'2000 춘계학술발표회 논문집
한국원자력학회

SMART MMIS 설계에 적용을 위한 소프트웨어 개발 개념

Software Development Concept for SMART MMIS Design

서용석, 박근옥, 이종복, 김동훈, 허섭
한국원자력연구소

요 약

디지틀 중심의 SMART MMIS 설계 개념을 기반으로 고품질의 디지틀 SMART MMIS를 개발하기 위해 소프트웨어 개발 개념이 고려되어야 한다. 본 논문은 소프트웨어 공통유형고장에 대한 원전 규제기관의 입장, 소프트웨어 안전등급, 소프트웨어 개발에 적용될 코드 및 표준, 소프트웨어 수명주기, 소프트웨어 개발에 적용될 주요 기술에 대해 논의한다.

Abstract

Based on the design concept of SMART MMIS which is developed with fully digitalized system, software development concept should be considered to achieve high quality of digitalized SMART MMIS. In this paper, nuclear regulatory position on software common mode failure, software safety class, code and standards for software development, software life cycle and major techniques for software development are discussed.

1. 서론

전력생산 외에도 담수화 목적을 위해 설계되는 330MWt급의 일체형원자로인 SMART(System-integrated Modular Advanced Reactor)를 제어하고 감시하는 MMIS(Man-Machine Interface System)는 입력신호를 취득하는 계측계통부터 제어 및 보호, 감시, 제어실에 경보 및 정보를 표시하는 모든 계통이 컴퓨터기반의 디지틀시스템으로 설계된다[1]. 고품질의 SMART MMIS를 설계하기 위해 소프트웨어 품질보증계획과 계통별 개발 지침이 작성되어야 하며 이에 앞서 전반적인 소프트웨어 개발 개념이 설정될 필요가 있다. 이를 위해 본 논문은 원전 계측제어계통(I&C)에 대한 규제입장을 분석하고, SMART MMIS의 소프트웨어 등급분류, 적용하고자 하는 소프트웨어 개발 코드 및 표준체계, 소프트웨어개발 수명주기, 고품질의 소프트웨어를 개발하기 위해 적용할 주요 기법에 대해 기술한다.

2. 규제입장 분석

미국 원자력규제기관(NRC)에서 1997년 개정한 안전성검토계획서(NUREG-0800)의 7장의 주된 요지는 원자력발전소 계측제어계통(I&C)에 컴퓨터기반의 디지틀시스템을 적용하였을 때 어떠한 관점을 두고 안전성을 검토할 것인가에 대한 내용이며 주요 검토사항으로써 디지틀 I&C 시스템의 품질등급, 디지틀 공통유형고장 대처, 잘 정립된 소프트웨어 개발공정, on-line 시험, 실시간 성능과 같은 디지틀 I&C 시스템 고유의 특성 등을 들 수 있다. 이 가운데 아나로그 중심의 I&C에서 디지틀 중심의 I&C로 옮겨지면서 특히 강조되고 있는 부분이 디지틀시스템이 잠재적으로 품고 있는 공통유형고장에 대한 대처방안이다. 개정된 안전성검토계획서의 7.7절에서 제어계통의 승인기준(acceptance criteria) 중의 하나로써 SECY 93-087 II.Q, "Defense Against Common-Mode Failures in Digital Instrumentation and Control System"의 만족여부를 확인하라는 요건을 통해서 공통유형고장 대처의 중요성을 이해할 수 있다. 디지틀시스템의 공통유형고장은 하드웨어뿐만 아니라 소프트웨어에 대해서도 규명되고 대처방안이 강구되어야 한다. 이에 대해 한국안전기술원에서 발간한 “디지털 계측제어계통의 심층방어 및 다양성 규제기술 개발“ 보고서[2]를 통해 디지틀시스템의 고통유형고장에 대한 분류와 컴퓨터기반 시스템의 고장유형과 특히 데이터통신시스템의 고장유형을 참조할 수 있다. 또한 이 보고서는 방어수단으로써 품질보증, 격리 및 다양성을 제시하고 있다. 디지틀시스템의 고통유형고장에서 소프트웨어의 공통유형고장 분류를 간과할 수 없으며 본 논문에서는 소프트웨어 고유의 관점에서 공통유형고장을 다음과 같이 분류하였다: 1)하드웨어 고장, 2)시스템 커널과 드라이버 등을 포함한 펌웨어의 오류, 3)프로그래밍 과정에서 시스템자원의 잘못된 사용으로 인한 오류, 4)프로그래밍 논리 자체의 오류 및 구현상의 오류.

