중수로 노심관리 자동화 시스템의 요구사항 추출 사례

A Study on the Elicitation of the Requirements of the CANDU Core Management Automation System

김진일*, 이상훈, 박대유, 염충섭
고등기술연구원 엔지니어링시스템 개발팀

1. 서 론

본 논문은 가압중수로형 발전소인 월성 원자력 제1 발전소의 노심관리 자동화 시스템(COMPAS: CANDU Core Management Automation System) 개발업무를 위해 수행한 요구사항 추출 사례에 대한 연구이다. 요구사항을 추출하기 위한 방법은 여러 가지가 있으나 본 사례에서는 도제법을 근간으로 요구사항을 추출하였다. 본 논문은 COMPAS 요구사항 추출 사례에서 사용한 도제법의 구체적인 수행 내용에 대해서 논의하고 체득한 교훈 및 개선사항에 대해서 기술한다.

2. 본 론

2.1 프로젝트 개요

월성 제1원자력 발전소는 가압중수로형 원자력 발전소이다. 가압경수로형 원자력 발전소는 예방정비계획에 의해 원자력 발전소를 정지시키고 핵연료를 교체하는데 비해 가압중수로형 원자력 발전소는 원자로를 정지시키지 않고 운전 중에 매일 핵연료를 교체한다. 또한 가압중수로의 노심은 380개의 압력관속으로 핵연료를 장전하는 구조로 되어 있고 노심의 안전운전을 위한 노내의 출력제어장치들의 구조가 아주 복잡하게 이루어져 있다. 이러한 노심의 구조 및 운전 특성상 가압중수로에서는 노심관리가 가압경수로에 비해 매우 복잡하고 관리 업무량도 많다. 현재 월성 제1발전소를 비롯한 중수로형 원자력 발전소에서는 노심관리를 위한 업무 절차서가 잘 정리되어 있으나 실제 업무를 수행하는데 있어서 매우 복잡한 계산을 수행해야 하는 경우가 많고, 담당자의 노하우에 따라 업무의 효율성이 달라지기도 한다. 이러한 여러 가지 문제를 해결하고자 중수로형 원자로 노심관리 자동화 시스템을 개발하고자 한 것이 COMPAS 개발 프로젝트이다.

2,2 요구사항 추출 방법 선정

요구사항 추출 방법에 대해서는 여러 가지 방법이 논의된 바 있다.[1] 요구사항의 추출 방법을 선정함에 있어 정확한 정답이 있다고 보기는 어렵지만 주어진 상황과 참여자들의 특성에 따라 효과적인 추출방법을 선정하는 것은 매우 중요한 것이다. COMPAS의 사용자는 업무에 대한 전문가들이고 개발자는 소프트웨어 전문가로 구성되어 있다. 따라서 사용자의 요구사항을 소프트웨어 개발자들이 정확히 인식하고 개발하는 것이 매우 중요하다. 특히 중수로 노심관리 업무를 위한 절차서가 있고 모든 업무는 절차서에 의해 수행되므로 모든 요구사항의 근간은 절차서가 될 수 밖에 없었다. 이에 고기원의 시스템 개발자들은 절차서를 기본으로 한 도제법(apprenticing)을 사용하였다.

2.3 도제법

도제법은 이해당사자/스승(master)이 무엇을 하는지 관찰하고, 질문을 하고, 스승이 하는 것을 숙련될 때 까지 모방하는 도제살이 개념을 사용한다. 요구사항 추출자(시스템엔지니어 또는 도제자)는 이해당사자의 업무 한 부분을 직접 수행한다. 시스템 엔지니어가 업무를 익히고 난 후 요구사항 세트를 개발한다.

이 방법은 이해당사자가 요구사항 추출에 직접 참여할 시간을 낼 수 없는 경우에 유용하며, 이해당사자가 자기가 수행하는 업무의 목적을 표현하기 어려울 때 또는 이해당사자가 전체 시스템 중 일부분에 대한 지식만 있을 경우에 유용하다. 그러나 이 방법은 시스템엔지니어에게 매우 많은 시간투자를 요구하는 방법이다.

