

한국형 에너지관리 시스템 개발에서의 기능통합 및 시험사례

김완홍, 박민경, 윤여준, 이효상
한국전력거래소*

Integration of Korean Energy Management System Functions and Test Cases

Wan-Hong Kim, Min-Ryung Park, Yeo-Jun Yoon, Hyo-Sang Lee
Kora Power Exchange

Abstract - 한국형 에너지관리 시스템 개발 사업은 전력거래소에서 사용하는 현 EMS 설비의 교체수명 주기가 다가오게 되어 새로운 EMS 설비를 순수 국내 전력IT 기술을 통해 개발하는 총 개발기간 5년의 대규모 국가 프로젝트이다. 본 논문에서는 현 Baseline 단계에서의 KEMS 개발내용을 소개하고 기능개발과 관리, 어플리케이션의 통합 및 검증기법 등에 대해 소개하고자 한다.

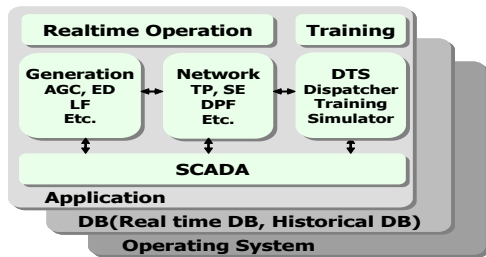
1. 서 론

전력거래소는 전력시장의 구조 개편과 함께 경쟁 전력시장 하에서 계통운영과 시장운영을 위해 2001년 4월에 설립된 기관이다. 전력시장의 개편과정에 따라 EMS, MOS, CBP 시스템을 도입하여 이들을 연계 운영하고 있으며 이 중 EMS의 경우 1979년 ELD 시스템, 1988년 Toshiba EMS 시스템 그리고 2001년 Aрева의 EMS 시스템을 순차적으로 도입하여 안정적인 계통운영을 해오고 있다.

최근 전력계통은 설비 대형화와 다양한 분산전원의 등장으로 운영의 복잡도가 날이 증가하여 계통운영의 핵심 설비인 EMS에 대한 중요성이 날로 커지고 있다. 이에 2005년 말, 국책 전력IT 과제로 한국형 EMS(이하 KEMS)를 선정하고 계통운영자인 전력거래소와 전력IT 산업체, 연구소, 대학 등 9개의 참여기관과 6개 위탁기관으로부터 약 300명 가량의 연구원이 참여한 산학연 공동 연구 컨소시엄이 구성되었다[1]. 실 EMS의 운영기관인 KPX가 총괄주관을 담당하고 SCADA, 발전제어, 계통해석, DTS 개발 등 5개의 과제로 나뉘어 진행된다. 본 논문에서는 Baseline 단계의 개발결과를 바탕으로 KEMS 어플리케이션의 개발 및 통합, 그리고 검증 기법 등에 대해 소개하고자 한다.

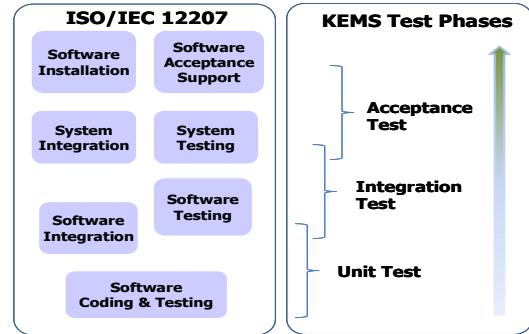
2. 개발 환경

총 30개월(2001~ 2005)에 걸쳐 개발이 이루어지며 효율적 개발 및 위험관리를 위해 Baseline, Prototype, Full-Scale 등 3단계 구분을 통해 개발, 검증 및 기능 확장이 이루어지게 된다. 현재 KEMS는 전체 개발 단계 중 첫 단계인 Baseline의 개발 및 검증이 완료된 상태이다. K-EMS 개발은 그림 1과 같이 SCADA, 발전계획, 계통해석, 보고서, 급전원 교육시스템 등 EMS의 모든 어플리케이션의 개발은 물론 이러한 어플리케이션의 기반이 되는 실시간 통합 데이터베이스, 이력 데이터베이스, MMI, Engineering 툴 등 EMS 상용을 위한 모든 부분을 포함한다[2].