한국안전기술원은 원자력발전소 계측제어계통의 방어계층을 제어계통, ATWS 완화설비, 원자로정지계통, ESF 작동계통, 다양성계통, 행정조치 순서로 정의하고 있다[2]. 여기서 주목할 것은 제어계통이 첫 번째 방어계층으로써 기능을 수행한다는 것이다. 또한 한국안전기술원은 1E 등급의 보호계통에서 가끔씩 발생하는 공통유형고장을 보상하는데 제어계통이 사용될 수 있음을 언급하고 있다. 즉, 비안전계통인 제어계통도 충분한 품질이 보장된다면 다양성을 가진 수단(diverse or different means)으로 인정될 수 있음을 의미한다. 이것을 통해 디지틀 제어계통을 아나로그 제어계통에서 보다 고품질의 Non-1E 설비로 설계할 것을 요구하고 있음을 알 수 있다. 현재, 디지틀시스템의 품질보증과 관련하여 하드웨어 관점에서는 어느 정도 경험자료나 고장율 데이터를 얻을 수 있으므로 이를 통해 하드웨어 신뢰도는 제시될 수 있을 것이다. 그러나 소프트웨어의 품질보증은 아직도 논의대상이며 지침이나 표준이 개발 또는 개정 중에 있다. 본 논문을 통해 SMART MMIS 설계에 적용될 수 있는 소프트웨어 품질보증 특히 높은 신뢰성을 갖는 소프트웨어 개발을 위한 개념에 대해 기술한다.

3. 소프트웨어등급 분류

한국안전기술원의 원자력발전소 컴퓨터-기반 계측제어계통에 대한 안전규제 일반원칙 및 규제지침 초안[3]에서 소프트웨어 안전등급을 안전성-필수 소프트웨어, 안전성-관련 소프트웨어, 비안전관련 소프

트웨어로 분류하고 있다. 이것은 기존의 안전관련과 비안전관련만으로 분류하던 방식과는 차이점이 있으며 프랑스의 N4 발전소에 적용되었던 1E와 2E 분류개념과 유사하다고 할 수 있다. 한국안전기술원의 이러한 등급분류는 제어계통의 품질을 강화하려는 취지로 해석할 수 있다. 이는 디지틀시스템의 유연성이라는 장점을 인정하는 대신 공통유형고장에 대해 취약하다는 단점을 보완하여야 하다는 의미를 내포하고 있다. 즉, 발전소 안전정지 상태를 달성하고 유지하는 기능을 수행하는 일부 제어계통을 안전관련 계측제어계통으로 분류함으로써 제어계통이 원자로 보호계통의 기능을 상당히 보상해줄 것으로 판단하고 있는 것이다. 이와 관련하여 SMART MMIS에서는 제어계통의 오동작 또는 고장이 발전소 안전에 어떠한 영향을 주는 지에 대한 안전해석이 아직은 정확히 수행되지 않고 있으나 컴퓨터기반의 디지틀 보호 및 제어계통에 대한 안전해석이 수행될 예정이다.

SMART MMIS 설계에 적용할 소프트웨어 등급을 안전성-필수, 안전성-관련, 비안전관련으로 분류하며 각 등급별로 요구되는 확인 및 검증방법은 한국안전기술원의 원자력발전소 컴퓨터-기반 계측제어계통에 대한 안전규제 일반원칙 및 규제지침 초안의 4장 표1을 따른다. 이 등급분류는 개발 소프트웨어 및 상용 소프트웨어(COTS)에 모두 적용한다.

4. 적용 코드체계

한국원자력연구소에서 원전 계측제어 고신뢰도 소프트웨어에 대한 확인 및 검증 기술을 개발하기 위해 관련 표준 및 코드를 지속적으로 추적하고 적용성 여부를 연구해 왔다[4][5]. 소프트웨어 관련 국제표준은 계속 개정되고 있으며 특히 표1에서와 같이 90년대에 들어와서 많은 코드들이 개정 또는 새로이 개발되고 있다. 원전 계측제어 안전계통의 소프트웨어 개발에 적용될 지침으로써 원전 규제기관에서 제시하고 있는 규제지침(Regulatory Guide)을 준용하는 것이 일반적이다. 특히, 소프트웨어 개발과 관련된 원전 규제지침을 검토해보면 많은 부분에서 IEEE 산업표준 사용을 인준하고 있는데 이것은 산업표준의 신뢰성이 상당히 향상되고 있음을 알 수 있다. 예를 들어, 미국 국방성(DoD)에서 그 동안 군용 소프트웨어 개발에 적용해왔던 MIL Std 498을 중단하고 대신에 IEEE/EIA 12207을 적용할 것을 권고하고 있다[6].

SMART MMIS 안전성-필수 소프트웨어에 대한 품질보증요건은 Appendix B to 10 CFR 50, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants"부터 시작된다. 이 요건을 만족하기 위해 참조할 수 있는 지침서로써 RG-1.152, "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants"와 ASME/NQA-1,2, "Quality Assurance Requirements for Nuclear Facilities"를 들 수 있다. 이 두 지침서는 IEEE Std 7-4.3.2-1993, "Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations"를 인준하고 있으며 이 표준은 컴퓨터기반의 디지틀시스템을 안전계통에 적용하였을 경우 하드웨어, 소프트웨어, 인터페이스 등을 포함한 디지틀시스템의 포괄적인 검증요건을 기술하고 있다. 따라서 SMART MMIS 컴퓨터기반의 안전계통에 포함되는 안전-필수 소프트웨어에 대한 기본적인 품질보증요건은 IEEE Std 7-4.3.2-1993이

다. 세부 지침서로써 표1에 나타난 바와 같이 RG-1.168부터 RG-1.173을 적용한다.