2.4 수행 내용

● 절차서의 파악 및 업무 분담

사용자(월성 원자력 현장 엔지니어)들과 개발자(소프트웨어 엔지니어)들의 중간에는 노심관리 업무 절차서가 있었다. 따라서 모든 요구사항 역시 절차서를 중심으로 추출하였다. 처음에 소프트웨어 엔지니어들은

절차서의 내용을 숙독하여 업무를 파악하였다. 효과적인 도제법을 수행하기 위하여 노심관리 절차서를 개발자들에게 배분하여 담당자를 지정하였다. 그러나 절차서는 업무 지침서의 성격이 강해서 업무에 대해 기본적인 지식을 포함하고 있지는 않았다. 이에 사용자들이 각 절차서에 대해서 개발자들에게 설명해 주는 단계를 거쳤다. 이 단계를 통하여 개발자들은 절차서의 내용을 파악할 수 있었다.

● 소프트웨어 요구사항서 초안 작성

개발자들이 절차서를 파악한 후 바로 도제살이를 하지는 않았다. 그 대신 소프트웨어 요구사항서 초안을 작성하였다. 소프트웨어 요구사항서 초안을 작성한 이유는 일단 개발자들이 파악한 요구사항을 문서화 함으로서 알고 있는 부분과 모르고 있는 부분을 파악하기 또한 사용자들이 소프트웨어 요구사항서를 검토하여 누락된 요구사항 및 오해하고 있는 요구사항을 지적하고, 서로간의 눈높이를 맞추기 위한 것이었다.

소프트웨어 요구사항서는 IEEE 830[2] 표준을 이용하였다. 이는 원자력 분야의 소프트웨어 요구사항서를 작성함에 있어 IEEE 830 표준을 사용할 것을 미국 원자력 안전위원회에서 권고하고 있기 때문이다.

IEEE 830 표준을 적용함에 있어 중요한 결정 사항은 기능 요구사항을 어떻게 체계적으로 표현할 것인가이다. IEEE 830에서는 기능 요구사항을 시스템의 모드, 사용자 클래스, 시스템의 feature, 외부 자극(stimululas), 기능 계층구조 또는 조직별로 표현하는 여러 방안을 제시하고 있다. COMPAS이 기능은 업무 절차서와 연관이 깊으므로 본 사례에서는 업무 절차서를 최상위 기능으로 정하고 이에 기능 부분의 목차를 구성하였다. 이 최상위 기능별로 기능을 수행하기 위하여 필요한 입.출력 데이터와 기능 수행 프로세스를 정의하였다.

● 소프트웨어 요구사항서 초안 검토

소프트웨어 요구사항서 초안을 작성하면서 개발자들은 여러 가지 의문을 갖게 되었다. 이러한 의문은 주로 절차서의 내용을 완벽하게 이해하지 못하는 부분, 소프트웨어 구현을 염두에 두었을 때 결정해야 하는 부분 등이었다. 이러한 문제를 해결하기 위하여 소프트웨어 요구사항 초안에 대한 검토회를 수행하였다. 검토회는 담당자별 개별 검토회와 전체적인 검토회의로 나누어 수행하였다. 개별 검토회는 사용자와 개발자가 각 분야별로 의문사항에 대해 토의하는 단계였고, 전체 회의는 시스템 전체에 걸친 문제로서 모든 참여자가 참석하였다. 각 분야별 회의를 통하여 결정된 사항은 전체 회의시간에 공지하여 다른 의견이 있는 엔지니어들이 의견을 개진할 수 있는 기회를 주었다.

소프트웨어 초안 검토회의를 통하여 많은 부분이 결정되었으나 여전히 결정할 수 없는 부분이 있었다. 또한 개발자가 궁금했던 부분에 대한 해답을 얻는 과정에서 새로운 쟁점사항이 발생하기도 하였다.

소프트웨어 초안 검토회를 통해서 개발자들은 개발할 소프트웨어에 대해 좀 더 명확한 비전을 가질 수 있었다.

● 도제 수행

절차서에 대한 검토와 소프트웨어 초안에 대한 검토과정을 거치면서 개발자들은 업무에 대해 어느 정도 파악하게 되었고, 실제 업무를 수행하면서 요구사항을 좀 더 명확히 하기 위한 도제수행을 위한 기본적인 능력을 갖추게 되었다.

도제수행은 각 담당자가 수행하는 업무를 직접 현장에서 수행해 보는 방식을 택했다.

이 과정을 통해서 개발자어들은 사용자들의 요구사항을 잘 못 이해하고 있는 부분을 발견하기도 하였고, 업무에 대한 확신을 갖게 되기도 하였다. 원자력 발전소의 노심관리 업무가 매우 전문화된 업무이기 때문에 많은 준비 작업에도 불구하고 각 업무를 실제로 수행하기 위해서는 상당한 노력이 필요했다.