〈그림 1〉 KEMS 기능구성

EMS는 많은 소프트웨어가 유기적으로 동작함으로써 제 기능을 수행할 수 있다. 일반적으로 대규모 소프트웨어의 개발은 여러 가지의 정보 시스템 개발방법론[3-5]을 적용하고 있는데 국제 표준인 ISO/IEC 12207[6]에 따르면 구축 및 시험과정을 그림 2와 같이 7개의 단계로 정의하고 있다. KEMS 개발의 경우, ISO/IEC 12207을 그대로 준용하지는 않았으나 통합 및 시험 절차는 그와 유사한 형태를 택하고 있다. KEMS의 시험은 Unit Test, Integration Test, Acceptance Test로 구분되며 ISO/IEC 12207 기준의 절차와의 유사성은 다음 그림과 같다.



〈그림 2〉 ISO/IEC12207 표준과 KEMS 시험절차

시험에 소요되는 노력은 전체 개발노력의 50% 이상이 된다. KEMS의 경우 13개월의 요구사항 분석 및 설계과정과 8개월의 개발과정, 9개월의 시험과정을 거치고 있다. 소프트웨어 통합시험의 경우 한 번에 모든 모듈을 시험하기 어려운 경우가 많아 모듈을 하나씩 추가하면서 시험하는 점진적 접근방법(Incremental Approach)을 적용하였다.

KEMS를 개발하는데 있어서 개발자들에 대한 권고 환경은 HP UNIX 기반의 C, C++이었으나 개발자들의 역량과 개발환경에 따라 UNIX 기반의 C, C++ 환경과 Windows 기반의 PC 환경에서 개발되는 부분이 혼재하게 되었으며, 개발언어도 선호도에 따라 C, C++, C#, Java, Fortran 등으로 다양하게 사용되었다. 소프트웨어 개발 단계에서부터 통일된 개발언어를 사용하는 것이 유리하였으나 일부 개발자들은 이미 개발된 소프트웨어를 재사용하는 경우가 있었고, 프로그램이 개발되는 지역적인 위치도 서로 떨어져 있어서 상호 협력하여 개발하는 환경보다는 독립적으로 개발하는 환경이 되었다.

KEMS의 운영환경(HP UNIX 기반)에 맞추어 일부 독자적으로 개발된 소프트웨어의 언어를 HP-UX 기반의 C 또는 C++로 변환하는 작업이 수행되었고, 다른 환경에서 개발된 소프트웨어나 다른 개발언어로 작성된 소프트웨어는 UNIX 기반으로 porting된 후에 다시 단위시험이 수행되었다.

3. 시스템 통합 및 시험

각 단계마다 개발기능의 신뢰성을 검증하기 위해 단위시험, 통합시험, 실증시험이 수행된다. 단위시험에서는 알고리즘에 대한 검증이 이루어지고, 통합시험에서는 모듈 프로그램과 데이터베이스의 연계, MMI에 대한 검증이 이루어진다. 각 단계의 최종 검증단계인 실증시험에서는 EMS 실무자를 대상으로 별도로 구축된 Testbed에서 실제 계통 연계 환경에서의 기능 및 성능을 검증 받게 된다. 각 단계별 주요 시험환경과 내용은 다음과 같다.

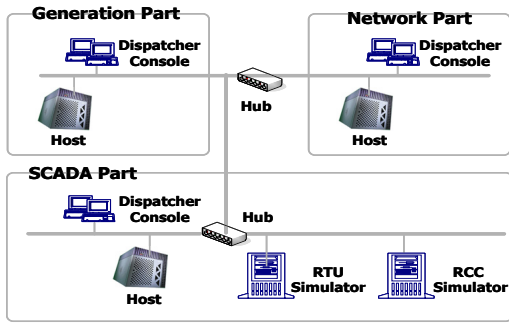
3.1 단위 시험

소프트웨어 모듈의 논리적 오류를 발견할 목적으로 단위시험이 수행된다. 인터페이스되는 상위 모듈과 하위모듈을 대신하여 각각을 흉내 내는 'Driver'와 'Stub'이 필요에 따라 작성하여 사용하였다. 단위시험 단계에서는 EMS 실제 운영환경과 동일한 DB 인터페이스나 GUI는 생략되었으며 필요한 데이터는 Flat File 형태로 자료파일이 활용되었다.