안전성-관련 소프트웨어의 품질보증요건으로써 IEEE Std 7-4.3.2-1993의 적용은 요구되지 않으나 소프트웨어의 안전성 및 신뢰성을 보이려는 노력을 최대한 수행하며 IEEE 표준을 선택적으로 적용한다.

비안전관련 소프트웨어에 적용하여야 할 규제지침은 상황에 따라 다르다. 예를 들어, 안전변수감시계통(SPDS)에 대해서는 NSAC-39에 의거 확인 및 검증을 수행해야 한다. 고품질의 비안전관련 소프트웨어를 개발하는데 적용하기 위한 품질보증계획, 절차 및 지침을 자체적으로 만들어야 하며 IEEE 표준을 선택적으로 참조할 수 있다. 비안전계통의 소프트웨어 규모는 크고 복잡하여 소프트웨어개발지원도구를 사용하여 프로그래밍 하게 되며 다중처리와 그래픽처리가 지원되는 운영체제를 이용하므로 품질보증계획에는 주로 상용소프트웨어 선정기준, 소프트웨어 형상관리, 소프트웨어 시험이 강조될 것이다.

국내 산업코드체계는 아직 개발단계에 있으므로 적용하기에는 이르다. ASME에 해당하는 국내표준으로써 대한전기협회의 전력산업기술기준(KEPIC) 가운데 원자력품질보증(QAP)이 있으며 점차 국내에서도 ASME에 상응하는 KEPIC을 적용하려는 움직임이 있다. 또한 국내의 KSPICE는 소프트웨어 개발 프로세스를 심사하는 국내기관으로써 ISO 9002, ISO 12207, CMM, Trillium 등의 국제적인 산업체의 소프트웨어 개발 프로세스를 수용하여 국제적인 심사표준을 제정하는 SPICE(Software Process Improvement and Capability dEtermination) 한국지부이다. 한국안전기술원은 원전 컴퓨터기반 계측제어계통에 대한 안전규제 일반원칙 및 규제지침을 개발하고 있다. 이러한 움직임은 안전성이 강요되는 산업전반에 디지틀시스템 사용이 점차 증가되고 있으며 디지틀시스템의 신뢰성이 강조되고 있다.

5. 소프트웨어개발 수명주기

소프트웨어 개발은 개발자의 유연성으로 인해 일관되지 않거나 누락된 개발업무가 발생될 특수성이 내재되어 있다. 따라서 소프트웨어의 고품질을 보장하기 위해서는 소프트웨어 개발 수명주기가 잘 정립되어 있어야 한다. 일반적으로 소프트웨어 개발 수명주기는 계획, 요건분석, 설계, 구현, 시험, 통합, 확인 및 검증, 설치, 운전 및 유지보수 단계로 구분된다. 원전 소프트웨어 개발 수명주기는 개정된 안전성검토계획서 7장과 BTP HICB-14절에 상세히 기술되어 있으며 생명주기에 따른 수행업무와 생산물이 체계적으로 명시되어 있으므로 SMART MMIS의 모든 소프트웨어개발 수명주기는 이를 따른다. 소프트웨어 개발 모델로써 IEEE/EIA 12207.2를 참조하여 폭포수(waterfall), 점진형(incremental), 진화형(evolutionary), 재공학(reengineering) 모델을 고려할 수 있으며 rapid prototyping 모델도 고려할 수 있다. SMART MMIS 설계 전반적인 흐름은 폭포수 모델을 따른다. 그러나 SMART MMIS 설계가 디지틀기반으로 이루어지기 때문에 디지틀 특성에 따른 기능 및 성능을 rapid prototyping을 통하여 확인할 예정이다. 또한 확인과정에서 요건의 수정 또는 개선이 발생할 수 있으므로 어느 정도 점진형의 성격도 있다고 할 수 있다. 따라서 SMART MMIS는 폭포수모델을 근간으로 하되 rapid prototyping과 점진형이 혼합된 hybrid 모델 적용을 고려한다.

6. 소프트웨어개발 기법

본 절에서는 원전 규제에 부합되고 고품질의 SMART MMIS 소프트웨어를 개발하기 위해 적용하여야 할 중요 소프트웨어개발 기법을 정형화 기법, 위해도분석 기법, 시험 기법으로 분류하여 기술한다.