● 최종 소프트웨어 요구사항서 작성

소프트웨어 요구사항서 초안에 대한 쟁점사항과, 도제 수행 과정에서 새로 발견되거나 수정된 요구사항을 정리하여 최종 소프트웨어 요구사항서를 작성하였다. 비록 최종 소프트웨어 요구항서라고는 하지만 아직 해결되지 않은 쟁점사항이 남아 있었다. 그러나 이러한 개발의 다음단계로 진행하기 위해서는 베이스라인을 설정할 필요가 있었기 때문에 최종 요구사항서를 작성하여 요구사항 관리 체제로 전환할 필요가 있었다. 미해결된 쟁점사항에 대해서는 해결 완료 시점을 지정하여 관리하였다.

최종 소프트웨어 요구사항서에 대한 검토는 공식 회의를 통하지 않고 문서 검토 작업으로 대체하였다. 이는 대부분의 변경내용이 미리 논의된 사항이기 때문이었다.

최종 소프트웨어 요구사항서는 다음과 같이 구성하였다.

1. 서론
2. 시스템 일반사항
   2.1 시스템 인터페이스 및 운용
   2.2 시스템의 기능
   2.3 시스템 구성
   2.4 제약사항
   2.5 가정 및 기타 의존사항
   2.6 부록
3. 세부 요구사항
   3.1 핵연료 관리
   3.2 원자로 안전관리
   3.3 핵 계측기 관리
   3.4 업무 기술 지원
4. 시스템 성능 및 보안 요구사항
   4.1 성능 요구사항
   4.2 신뢰성
   4.3 보안성
   4.4 소프트웨어 확인 및 검증기준
5. 용어 및 약어

3. 결론

소프트웨어 요구사항을 추출함에 있어 가장 중요한 점은 추출 방법을 결정하는 것이다. 추출방법은 대상 소프트웨어 시스템의 특성, 사용자 및 개발자의 특성, 기간 및 비용 등이 고려되어야 할 것이다.

본 사례에서는 업무의 전문성을 고려하여 도제법을 이용하여 소프트웨어 요구사항을 추출하였다. 도제법의 특성은 도제를 수행하는 시스템 엔지니어의 노력이 많이 소요된다는 점이며 본 사례에서도 소프트웨어 엔지니어들이 노심관리 업무 자동화를 위한 사용자들의 요구사항을 추출하는데 많은 노력이 소요되었다.

본 사례를 통하여 체득한 교훈은 다음과 같다.

1) 도제법을 수행하기 전에 업무를 미리 파악하는 것이 시스템 엔지니어가 많은 노력이 소요된다는 도제법의 단점을 조금이나마 보완할 수 있다.

2) 요구사항서를 문서화 하는 것이 사용자와 개발자 모두에게 의사소통을 원활하게 하기 위한 도구로 유용하게 사용되므로, 가능한 초기에 요구사항을 문서화하는 것이 좋다.

3) 요구사항 추출과정을 심도 있게 수행하더라도 식별되지 않은 요구사항과 쟁점사항은 존재하게 된다. 이는 현 단계에서 결정하기 어렵고, 설계가 구체적으로 진행되어 다른 내용이 결정된 후에 정할 수 있는 경우와, 사용자가 결정하지 못하는 경우로 나뉘어 질 수 있다. 모든 쟁점사항을 해결하지 못하더라도 쟁점사항이잘 관리되기만 한다면 예상치 않은 문제에 봉착하는 상황을 만들지 않을 수 있다.

4) 시스템 엔지니어가 요구사항서 초안을 작성할 경우 누락되는 항목이 없도록 주의하여야 한다. 사용자는 시스템 엔지니어가 요구사항서를 작성하고 검토를 요청할 경우 스스로 누락된 요구사항을 제시하지 못하는 경향이 있다.

아직까지 요구사항 추출에 대한 국내 사례가 많지 않은 것 같다. 본 사례는 이러한 상황을 극복하고자 COMPAS 개발 과정에서 요구사항을 추출한 사례를 제시하였다. 다른 개발사업에서도 실제 요구사항을 추출한 사례에 대한 경험과 교훈을 공유하여 우리나라의 시스템 엔지니어링 발전에 기여했으면 한다.

4. 참고 문헌

[1] Harold J. Heydt. et. al., " Tools for Requirements Discovery, Creation, and Elicitation", INCOSE 2002 Symposium.

[2]IEEE-SA Standard Board, IEEE Std 830-1998 IEEE Recommended Practice for Software Requirement Specification, 1998.