3.2 통합 시험

통합시험을 통해서 각 어플리케이션 기능 간 통합 여부 및 인터페이스 오류(Alarm, DB, UI etc.) 등을 확인한다. 통합시험에서는 실제 계통 통합 DB를 구축하여 활용하였으며 실 사용자 참여를 통해 요구사항 대비 기능구현 여부를 확인하였다.

A) 통합시험 Configuration : 통합DB, API, UI환경에서 각 과제별 어플리케이션 검증 통합시험 구성도는 그림 3과 같다.



<그림 3> 통합시험 구성도

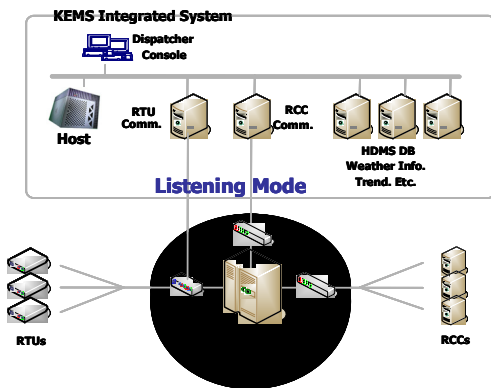
B) 통합시험용 DB : DB의 신뢰도 확보를 위해 현 운용 중인 EMS 계통DB로부터의 Migration 방안을 채택하였으며 단계적으로 제주계통, 전 계통 DB를 구축하였다.

C) 데이터취득: 실 계통 데이터 취득이 불가능한 관계로 RTU, ICCP 시뮬레이터를 활용하여 임의의 데이터를 생성하여 사용하였으며, 데이터의 의미가 중요한 일부 어플리케이션의 경우 실제 계통운영 이력자료를 활용하였다.

3.3 실증 시험

실제 계통을 연계하여 EMS의 전체기능이 안정적이고 원활하게 수행되는지 확인하고 통합시스템에 대한 Functional Test와 100시간 연속운전시험을 수행하였다. 이를 위해 현 운영 중인 후비 급전소에 별도의 Testbed를 마련하고 리스닝 모드 환경을 구축하여 현 EMS와의 비교검증이 가능하게 했다. 또한 실증시험에는 EMS 관련 실무자 및 급전원들을 참여시켜 KEMS 기능의 실용성을 검증하였다.

A) 실증시험 Configuration : 전력거래소 후비급전소의 별도 공간에 그림 5와 같이 KEMS실증시험용 Test bed를 마련하고 통합시스템 설치하였다.



<그림 5> 실증 시험 리스닝 모드 구성도

B) 실증시험용 DB : 현 운용 중인 EMS 계통DB[7]의 Migration을 통해 전 계통 DB 구축하였다.

C) 데이터취득 : 그림 5와 같은 리스닝 모드환경을 통해 전국의 RTU, RCC 현장 데이터를 현 EMS와 동시에 취득하였다.

4. 교 혼

KEMS개발 프로젝트 개발 환경은 여러 단계의 많은 인력이 참여하고 있기 때문에 완벽한 프로젝트 관리 측면에서 어려운 점이 많았다. 개발단계에서 겪은 오류와 시행착오 등 통합 및 시험단계에서 얻은 교훈은 다음과 같다.

A) 현실적인 시험계획의 수립

소프트웨어의 개발이 완료되는 시점에서 시험계획을 수립하게 되면 일정애 쫓기게 되고 시험계획이 상세하게 작성되기 어렵다. 시험계획이 현실적이지 못하면 도중에 시험계획을 재수립하는 오류가 자주 발생하게 된다. 시험의 목적을 분명히 하고 시험시기와 방법에 대하여 실현성을 확보하여야 한다. 너무 이상적이거나 낙천적인 계획은 프로젝트에 투입된 많은 인력들이 혼란을 겪게 만든다.

B) 시험 제약요소에 대한 철저한 분석

예기치 못한 돌발적 제약사항이 발생하여 시험 수행이 불가능할 수도 있다. 예를 들어, 기능시험에 필요한 하드웨어가 시험시점까지 제공되지 않거나 시험에 필요한 데이터가 올바르게 제공되지 않을 수 있고 특정이 필요한 시점에서 해당 인력이 가용하지 않을 수도 있다. 시험환경, 데이터, 하드웨어, 데이터 인터페이스 등 시험에 필요한 요소를 최대한 나열하고 그에 대한 가용성을 파악함으로써 제약요소를 판별하여야 한다.