6.1 정형화 기법

소프트웨어 사양서의 내용이 일반 서술식으로 되어 있는 경우, 요건작성자와 소프트웨어개발자간의 일치된 요건해석과 의사전달이 명확하지 않을 수 있으며 양측의 오류를 발견하기가 어렵다. 따라서 요건을 의사어(pseudo code) 또는 특정 알고리즘 형식 등으로 소프트웨어 사양서를 작성하게 되는데 이와 같은 작성방법 중의 하나가 정형화 기법(formal methods)이다. 정형화 기법은 자연어 형태의 요건서 내용을 참과 거짓이 명확히 증명될 수 있는 수학적 표현방법이다. 정형화 기법 사용의 가장 근본적인 목적은 요건서의 논리적 결함을 조기에 발견하여 요건을 정확하게 해석하고, 표현하고, 구현하게 하고자 함이다. 심지어는 시험을 통해서도 발견할 수 없는 결함을 정형화를 통해서 발견할 수도 있다. 또한 소프트웨어 개발공정에 정형화 기법을 적용하는 것은 강력한 규범(strong discipline)을 정하기 위함이다. 정형화 기법을 적용함으로써 얻을 수 있는 장점은 다음과 같다: 1)정형화 기법에서 표현되는 변수 또는 서술자들이 전체적으로 일관성 있게 사용되고 있는가를 확인할 수 있으며, 2)논리의 흐름을 전체적으로 파악할 수 있으며, 3) 정형화 기법에서 표현되는 어떤 조건 및 결론이 모든 경우에 대해 논리적으로 증명이 가능한지 확인할 수 있다. CANDU 발전소의 발전소정지계통(SDS) 소프트웨어 설계는 정형과 기법이 비교적 성공적으로 적용된 예를 보여준다[7]. 정형화 기법의 사용에는 세 가지의 문제점을 안고 있다. 첫 번째는 요건이 복잡하면 정형화 기법의 적용에 한계가 있다. 두 번째는 정형화 기법의 효과에 대한 의문이다. 아직도 원자력 소프트웨어 규제 전문가들은 정형화 기법이 기존에 잘 사용해왔던 표현 방법(well-disciplined traditional approach)보다 낫다는 것에 대해서는 일치된 의견을 보이고 있지 않다[8]. 세 번째는 정형화 기법을 지원하는 상용 도구가 많지 않다. 많은 도구들이 학계의 연구단계에서 생산된 산출물로써 원전 안전계통에 적용하기 위해서 요구되는 품질보증요건을 만족하는데는 부족한 점이 많다. 표2는 미국 NASA에서 조사한 정형화 기법 도구를 보여준다. SMART MMIS에서는 정형화 기법 적용을 다음과 같이 결정한다: 1)안전-필수 소프트웨어에 대해서는 정형화 기법을 적용하여 소프트웨어 사양서를 작성하며, 2)안전-관련 소프트웨어는 정형화 기법 적용을 권고하며, 3)비안전관련 소프트웨어는 소프트웨어개발지원도구를 사용하므로 정형화 기법을 적용하지 않는다.

6.2 위해도분석 기법

소프트웨어 위해도 분석은 소프트웨어 설계가 비정상 상태를 얼마나 고려하였으며 그에 대한 대책이 강구되어 있는가를 확인하기 위함이다. IEEE Std 7-4.3.2-1993은 컴퓨터기반의 원전 안전계통에 대해서 위해도 분석을 수행할 것을 요구하고 있다. 이 요건은 안전계통의 ACE(Abnormal Conditions and Events)을 규명하고 규명된 ACE에 대해서 위험을 줄일 수 있는 대책을 세울 것을 요구한다. 이에 대한

구체적인 분석방법으로써 고장유형영향분석(Failure Modes and Effects Analysis)과 고장수목분석(Fault Tree Analysis)을 권고하고 있다. 이 두 분석방법은 ANSI/IEEE Std 352-1987, "IEEE Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems"에 비교적 상세히 설명되어 있다. 고장유형영향분석은 시스템에 잠재된 고장요소를 최대한 밝혀내는 작업이다. 주요부품의 고장요소를 나열하고 그 고장으로 인해 기능적으로 어떠한 영향이 발생되는지 규명하고 평가한다. 고장유형영향분석은 테이블 형식으로 작성되며 테이블에는 부품명, 기능, 고장모드, 고장발생원인, 고장영향, 고장발견방법, 고장대책 등이 채워진다. 이 테이블에는 설계초기에 완벽한 데이터를 채워 넣을 수 없으며 설계공정에 따라 그 내용이 채워질 수 있다. 따라서 고장유형영향분석이 설계에 영향을 끼칠 수 있다. 고장수목분석은 고장이라는 요소를 더 이상 하부고장이 고려되지 않는 요소까지 수목형태의 그래픽 다이아그램을 작성하는 하향식 분석작업이다. 고장유형영향분석과 고장수목분석은 병렬적으로 작성되거나 고장유형영향분석에서 규명한 고장요소를 토대로 고장수목분석에서 수목을 그려나간다. 이와 같은 개념을 소프트웨어 위해도분석에도 적용한다. 소프트웨어에서 더 이상 세분화 할 수 없는 고장요소를 결정하는 일이 매우 중요하다 또한 주요 모듈명 또는 주요 기능을 수행하는 프로그램이 작동되지 못하였을 경우에 대한 대책을 고려하여야 한다. 프로그램 모듈 내에서는 프로그램 구문별로 template를 만들어 고장수목분석을 작성하는 방법이 제시되고 있다[9]. 이 방법을 이용하여 프로그래밍 언어의 구문을 template 형태로 만들어 프로그램의 고장요소를 분석할 수 있다. 소프트웨어만으로는 컴퓨터 시스템의 위해요소를 충분히 고려할 수 없으므로 하드웨어 자원을 적절히 이용하여야 한다. 예를 들어, 실행시간이 만족되지 못하는 경우, watchdog 타이머를 이용하여 강압적으로 트립시켜야 한다. 매 주기를 갖고 실행하는 경우, 주기 시작점에서 watchdog 타이머를 구동하고 마감시간(deadline) 시간을 설정한다. 그 주기의 마지막 점에서 마감시간 이전에 watchdog 타이머를 원상복구(reset)시켜야 한다. 그렇지 못한 경우 마감시간 초과에 의한 watchdog 타이머 callback 루틴이 실행되어 트립이 발생된다. 이것은 어떤 이유에 의해 소프트웨어 실행시간이 만족되지 못하는 위험요소를 하드웨어 자원을 이용하여 줄이는 대책 가운데 하나이다. SMART MMIS에서는 위해도 분석 적용을 다음과 같이 결정한다: 1) 안전-필수 소프트웨어에 대해서는 위해도 분석을 수행하며, 2)안전-관련 소프트웨어는 위해도 분석을 수행을 권고하며, 3)비안전관련 소프트웨어는 위해도 분석을 수행하지 않는다.

