전력계통 데이터베이스 기반 EMS 응용프로그램 개발에 관한 고찰

허성일*, 김선구*, 이효상*, 이진호°
* 한국전력거래소, ° LS산전

Considerations on development of EMS applications based on power system database

HUR, SEONG IL* KIM, SEON GU* LEE, HYO-SANG* LEE, JIN-HO°
* Korea Power Exchange ° LS Industrial System Co.

Abstract - 전력계통 컨트롤센터에서 사용되는 SCADA/EMS 응용프로그램은, 전력계통의 해석을 위해 특정한 포맷의 입력데이터 파일을 사용하는 일반적인 오프라인 응용프로그램과는 달리, 물리적 특성이 통합된 전력계통 데이터베이스 기반위에서 수행된다. 응용프로그램들은 데이터베이스를 통해 필요한 데이터를 상호 교환하며, 개별 응용프로그램은 데이터베이스와 연동하는 사용자화면을 통해 컨트롤센터의 운전원과 상호작용을 하게 된다. 따라서 전력계통 데이터베이스 기반위에서 수행되는 EMS용 응용프로그램을 개발하기 위해서는 먼저 프로그램 고유의 기능적인 측면은 물론 운전원을 포함한 여러 사용자를 위한 프로그램 운용적인 측면의 요소까지 종합 분석하여 필요한 기능과 화면 그리고 그에 따른 데이터를 도출해야 한다. 도출된 기능 및 화면설계안에 따라 개발된 프로그램과 사용자화면은 모든 입출력 데이터들이 반영된 데이터베이스와의 인터페이스를 통해 통합 EMS용 응용프로그램으로서의 형태를 갖추게 된다. 최종적으로 응용프로그램은 사전에 준비된 시험절차에 따라 검증(프로그램의 정확성, 신속성 그리고 신뢰성을 평가)과정을 수행한다. 본 논문은 전력IT 국가전략과제로 개발 진행 중인 한국형 EMS(K-EMS) 응용프로그램의 개발 프로세스와 각 프로세스에서의 고려사항들을 소개함으로써, 한국형 EMS를 위한 응용프로그램의 추가 개발 혹은 전력계통 데이터베이스 구축과 이를 활용한 응용프로그램의 개발을 위한 방법론을 제안하고자 한다.

1. 개 요

전력계통 컨트롤센터의 운전원은 SCADA/EMS 응용프로그램을 통해 실시간 전력수급제어는 물론 송전망의 안전성 평가를 수행하고 있다. 응용프로그램들은 설정된 실행주기간격마다 반복적으로 구동되며, 실행주기내에서는 정의된 시퀀스에 맞게 순차적으로 수행된다. 응용프로그램들이 동일한 시점에 해당하는 전체 전력계통의 실시간 정적정보를 공유하기 위해서는 프로그램 실행동안 데이터의 일관성과 지속성이 확보되어야 함은 필수적이다. 프로그램에 사용되는 데이터들은 전력계통 설비정수 데이터, 측정데이터, 사용자 입력 데이터, 출력 데이터, 그리고 스케줄 데이터 등 다양하게 존재한다. 데이터 성격에 따라 갱신주기도 다르며, 순차적으로 실행되는 응용프로그램의 경우 출력데이터는 타 응용프로그램의 입력데이터로 사용되므로 데이터의 동시성도 반드시 고려되어야 한다. 일반적으로 EMS 응용프로그램을 위한 전력계통의 통합 데이터저장소격인 전력계통 데이터베이스는 이러한 요소들을 종합적으로 고려하여 설계되고 구축된다.

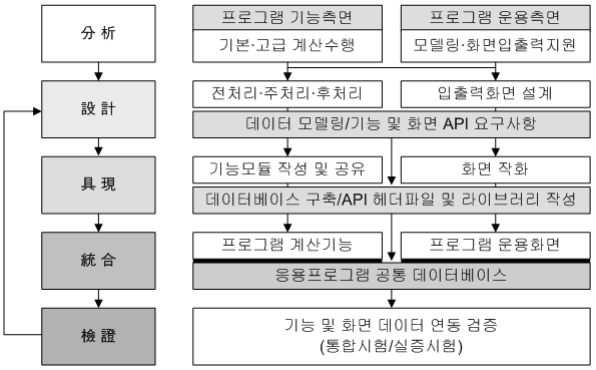
응용프로그램에서 사용되는 다양한 입·출력데이터들은 프로그램의 기능적인 측면과 운용적인 측면을 종합적으로 고려해서 도출해야 한다. 전자는 프로그램 고유의 계산수행에 관한 사항들을, 후자는 사용자 화면을 통한 프로그램의 활용에 관한 사항들을 말하는 것이다. 운용적인 측면을 분석할 경우에는 사용자 화면을 통한 모델링 및 입출력 기능과 같은 해당 프로그램의 사용자 운용지원요소뿐만 아니라 동일한 데이터를 공유하는 전체 응용프로그램간의 공통적인 설계요소를 함께 고려해서 프로그램간의 중복적인 요소를 배제하여 데이터베이스의 효율성을 저해시키지 않도록 해야 한다. 기능 및 화면 요구사항 분석을 통해 도출된 데이터는 유형에 따라 분류하여 데이터베이스에 반영하도록 한다(<표.1>). 응용 프로그램의 주요 연산 기능과 전처리 및 후처리 기능은 상호호환을 고려하여 표준코드를 통해 생성하고, 필요로 하는 입출력 화면들의 작화가 완료되면 데이터베이스와의 인터페이스를 통해 프로그램을 통합하는 과정을 거친다. 최종적으로 준비된 시험절차에 의거하여 프로그램 기능 및 화면의 데이터 연동 그리고 프로그램 성능 검증을 수행한다. 검증과정에서 도출된 문제점에 대한 보완사항은 설계단계에 다시 반영한 후 구현, 통합 그리고 검증과정의 이후 단계를 차례대로 수행하게 된다. <그림.1>은 EMS 응용프로그램의 개발 프로세스를 도식화하여 보여주고 있다. 다음 장에서는 각 프로세스 단계별 구체적인 내용을 기술하고자 한다.

