

MMIS 동적 모의를 위한 우선순위 기반의 소프트웨어 요구사항

박근옥^U 이종복 구인수
한국원자력연구소
{gopark, jbleel, iskoo}@kaeri.re.kr

Software requirements based on priority for MMIS dynamic simulation

Park-Geun Ok^U Lee-Jong Bok In-Soo Koo
Korea Atomic Energy Research Institute

요약

소프트웨어 개발 프로젝트에서 소프트웨어 요구사항의 잦은 변경 또는 요구사항 확정의 지연은 소프트웨어의 품질저하, 납기지연, 개발요구자와 개발담당자 간의 의견충돌을 야기한다. 특히 원자력발전소 환경과 같은 mission-critical 영역의 경우에는 발전소 건설자체를 지연시킬 수 있는 심각한 결과를 초래한다. 본 논문에서는 원자력발전소의 감시와 제어를 담당하는 MMIS(Man Machine Interface System) 환경을 동적으로 모의하는 소프트웨어 개발 과정에서 요구사항 변경의 가능성을 최소화시킬 의도로 시도하였던 요구사항 우선순위 설정에 대한 경험과 교훈을 살펴본다. 또한, 요구사항 우선순위를 기반으로 한 소프트웨어 개발 공정활동을 토의한다.

1. 서론

원자력발전소에서 사용되는 MMIS(Man Machine Interface System) 개발 관련 소프트웨어 요구명세서를 작성하는 경우에 많이 참조하는 표준들로는 IEEE 표준 830과 1233이 있다[1][2]. 다른 산업분야와 마찬가지로 원자력발전소 MMIS 소프트웨어의 최상위 개발 요구사항은 MMIS를 구성하는 각 계통의 체계요건(system requirements)으로부터 나온다. IEEE 표준에 따르면 사전검토 업무수행과 요건분석 업무수행 결과를 바탕으로 요구명세서를 작성하도록 제안하고 있다. 또한 요구명세서는 정확성, 기능성, 신뢰성, 안전성, 완전성, 타이밍, 보안, 일관성, 스타일, 추적성, 명확성이 반영되고 포함되어야 한다고 기술하고 있다. 원자력발전소는 거대하고 복잡한 시스템이므로 이러한 표준의 요건을 완전하게 충족하는 요구명세서를 단기간에 작성하기는 사실상 가능하지 않다. 따라서 실제의 원자력발전소에 설치되기 이전에 MMIS 환경을 동적으로 모의하는 설비(예 : 시뮬레이터, test bed)를 개발하고, 이의 경험을 바탕으로 요구명세서와 설계사양서를 개발하는 전략을 사용하

기도 한다. 본 연구는 MMIS 환경을 동적으로 모의하는 환경 개발을 수년 전부터 수행해오고 있으며, 현재는 실제의 발전소에 설치할 MMIS 소프트웨어의 개발을 병행하여 진행하고 있다[3][4][5]. 본 논문과 관련된 최근의 연구내용은 참고문헌 [6]에 있다. 본 논문에서는 실제의 원자력발전소에 적용할 MMIS 소프트웨어 개발의 사전단계로 수행된 MMIS 동적 모의를 위한 소프트웨어 개발에서 경험한 우선순위 기반의 요구사항 설정에 대하여 토의한다. 또한 설정한 우선순위별 요구사항이 소프트웨어 개발 공정에서 어떠한 관계를 갖고 사용되었는지 살펴본다.

2. MMIS 동적 모의 소프트웨어의 기능 구조

MMIS의 동적 모의에 사용되는 응용 소프트웨어의 일반적인 기능 구조는 그림 1과 같다. 응용 소프트웨어는 시뮬레이션 엔진, 실행통제 소프트웨어, 인간기계연계 소프트웨어로 구성된다. 시뮬레이션 엔진은 원자력발전소의 노심(Reactor Core), 발전소의 일차계통과 이차계통에 대한 열수력학적 특성을 수확방정식으로 표현

하는 기능을 수행한다. 실행통제 소프트웨어는 강사작업만 소프트웨어라고도 부르며 MMIS 모사 환경의 전체적인 실행을 통제하고 관리하며, 서로 다른 응용 소프트웨어간의 공유 데이터(shared data) 전송과 사용권을 통제한다. 인간기계연계 소프트웨어는 운전원이 모의 환경을 통하여 발전소를 감시 및 제어할 수 있도록 그래픽 기반의 인터페이스를 제공한다.

MMIS 모의 환경에서 세 가지의 응용 소프트웨어 중 시뮬레이션 엔진은 그 존재가 사용자(강사, 운전원)에게 드러나지 않는다. 기능측면에서 실제의 발전소가 갖는 신호(운전 변수 값)이며 실행통제 소프트웨어와 인간기계연계 소프트웨어에서 사용될 공유 데이터를 생산하는 원천이므로 하나의 black box로 간주된다.