개정된 안전성검토계획서에서 소프트웨어에 대한 신뢰도 문제는 정량적인 분석방법만으로 모든 것을 해결하거나 확신하지 말 것을 경고하고 있다. 소프트웨어 안전성분석은 현재로선 정량적 분석방법보다 정성적 분석방법이 최선으로 알려져 있다.

6.3 시험 기법

소프트웨어 시험은 독립적인 기능을 수행하는 최소 단위 모듈별로 시험하는 단위시험, 기능요건 만족을 확인하는 기능시험, 계통이 완성되어 정상적으로 동작하는지 확인하는 계통시험, 다수의 계통이 존재하는 경우 계통들을 통합하여 전체기능을 확인하는 통합시험으로 구분되어 수행된다. 이러한 시험들은

SMART MMIS 소프트웨어 시험에 기본적으로 적용되며 보다 구체적인 시험기법으로써 NUREG/CR-6421을 참조하여 정적 원시코드 분석, 구조적 시험, 기능적 시험, 통계적 시험, 스트레스 시험 기법을 적용한다[10]. 정적 원시코드 분석은 프로그램을 실행하지 않고 코드검사(code inspection), 직접검토(peer reviews), 예비검토(walkthroughs), 컴파일러를 통한 구문확인 등을 통해 오류를 발견하는 기법이다. 프로그램의 많은 오류가 이 기법을 통해 발견된다. 정적 원시코드 분석을 통해 발견할 수 있는 오류로써 데이터 참조의 오류, 데이터 선언의 오류, 계산 오류, 비교 오류, 제어흐름 오류, 연계 오류, 입출력 오류 등을 들 수 있다. 구조적 시험은 프로그램 내부구조를 line-by-line 검사해 나가는 white box 시험을 의미한다. 이 시험은 주로 프로그램 개발자에 의해 수행되는 모듈시험을 의미한다. 구조적 시험의 핵심은 제어흐름 검사와 데이터흐름 검사이다. 제어흐름 검사는 분기(branch), 순환(loop)을 중심으로 모든 실행문이 적어도 한 번은 실행될 수 있음을 확인하는 것이며 데이터흐름 검사는 변수의 생성과 소멸과정에서 모순점이 없는지를 검사하는 것이다. 이 시험의 어려움은 프로그램의 복잡도와 변수의 개수에 따라 모든 경우를 고려한 test case를 어느 정도 만들어야 하는 것이다. 구조적 검사의 효율적인 방법으로써 McCabe에 의해 고안된 Cyclomatic Complexity을 들 수 있다[11]. Cyclomatic Complexity의 목적은 소프트웨어 단위모듈 내에 판단로직의 얼마나 복잡한가를 판단하기 위함이다. 이 복잡도를 기본으로 시험의 정도 등을 판단할 수 있다. Cyclomatic Complexity를 구하기 위해서는 원시소프트웨어의 실행 라인(line)을 노드(node)와 에지(edge)로 구성된 control flow graph 형태로 도식화한 후, (에지-노드+2) 값이 복잡도가 된다. 기능시험은 소프트웨어 사양서의 내용을 확인하는 시험이다. 소프트웨어 사양서는 모듈 단위, 세부 기능 단위, 세부 기능이 결합된 기능 단위로 작성될 수 있으며 이에 따라 시험의 범위도 결정된다. 이 시험은 입력 변수가 가질 수 있는 값의 범위를 구분하여 각각의 범위에서 프로그램의 반응을 시험하는 것이다. 이 시험은 입출력 변수와 기능요건만 있으면 프로그램이 어떻게 구현되었든 상관없이 수행될 수 있는 black box 시험을 의미한다. 이 시험을 통하여 출력값을 확인함으로써 프로그램의 로직이 사양서의 의도와 일치하는지 확인할 수 있다. test case는 입력 변수의 값을 임의로 모사하여 작성한다. 통계시험은 입력은 난수발생기를 통해 이루어진다. 많은 데이터를 장시간 입력하여 고장율을 얻기 위함이다. 사실상 고장율 데이터를 산출하는 데 있어서 test case의 종류와 신뢰성에 대한 문제가 대두된다. 프로그램의 고장율이 10^{-6}이고 이에 대한 99%의 확신을 얻기 위해서 460만 test cases를 수행하여야 하며 하나의 test cases를 수행하는 데 1초가 소요된다면 1.75개월 소요된다[10]. 스트레스 시험은 오류주입 시험(fault injection testing)과 유사하다. 시스템에 과부하를 걸거나 비정상 상태를 발생시켜 시스템의 반응을 시험한다.