<표 1> EMS 응용프로그램용 입출력 데이터 종류

데이터 종류	데이터 성격	데이터 생성	데이터 갱신
전력계통 토폴로지 및 설비정수데이터 온라인 입력데이터	정적 데이터 (Static)	시스템 생성 및 갱신 시점	시스템 생성 및 갱신주기 간격
측정데이터 (아날로그/상태)	실시간 데이터 (Real-time)	해당데이터 추출시점 (취득주기내에서 불변)	취득주기 간격
응용프로그램 출력 데이터	파생 데이터 (Derived)	해당데이터 계산시점 (실행주기내에서 불변)	실행주기 간격

2. EMS 응용프로그램 개발

이 장에서는 EMS 응용프로그램 개발을 위해 거쳐야하는 분석-설계-구현-통합-검증의 5가지 프로세스의 내용에 대해 설명하고자 한다. 특히, 데이터 파일기반의 오프라인 전력계통 해석도구와 구별되는 데이터 모델링 그리고 응용프로그램과 데이터베이스 통합에 초점을 맞추고자 한다.



<그림 1> EMS 응용프로그램 개발 프로세스

2.1 요구사항 분석

응용프로그램 개발의 첫 단계로서 프로그램 수행목적에 맞는 기능과 운용측면의 요구사항을 면밀히 분석하여 개발항목들을 도출한다. 도출된 항목가운데 기 개발된 타 응용프로그램과의 유사·중복되는 기능과 화면 항목이 있는지 살펴서 함께 사용할 수 있을 경우에는 공유하는 방안을 검토한다.

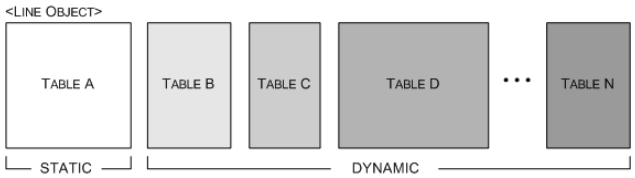
2.2 응용프로그램 설계

프로그램 구현을 위한 구체적인 수준의 설계서를 작성하는 단계로 요구사항 분석을 통해 도출된 기능과 화면설계 그리고 각각의 필요한 데이터 모델링을 수행한다.

프로그램 기능들은 크게 입력데이터 준비기능이나 출력데이터 처리기능 그리고 주요 연산기능들로 분류하고 각각의 세부기능을 잘게 쪼개어 기존에 개발된 타 응용프로그램의 세부기능을 모듈화하여 재활용할 수 있다면 프로그램 개발을 수월하게 진행할 수 있다. 화면의 경우에는 사용자 입력지원 화면, 응용프로그램 수행 제어화면, 응용프로그램의 정확성과 성능을 확인할 수 있는 분석화면 그리고 응용프로그램에 의한 계산결과로서 출력화면으로 나뉜다. 특히 출력화면의 경우 전체 계통상태의 중요한 정보를 쉽고 정확하게 보여줌으로써 운전원의 신속한 판단과 의사결정을 지원할 수 있도록 설계해야 한다. 이를 위해서 화면을 분류하고 개별 화면에 나타나는 출력요소들의 선정과 배치 그리고 크기와 색상과 같은 표현에 있어 인지공학적인 차원에서 접근이 필요하다. 화면을 통틀어 운전원이 보고자 하는 정보의 빠르고 효율적인 탐색이 가능

하도록 화면간의 전개(이동)도 고려해야 한다.

응용프로그램의 기능과 화면을 위해 도출된 데이터의 경우 공통 데이터베이스에 반영해야 한다. 이를 위해서는 프로그램 기능과 화면에 의한 데이터 접근이 빠르게 이뤄질 수 있도록 데이터 저장을 최적화해야 한다. 기존에 데이터베이스가 구축되어 있는 경우라면 먼저 데이터베이스 구조를 살펴서 기존 데이터와 중복이 되는지 살핀 후에 신규 추가 데이터들이 적절한 곳에 배치될 수 있도록 설계해야 한다.