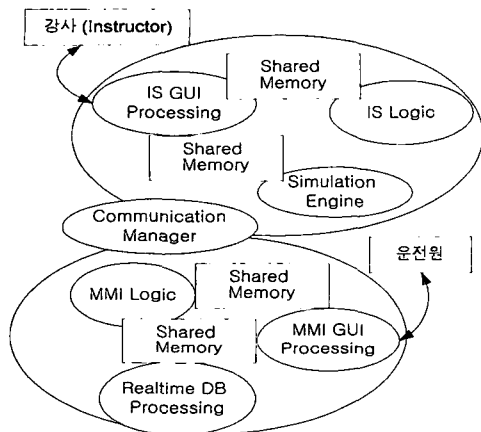


그림 1 응용 소프트웨어의 기능 구조

실행통제 소프트웨어와 인간기계연계 소프트웨어는 각각 GUI 부분과 논리처리 부분으로 분할 할 수 있다. 실행통제 소프트웨어의 GUI 부분은 강사가 시뮬레이션 엔진에 대한 명령(run, freeze, fast, slow, snapshot, backtrack 등)을 사용할 수 있는 그래픽 기반의 입력 인터페이스를 제공함과 동시에 입력 명령에 대한 시뮬레이션 엔진의 처리진행과 처리결과를 표시화면에 출력한다. 실행통제 소프트웨어의 논리처리 부분은 GUI 부분으로부터의 강사 입력 명령을 해독하여 시뮬레이션 엔진에 전달하고 시뮬레이션 엔진의 응답 및 처리결과를 가공하여 GUI 부분에 전달한다. 인간기계연계 소프트웨어의 GUI 부분은 발전소 운전상태 감시정보를 표시화면에 그래픽 기반으로 표시하고 운전원의 제어요구를 소프트웨어가 부르는 인터페이스로부터 입력받아 인간기계연계 소프트웨어의 논리처리 부분에 전달한다. 소프트웨어는 hard-wired 스위치를 그래픽 객체로 형상화한 것이다. 인간기계연계 소프트웨어의 논리처리

부분은 실행통제 소프트웨어와의 공유 데이터 전송을 처리하며 GUI 부분에서 사용할 데이터를 가공 처리한다. 이러한 처리에 실시간 데이터베이스가 사용된다.

3. 소프트웨어 개발 요구사항 우선순위

MMIS의 동적 모사 환경 개발은 보통 2~3년의 기간이 소요되며 2개 이상의 소프트웨어 개발조직 또는 외주개발 회사가 참여하게 된다. 따라서 개발조직 간의 책임과 권한, 연계사항 정의 및 관리가 프로젝트의 성공과 시스템 통합의 중요한 관건이 된다. 따라서 요구사항의 변경, 추가, 삭제는 최악의 경우, 개발조직간의 업무충돌을 발생시킨다. 또한 그림 1의 응용 소프트웨어 기능 구조 관점에서 알 수 있듯이 공유 메모리를 사용한 소프트웨어 모듈간의 연계 요구사항 처리가 높은 중요성을 갖는다. 이러한 특성을 감안하여 본 연구는 요구사항을 4가지 수준으로 구분하여 다음과 같이 정의하였다.

- 1) 우선순위 1은 세 가지로 구분되는 응용 소프트웨어(시뮬레이션 엔진, 실행통제 소프트웨어, 인간기계연계 소프트웨어) 상호간에 준수되어야 할 연계처리요건에 적용한다. 이 수준의 요구사항은 소프트웨어 요구명세서에서 TBD(to be determined)를 허용하지 않으며, 요구사항 확정 후에는 변경을 허용하지 않는다.
- 2) 우선순위 2는 실행통제 소프트웨어와 인간기계연계 소프트웨어의 GUI 부분과 논리처리 부분의 연계처리요건에 적용한다. 우선순위 1과 같이 TBD와 변경을 허용하지 않는다.
- 3) 우선순위 3은 실행통제 소프트웨어와 인간기계연계 소프트웨어의 GUI 부분 및 논리처리 부분의 차체 처리기능과 성능요건에 적용한다. 이 수준의 요구사항은 요구명세서에서 부분적인 TBD를 허용하며, 요구사항 확정 후에는 개발조직간 협의를 통한 변경을 허용한다.
- 4) 우선순위 4는 MMIS 모의 환경의 사용자와 직접적인 시각적 및 촉각적 인터페이스(입력 및 출력)를 갖는 요건에 적용한다. 이 수준의 요구사항 변경은 상용 GUI 개발도구를 활용하여 쉽게 수용 가능하므로 FAT(Final Acceptance Test) 직전 단계까지 TBD를 허용한다. 또한 요구명세서에서 개념적 또는 추상적인 요구사항 표현을 허용한다. 그러나 개발자가 TBD 해결을 요구할 경우에는 요구의 타당성이 평가되어야 하고 대안제시 또는 요구사항 확정이 이루어져야 한다.