7. 결론

고품질의 소프트웨어를 개발하기 위해서는 소프트웨어개발 수명주기의 설정, 잘 정립된 소프트웨어개발 규범, 상세한 소프트웨어 품질보증계획 및 절차, 소프트웨어 안전성 계획 수립 등이 강조되고 있다. 소프트웨어의 신뢰도는 정량적인 방법만으로는 규명될 수 없으며 명확한 확인 및 검증 절차를 통해 개

발과정에서의 오류를 최대한 줄이는 정성적인 노력이 반드시 병행되어야 한다. 본 논문은 SMART MMIS를 디지틀화 함에 따라 요구되는 소프트웨어 개발 개념의 정립 필요성에 의해 원전 디지틀 계측제어에 대한 규제입장 분석, 소프트웨어 등급 분류, 적용할 소프트웨어 코드 및 표준 체계 파악, 소프트웨어 수명주기, 고신뢰도의 소프트웨어를 개발하기 위해 적용할 기법에 대해서 논의하였다. 본 논문의 소프트웨어개발 개념은 SMART MMIS 설계에 필요한 상용 및 개발 소프트웨어에 기본적으로 적용한다. 본 연구는 전체적인 소프트웨어 개발의 전체적인 윤곽 및 방향을 설정하였으며 추후 세부 주제별 연구가 지속되어 체계화 할 예정이다.

Acknowledgement

본 연구는 과학기술부의 원자력연구개발사업 일환으로 수행되었음.

[참고문헌]

1. KAERI/RR-1901/98, "일체형원자로 MMIS설계기술개발", 한국원자력연구소, 1999. 3.
2. KINS/AR-663, "디지털 계측제어계통의 심층방어 및 다양성 규제기술 개발", 한국원자력안전기술원, 1999. 4.
3. "원자력발전소 컴퓨터-기반 계측제어계통에 대한 안전규제 일반원칙 및 규제지침", 초안 0, 한국원자력안전기술원, 1999. 8. 16.
4. 서용석 외, "SMART MMIS를 위한 상용소프트웨어 선정절차 개발", '99춘계원자력학회 학술발표회, 1999. 5.
5. 이장수 외, "원전 계측제어 고신뢰도 소프트웨어 확인 / 검증 기술 현황", Journal of the Korean Nuclear Society, 26권, 4호, 1994. 12, pp. 600-610.
6. MIL-STD-498, NOTICE 1, "Software Development and Documentation", USDoD, 1998, 3, 27.
7. N. M. Ichiyen et al, "Safety Critical Software Design Approaches Developed for Canadian Nuclear Power Plants", KERNTECHNIK, 1995, pp. 232-237.
8. Four Party Regulatory Consensus Report on the Safety Case for Computer-Based Systems in Nuclear Power Plants, USNRC.
9. Nancy G. Leveson et al, "Safety Verification of ADA Programs Using Software Fault Trees, IEEE Software", July 1991, pp. 48-59.
10. NUREG/CR-6421, "A Proposed Acceptance Process for Commercial Off-the-Shelf(COTS) Software in reactor Applications", USNRC, 1996. 3.
11. NIST Special Publication 500-235, "Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric", 1996, 9.

표 1 원자력 규제지침과 산업체 표준의 관계

번호	제목
RG 1.168	VERIFICATION, VALIDATION, REVIEWS, AND AUDITS FOR DIGITAL COMPUTER SOFTWARE USED IN SAFETY SYSTEMS OF NUCLEAR POWER PLANTS
RG 1.169	CONFIGURATION MANAGEMENT PLANS FOR DIGITAL COMPUTER SOFTWARE USED IN SAFETY SYSTEMS OF NUCLEAR POWER PLANTS
RG 1.170	SOFTWARE TEST DOCUMENTATION FOR DIGITAL COMPUTER SOFTWARE USED IN SAFETY SYSTEMS OF NUCLEAR POWER PLANTS
RG 1.171	SOFTWARE UNIT TESTING FOR DIGITAL COMPUTER SOFTWARE USED IN SAFETY SYSTEMS OF NUCLEAR POWER PLANTS
RG 1.172	SOFTWARE REQUIREMENTS SPECIFICATIONS FOR DIGITAL COMPUTER SOFTWARE USED IN SAFETY SYSTEMS OF NUCLEAR POWER PLANTS
RG 1.173	DEVELOPING SOFTWARE LIFE CYCLE PROCESSES FOR DIGITAL COMPUTER SOFTWARE USED IN SAFETY SYSTEMS OF NUCLEAR POWER PLANTS

번호	제목
MIL-STD-498	SOFTWARE DEVELOPMENT AND DOCUMENTATION(USE IEEE/EIA 12207)
IEEE/EIA 12207.0	Industry implementation of international standard ISO/IEC 12207:1995 software life cycle processes
IEEE/EIA 12207.1	Industry implementation of international standard ISO/IEC 12207:1995 software life cycle processes-life cycle data
IEEE/EIA 12207.2	Industry implementation of international standard ISO/IEC 12207:1995 software life cycle processes-implementation considerations
ISO/IEC 12207	Information technology-software life cycle processes

