 2. HAZOP 조직

HAZOP 분석 책임자
시스템 요건 명세요원
소프트웨어 설계요원
독립 확인 및 검증요원
계통 안전성 분석요원
소프트웨어 안전성 분석요원
기록요원

시스템을 구성하는 각 단위에 대하여 표 3의 Guide Phrase를 사용하여 의도된 동작에서 일탈이 일어 났을 때, 발생 가능한 모든 위해도에 대하여 전문가 팀이 설계에 대한 의문을 제기하는 형식으로 체계적으로 수행된다. 소프트웨어가 가질 수 있는 모든 품질 특성, 즉, 성능, 기능성, 신뢰성, 강인성, 안전성, 보안성 등에 대한 Guide Phrases도 고안하였다. 이러한 고안된 각 Guide Phrases에 대하여 표4와 같은 Checklist와 그 결과표를 개발하였다. HAZOP 조직은 그림 3의 절차에 따라 Hazard 발생의 각 질문에 대한 원인, 공통원인을 기록하고, 이에 따라 초래될 수 있는 결과의 위험영향, 위험도 등을 파악하고 권고사항을 기록하여야 한다.

표 3. 소프트웨어 HAZOP 을 위한 Guide Phrases

품질특성	적용 대상	적용단계	Guide Phrases
정확도	센서	RADC	Stuck at all zeroes
		RADC	Stuck at all ones
		RADC	Stuck elsewhere
		RADC	Below minimum range
		RADC	Above maximum range
		RADC	Within range, but wrong
		RADC	Physical units are incorrect
		RADC	Wrong data type or data size
	액츄에이터	RADC	Stuck at all zeroes
		RADC	Stuck at all ones
		RADC	Stuck elsewhere
		RADC	Below minimum range
		RADC	Above maximum range
		RADC	Within range, but wrong
		RADC	Physical units are incorrect
		RADC	Wrong data type or data size
	운전원 입출력	RA	Numerical value below acceptable range
		RA	Numerical value above acceptable range
		RA	Numerical value within range, but wrong
		RA	Numerical value has wrong physical units
		RA	Numerical value has wrong data type or data size
		RA	Non-numerical value incorrect
	논리연산	RDC	Calculated result is outside acceptable error bounds (too low)
		RDC	Calculated result is outside acceptable error bounds (too high)
		RDC	Formula or equation is wrong
		RDC	Physical units are incorrect
		RDC	Wrong data type or data size

R: Requirements, A: Architectual Design, D: Detail Design, C: Coding

표 4. HAZOP 결과표

품질특성	Guide Phrases	Deviation Checklist	원인	공통원인	위험영향	위험도	권고사항	비고
기능성	Function is not carried out as specified (for each mode of operation)	특정 기능이 각 운전 모드에 대해 명세 된 대로 수행하지 못할 때 어떤 위험을 초래하는가?						
	Function is not initialized properly before being executed	특정 기능이 각 운전 모드에 대해 명세 된 대로 수행하기 전에 적절히 초기화 되지 못할 때 어떤 위험을 초래하는가?						
	Function executes when trigger conditions are not satisfied	시작조건이 만족 되지 않았는데 특정 기능이 수행되면 어떤 위험을 초래하는가?						
	Trigger conditions are satisfied but function fails to execute	또는 반대로, 시작조건은 만족하였지만 특정 기능을 수행하지 못하면 어떤 위험을 초래하는가?						
	Function continues to execute after termination conditions are satisfied	종료조건이 만족된 후에도 특정 기능이 계속 수행되면 어떤 위험을 초래하는가?						
	Termination conditions are not satisfied but function terminates	종료조건이 만족되기 전에 특정 기능이 종료되면 어떤 위험을 초래하는가?						
	Function terminates before necessary actions, calculations, events, etc. are completed	필요한 행동, 계산, 사건들을 완수하기 전에 특정 기능을 종료하면 어떤 위험을 초래하는가?						
	Function is executed in incorrect operating mode	부정확한 운전모드에서 기능이 수행되면 어떤 위험을 초래하는가?						
	Function uses incorrect inputs	특정 기능이 부정확한 입력신호를 사용하면 어떤 위험을 초래하는가?						
	Function produces incorrect outputs	특정기능이 부정확한 출력을 생산하면 어떤 위험을 초래하는가?						

품질특성	Guide Phrases	Deviation Checklist	원인	공통원인	위험영향	위험도	설계권고사항	비고
강인성	Software fails in the presence of unexpected input data	예상치 않은 입력신호에 대해 소프트웨어가 원래 기능을 상실하게 되면 어떤 위험을 초래하는가?						
	Software fails in the presence of incorrect input data	부정확한 입력신호에 대해 소프트웨어가 원래 기능을 상실하게 되면 어떤 위험을 초래하는가?						
	Software fails when anomalous conditions occur	특이조건 발생으로 소프트웨어가 원래 기능을 상실하게 되면 어떤 위험을 초래하는가?						
	Software fails to recover itself when required	소프트웨어가 요구되는 자체 복구를 실패할 때 어떤 위험을 초래하는가?						

4. 결론

소프트웨어 개발 공정 단계 중 요구사항 명세 단계에서 안전성 분석을 수행하기 위한 HAZOP 방안을 제시하였다.. 이 HAZOP 개념을 소프트웨어 요구사항 명세의 안전성 분석에 활용할 수 있도록 하기 위해 지침구문과 체크리스트를 개발하고 이를 이용한 소프트웨어 HAZOP 절차를 개발하였다. 이 HAZOP 분석 동안 수집된 정보는 다음 소프트웨어 개발 단계에서 사용될 수 있다. 다양한 소프트웨어 요구사항에 부과된 위험도는 추후 개발, 검증, 테스트 등에 투여할 자원을 결정하는데 필요한 정보가 된다. 또한 그러한 분석결과는 소프트웨어로 인한 위험을 줄이기 위한 소프트웨어 응용시스템의 재설계 필요성을 제기할 수도 있다. 또한 소프트웨어 요구사항 위해도 분석결과는 계통 설계의 일부 변경을 유도할 수도 있다. 예를 들어 소프트웨어에 부과된 일부 요구사항은 하드웨어로 구현함이 더 적절할 수도 있다.

소프트웨어 요구사항 위해도 분석을 통해 어떤 요구사항은 위험을 내포하고 있지 않다는 결론이 도출될 수 있으며, 이 경우 생명주기 공정 상 계속된 분석이 필요 없게 된다. 현재 개발되고 있는 KNICS 디지털 원전 안전계통 소프트웨어의 요구명세의 안전성 분석에 이 HAZOP 방법을 적용하고자 한다.

알림

본 논문은 과학기술부가 지원하는 원전계측제어시스템 개발사업단의 디지털 계측제어 인허가 확보기술 개발과제 연구 결과입니다.

참고문헌

[1] NUREG/CR-6101, "Software Reliability and Safety in Nuclear Reactor Protection Systems," Lawrence Livermore National Laboratory, November 1993.

[2] IEEE Std. 1228, "Standard for Software Safety Plan," Institute of Electronic and Electrical Engineers, 1994.

[3] NUREG/CR-6430, "Software Safety Hazard Analysis," Lawrence Livermore National Laboratory, February 1996.

[4] 이장수, "공학적안전설비-기기제어계통 소프트웨어 요구사항 명세 안전성 분석 절차서," KNICS-ESF-SSP121, 2003.

[5] McDermid, J. A & Pumfrey, D. J., "A Development of Hazard Analysis To Aid Software Design," COMPASS, 1994.