4. 개발공정에서 요구사항 우선순위의 역할

그림 2는 MMIS 모의 환경 개발의 공정이다. 우선순

위 기반의 요구사항을 적용하기 위하여 요구명세서 작성 공정이전에 연계요건서(IR : Interface Requirement) 작성이 선행되고 문서의 검토와 승인 활동이 수행되는 특징을 갖는다. IR 문서는 우선순위 1에 해당하는 요구사항으로 구성하였으며 FAT 시험단계에서 반드시 요건의 만족 여부가 시험되도록 공정활동을 정의하였다.

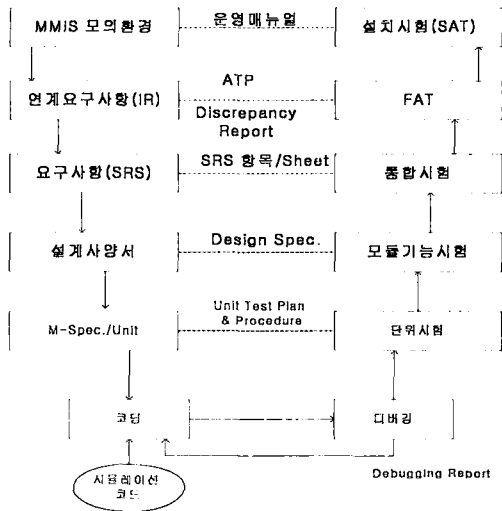


그림 2 소프트웨어 개발공정

GUI 부분과 논리처리 부분간의 연계요건인 우선순위 2에 해당하는 요구사항은 TBD 항목 없이 요구명세서에 기술하였으며, 이 요구사항은 통합시험 단계에서 요건의 만족이 반드시 시험되도록 공정활동을 정의하였다. 우선순위 3에 해당하는 요구사항은 소프트웨어 개발을 담당할 개발자를 직접 참여시켜 요구명세서를 공동으로 작성하는 방식으로 수행하였다. 이 과정에서 개발자가 이해하지 못하거나 개발요구의 요구사항을 분명하게 전달하기 어려운 요구사항은 요구명세서에서 TBD로 두었다. 우선순위 3의 TBD는 MMIS 모의 환경을 축소한 prototype 시스템 개발 통한 확인, 유사한 시스템 개발 사례[4]의 사양 검토 등을 통하여 해결하였고, 이를 설계사양서 작성 단계 이전에 요구명세서에 반영하였다. 우선순위 3 요구사항은 소프트웨어 개발조직이 모듈기능시험 단계에서 만족 여부를 확인한 후, 시험결과보고서를 제출하도록 공정활동을 정의하였다. 우선순위 4에 해당하는 요구사항은 유사 개발사례를 소프트웨어 개발자에게 제시하고 소프트웨어 개발조직이 재량권을 갖고 요구명세서를 작성하도록 하였다. 그러나 표시화면의 영역분할과 각 영역의 기능, 사용자의 명령 입력에 대한 소프트웨어의 반응(feedback)과 같이 중요한 요구사항은 디스플레이 요건서라는 별도의 문서

로 작성하여 개발자가 준수하도록 하였다. 우선순위 4에 해당하는 요구사항은 설계사양서 작성단계 종료이전에 확정되어야 하지만 확정에도 불구하고 모듈기능시험, 통합시험, FAT 과정에서 개발자에 의한 요구불만이 제시되면 상호협의를 통하여 변경 가능한 요구사항으로 정의하였다. 그래픽 객체의 위치, 크기, 형상, 색상 등이 이러한 유형의 요구사항에 해당된다.

5. 결론

소프트웨어 개발 요구사항에 우선순위를 부여하여 요구명세서를 작성하고 이의 충족 여부를 개발공정의 시험단계에서 확인함으로써 다음과 같은 효과를 얻을 수 있었다.

- 1) 서로 다른 개발조직(또는 회사)간의 업무분장 관리와 인터페이스 문제해결에 대한 근거가 분명하다.
- 2) 개발조직이 요구사항의 중요도에 따라 인력과 자원 투입을 실정에 맞게 조정할 수 있다.
- 3) 인력과 시간이 많이 소요되는 시험공정에서 시험해야 할 항목과 시험의 중요도가 자연스럽게 정의된다.

후 기

본 연구는 과학기술부에서 시행하는 원자력연구개발사업으로 수행되었음.

참고문헌

[1] IEEE Std 830-1998, Recommended Practice for Software Requirements Specifications.
 [2] IEEE Std 1233-1998, IEEE Guide for Developing System Requirements Specifications.
 [3] 박재창 외, "Compact Nuclear Simulator 성능향상 기술개발", 한국원자력학회 1999 추계학술발표논문집.
 [4] 심봉식 외, "Development of a full scope Human Machine Simulator for Human Factors Experiments", 한국원자력학회 1997년 춘계학술대회논문집.
 [5] 박근옥 외 "설계해석코드 기반의 원자력발전소 훈련용 시뮬레이터 개발전략", 2000 대한산업공학회/한국경영과학회 2000 춘계공동학술회의 논문집.
 [6] 박근옥 외, "공유메모리 변수 기반의 CNS 응용 소프트웨어 구조, 한국정보과학회 2001 추계학술대회 논문집.