인준(Endorse)

참조(Refer)

번호	제목
IEEE STD 610.12-1990	IEEE STANDARD GLOSSARY OF SOFTWARE ENGINEERING TERMINOLOGY
IEEE STD 730-1998	IEEE STANDARD FOR SOFTWARE QUALITY ASSURANCE PLANING
IEEE STD 730.1-1995	IEEE GUIDE FOR SOFTWARE QUALITY ASSURANCE PLANING
IEEE STD 828-1990, 1998	IEEE STANDARD FOR SOFTWARE CONFIGURATION MANAGEMENT PLANS
IEEE STD 829-1983, 1998	IEEE STANDARD FOR SOFTWARE TEST DOCUMENTATION
IEEE STD 830-1993, 1998	IEEE RECOMMENDED PRACTICE FOR SOFTWARE REQUIREMENTS SPECIFICATIONS
IEEE STD 982.1-1988	IEEE STANDARD DICTIONARY OF MEASURES TO PRODUCE RELIABLE SOFTWARE
IEEE STD 982.2-1988	IEEE GUIDE FOR THE USE OF IEEE STANDARD DICTIONARY OF MEASURES TO PRODUCE RELIABLE SOFTWARE
IEEE STD 1008-1987, 1993	ANSI/IEEE STANDARD FOR SOFTWARE UNIT TESTING
IEEE STD 1012-1998	IEEE STANDARD FOR SOFTWARE VERIFICATION AND VALIDATION
IEEE STD 1016-1998	IEEE STANDARD TO SOFTWARE DESIGN DESCRIPTIONS
IEEE STD 1016.1-1993	IEEE GUIDE TO SOFTWARE DESIGN DESCRIPTIONS
IEEE STD 1028-1997	IEEE STANDARD FOR SOFTWARE REVIEWS
IEEE STD 1042-1987, 1993	ANSI/IEEE GUIDE TO SOFTWARE CONFIGURATION MANAGEMENT
IEEE STD 1044-1993	IEEE STANDARD TO CLASSIFICATION FOR SOFTWARE ANOMALIES
IEEE STD 1044.1-1995	IEEE GUIDE CLASSIFICATION FOR SOFTWARE ANOMALIES
IEEE STD 1045-1992	IEEE STANDARD FOR SOFTWARE PRODUCTIVITY METRICS
IEEE STD 1058-1998	IEEE STANDARD FOR SOFTWARE PROJECT MANAGEMENT PLANS
IEEE STD 1059-1993	IEEE GUIDE FOR SOFTWARE VERIFICATION AND VALIDATION PLANS
IEEE STD 1061-1992, 1998	IEEE STANDARD FOR SOFTWARE QUALITY METRICS METHODOLOGY
IEEE STD 1062-1993, 1998	IEEE RECOMMENDED PRACTICE FOR SOFTWARE ACQUISITION
IEEE STD 1063-1987	IEEE STANDARD FOR SOFTWARE USER DOCUMENTATION
IEEE STD 1074-1995, 1997	IEEE STANDARD FOR DEVELOPING SOFTWARE LIFE CYCLE PROCESSES
IEEE STD 1074.1-1995	IEEE GUIDE FOR DEVELOPING SOFTWARE LIFE CYCLE PROCESSES
IEEE STD 1175-1992	IEEE TRIAL-USE STANDARD REFERENCE MODEL FOR COMPUTING SYSTEM TOOL INTERCONNECTIONS
IEEE STD 1209-1992	IEEE RECOMMENDED PRACTICE FOR THE EVALUATION AND SELECTION OF CASE TOOLS
IEEE STD 1219-1998	IEEE STANDARD FOR SOFTWARE MAINTENANCE
IEEE STD 1220-1998	IEEE STANDARD FOR THE APPLICATION AND MANAGEMENT OF THE SYSTEMS ENGINEERING PROCESSES
IEEE STD 1228-1994	IEEE STANDARD FOR SOFTWARE SAFETY PLANS
IEEE STD 1233-1996, 1998	IEEE GUIDE FOR DEVELOPING SYSTEM REQUIREMENTS SPECIFICATIONS
IEEE STD 1298-1992	IEEE STANDARD FOR SOFTWARE QUALITY MANAGEMENT SYSTEM
IEEE STD 1348-1995	IEEE RECOMMENDED PRACTICE FOR THE ADOPTION OF CASE TOOLS
IEEE STD 1420-1995	IEEE STANDARD FOR INFORMATION TECHNOLOGY - SOFTWARE REUSE
IEEE STD 1420.1a-1996	IEEE SUPPLEMENT TO STANDARD FOR INFORMATION TECHNOLOGY -S/W REUSE