C) 시험절차서 개발

시험계획에 따라 시험항목(test case), 시험절차, 시험 데이터, 예상결과, 성공 판정조건 (success criteria) 등을 개발하여야 한다. 특히 절차서의 내용은 개발자가 아닌 제3자에 의해 시험이 진행될 수 있도록 상세히 작성되어야 하며, 개발된 기능을 완벽히 시험할 수 있도록 내용이 충분하여야 한다. KEMS의 경우 시험절차서가 부족한 경우 시험을 성공적으로 완료했다고 하더라도 기능에 결함이 지속적으로 발생하게 된다.

D) 개발환경과 시험환경의 분리

소프트웨어의 개발환경과 시험환경을 완벽히 분리하는 것이 필요하다. 시험환경에서는 소프트웨어의 개발이나 디버깅 작업이 이루어져서는 안 된다. 이것이 지켜지지 않으면 개발환경과 시험환경이 혼용되어 소프트웨어나 DB, 환경정보 등의 형상관리가 어렵게 된다. 개발환경에서 freeze된 version의 실행파일만 시험환경에 설치하여야 하며, 개발자는 시험에 참여하지 않도록 하는 것이 좋다. 개발자가 시험에 참여하게 되면 개발자 시각에 편중(bias)되어 시험이 올바르게 진행되지 못할 우려가 있다.

E) Issue 관리방안 마련

기능에 대한 요구사항과 구현결과에 대한 Variance 또는 기능개선이 요구되는 'Enhancement' 사항 등을 추적, 관리할 수 있어야 한다. KEMS의 경우 자체 개발한 소프트웨어 툴을 사용하여 Variance 및 Enhancement에 대한 최종상태(open/closed), 등급(class), 내용, 조치계획 등을 관리하였다.

F) 소프트웨어 개발방법론의 적용

KEMS개발 프로젝트의 사례를 보면 프로젝트를 위해 통일된 개발방법론 적용이 매우 유리할 것으로 판단된다. 프로젝트의 효율적 진행을 위해서는 개발자 모두가 개발방법론에 대한 이해가 바탕이 되면 좋겠으나 최소한 각 개발 분야의 파트리더(Part Leader)들은 이에 대한 충분한 이해가 요구된다. 적용되는 개발방법론에 대한 이해가 부족할 경우 개발자간 의사소통이 원활치 못하고 산출물의 관리가 부실하게 된다. 올바른 개발방법론과 프로젝트 관리기법을 적용한다면 위에서 열거한 여러 교훈을 다시 겪지 않게 될 것이다.

5. 결 론

소프트웨어를 통합하고 시험하는 목적은 요구사항에 대한 올바른 구현 여부를 확인하는데 있다. 이를 확인하기 위해서는 단위시험부터 통합시험까지 일관된 기준과 절차에 따라 시험을 진행해야 한다. KEMS는 각 시험단계별로 시험의 성격과 목적에 따라 시험계획과 절차를 준비하여 진행하였다. 대부분의 개발자들이 유사한 형태의 프로젝트 경험이 부족하여 시행착오와 혼선이 많았으나 그 만큼 얻은 점도 많다. Baseline 단계에서 얻은 교훈을 통해 KEMS는 향후 Prototype, Fullscale단계에서는 훨씬 효율적으로 진행될 것으로 기대되고 있다.

본 논문은 지식경제부에서 시행한 전력산업 연구개발사업(과제번호 : R-2005-1-398-004)으로 수행되었습니다. 관계자분들께 감사드립니다.

[참 고 문 헌]

- [1] Korean Energy Management System Development Webpage, Available: <http://www.k-ems.re.kr>.
- [2] KEMS 기술규격서.(KPX, 2005)
- [3] V.R. Basili and R.W. Selby, "Comparing the effectiveness of software testing strategies," IEEE Transactions on Software Engineering, Vol.13, Issue 12, pp.1278 - 1296, Dec. 1987.
- [4] Kelvin Ross, "Practical Guide to Software System Testing," K.J.Ross & Associates Pty. Ltd., 1998.
- [5] IEEE Std 829-1983, "IEEE Standard For Software Test Documentation," pp.1-48, Feb 1983.
- [6] ISO/IEC 12207 Webpage, Available: <http://www.jtcl-sc7.org>.
- [7] Hyosang Lee, Wanhong Kim, Minryung Park and Yeojun Yoon, "A Study of extended and integrated database implementation in Korean Energy Management System," ELECO 5th International Conference, December 2007.