표 2 정형화 기법 도구 종류

이름	내용	개발자
Ac12	Boyer-Moore theorem prover를 기반으로 한다.	Texas Austin의 Computational Logic Inc.
Action Semantics	정형적 의미론(semantics)의 프로그래밍 언어를 표현할 수 있도록 한다.	Denmark의 Aarhus 대학
Algebraic Design Language	대수적 개념을 바탕으로 higher-order 소프트웨어 사양서를 작성할 수 있도록 한다.	Oregon Graduate Institute
BDDs(Binary Decision Diagrams)	finite-state problems을 지원한다.	
Boyer-Moore	theorem prover이다.	ICOT Free Software
B-Method	Abstract Machine Notation을 사용한다.	영국의 B-CORE Ltd.
CCS(Calculus of Communicating Systems)	병렬처리시스템을 표현하는 algebra를 지원한다.	
Circal(Circulate Calculus)	algebra를 이용하여 주로 병렬처리 시스템을 표현할 수 있도록 한다.	영국의 Strathclyde 대학
Concurrency Workbench	model checking 기법을 통해 병렬처리 시스템 분석을 지원한다.	Edinburgh 대학
Coq	귀납적(inductive) theorem prover이다.	프랑스의 INRIA
CSP(Communication Sequential Processing)	process algebra이다.	Oxford 대학의 C.A.R. Hoare
DisCo	Temporal Logic of Actions(TLA)에 기반을 둔 의미론 및 증명 방법을 제공한다.	핀란드의 Tampere University of Technology
Estelle	상태전이모델(state transition model)을 작성하도록 한다.	
EVES	ZFC set theory에 기반을 둔다.	캐나다의 ORA
HOL	수학적 theorem 증명기법을 제공한다.	
HyTech	temporal-logic을 만족하는 지 확인할 수 있다.	Cornell 대학
IMPS	전통적인 수학기법을 제공한다.	미국 Massachusetts Bedford의 MITRE Corp.
Isabelle	다양한 로직기법을 기반으로 한 theorem prover이다.	Cambridge 대학
JAPE(Just Another Proof Editor)	논리적 추론을 제공한다	Oxford 대학
KIV(Karlsruhe Interactive Verifier)	stepwise refinement를 기반으로 한다.	
LAMBDA	하드웨어/소프트웨어 co-design을 확인할 수 있다.	영국의 Abstract Hardware Ltd.
Larch and LP(Larch Prover)	First-order 논리를 확인할 수 있다	
LeanTap	전통적인 First-order 논리를 확인할 수 있도록 tableau-based 연역적(deductive) theorem prover이다.	독일의 Karlsrushe 대학
LOTOS(Language of Temporal Ordering Specifications)	정형 명세와 algebra를 처리한다.	네덜란드의 Twente 대학

표 2 정형화 기법 도구 종류(계속)

이름	내용	개발자
Maintainer's Assistant	역공학 및 재공학을 지원한다.	영국의 Durham 대학
Meije tools	병렬처리 확인을 지원한다.	프랑스의 INRIA
NP-Tools	범용 확인 tool-box이다. 원자력에 적용된 경험이 있다.	유럽에서 상용으로 팔리고 있다.
Nqthm	Boyer-Moore를 개선한 theorem prover이다.	Computational Logic Inc.
Nuprl	intuitionistic type theory를 제공한다	Cornell 대학
Otter	4세대 귀납시스템이다.	Illinois의 Argonne National Laboratory
Penelope	Ada 프로그래밍 확인을 지원한다.	NewYork의 Odyssey Research Associates
Petri Nets	그래픽적으로 모델링이 가능하다.	
ProofPower	higher-order 로직 및 Z 확인을 지원한다	영국의 ICL
PVS(Prototype Verification System)	higher-order 로직 확인을 지원한다.	
RAISE	rigorous development methodology를 기반으로 한다.	Denmark의 CRI
Refinement Calculus	stepwise refinement를 기반으로 한다.	필란드의 Abo Akademi 대학
RESOLVE	abstract data type을 기반으로 한 명세언어를 제공한다.	Ohio State 대학
RRL(Rewrite Rule Laboratory)	First-order 논리확인을 지원한다.	
SCR(Software Cost Reduction)	tabular 표기법을 기반으로 요건서를 작성한다.	Naval Research Laboratory.
SDL(Specification and Description Language)	extended state machine formalism을 기반으로 통신시스템을 모델링을 지원한다.	덴마크의 Tele Danmark Research
SDVS(State Delta Verification System)	state deltas/temporal logic을 지원한다.	California의 Aerospace Corp.
SPIN	분산시스템의 프로토콜 및 모델을 확인할 수 있다.	AT&T Bell Labs
STeP(Stanford Temporal Prover)	temporal specification과 model checking을 지원한다.	Stanford 대학
TAM(Temporal Agent Model)	실시간처리를 위한 모델링을 지원한다.	영국의 York 대학
TLA(Temporal Logic of Actions)	병렬처리를 위한 로직과 추론을 지원한다.	독일의 Dortmund 대학
TPS and ETPS(Theorem Proving System and the Educational Theorem Proving System)	First-order 논리확인을 지원한다.	Carnegie Mellon 대학
TTM/RTTL(Timed Transition Systems/Real-time Temporal Logic)	실시간을 확인을 지원한다.	카나다의 York 대학
UNITY	병렬처리 로직을 지원한다.	Austine의 Texas 대학
VDM(Vienna Development Method)	이산(discrete) 수학을 기본으로 한 모델중심의 명세서를 지원한다.	
Z	집합이론과 first-order 논리를 그래픽적으로 지원한다.	ISO 표준으로 추진 중