〈그림 2〉 응용프로그램 공통 데이터베이스 구조

<그림.2>는 한국형 EMS의 응용프로그램을 위한 공통 데이터베이스의 개체별(선로, 변압기 등) 데이터 저장 구조를 일례로 나타낸 것이다. <표.1>에 기술된 바와 같이 개체마다 저장되는 데이터 성격에 따라 여러 개의 데이터 저장 테이블을 갖는다. 메모리사용의 효율성과 데이터 활용목적에 따라 분류하였지만, 결과적으로 각 테이블의 레코드는 동일한 개체아이디로 평행하게 연결되므로 프로그램 입장에서는 각 개체에 대해 하나의 테이블로 바라보고 접근하는 것으로 간주할 수 있다.

데이터분류에 의해 응용프로그램의 실행주기와 상관없이 값이 일정한 데이터는 정적데이터 테이블에, 응용프로그램 실행주기마다 갱신되는 데이터는 동적데이터 테이블에 데이터 이름과 타입을 지정하여 추가하면 된다. 동적데이터 테이블의 경우 특정 응용프로그램의 필요에 따라 새로이 추가할 수도 있다. 만약 기존 데이터베이스에 개체가 정의되어 있지 않은 경우에는 개체를 신규로 정의하고 기존개체와의 연관관계를 정의해주어야 한다. 이상 응용프로그램 공통 데이터베이스에 신규데이터를 추가할 경우의 고려사항을 요약하면 다음과 같다.

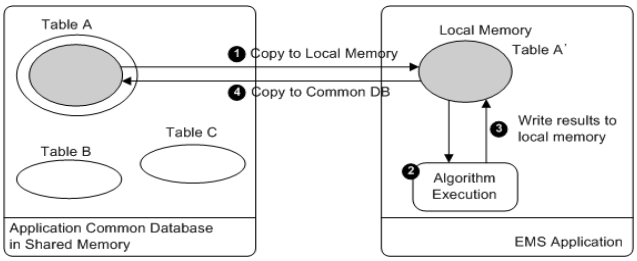
○ 데이터 중복여부 확인
○ 데이터베이스에 개체 테이블이 정의되어 있는 경우
  - 정적데이터와 동적데이터를 구분하여 필요한 테이블에 반영
  - 신규 동적테이블을 추가할 경우 기존 테이블과의 연관관계 정의
○ 데이터베이스에 개체 테이블이 정의되어 있지 않은 경우
  - 신규 개체를 추가할 경우 기존 개체와의 연관관계 정의
  - 신규 개체의 정적 그리고 동적 테이블 정의
  - 각 테이블에 필요한 데이터 레코드의 최대 개수 정의

2.3 응용프로그램 구현

기능과 화면에 대한 상세설계를 바탕으로 프로그램을 개발하고 화면을 제작하며 응용프로그램을 위한 데이터베이스를 구축(혹은 갱신)한다. 이 단계에서 응용프로그램은 기능상 기본 골격을 갖추게 되므로 특정한 포맷(csv)을 가지는 파일기반 데이터를 이용하여 프로그램을 구성하는 단위세부기능을 검증해볼 수 있다. 하지만 데이터베이스를 기반으로 한 응용프로그램으로서의 기능과 성능을 종합적으로 검증하기 위해서는 데이터베이스와의 인터페이스 작업을 거쳐야 하는데, 이를 위해 데이터베이스의 테이블 구조를 정의하고 필요한 정보의 접근을 위한 응용프로그램 인터페이스(API) 헤더파일과 함수 라이브러리를 작성해야 한다. 여기에 대해서는 2.4절에서 기술하고자 한다.

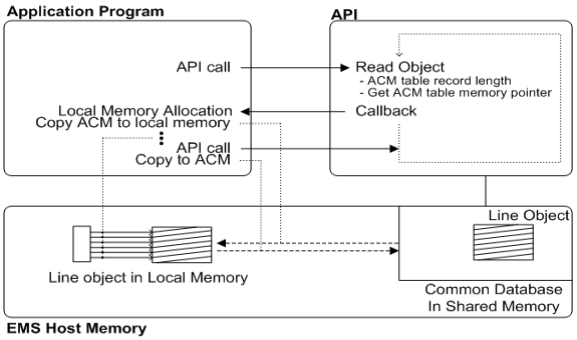
2.4 응용프로그램 인터페이스

모든 EMS 응용프로그램을 위한 공통 데이터베이스는 공유메모리에 상주하며, 개별 응용프로그램은 수행초기에 필요한 데이터를 로컬메모리에 복사하여 사용하게 된다. 다시 말해서, 응용프로그램은 로컬메모리의 데이터를 읽고 연산결과를 로컬메모리에 기록한 후 다시 공유메모리상의 데이터베이스로 데이터를 복사하게 된다(그림.3). 이를 통해 응용프로그램 상호간의 필요한 데이터를 공유하며, 화면을 통해 필요한 정보를 사용자에게 제공하게 된다.



〈그림 3〉 응용프로그램과 데이터베이스 연동 개념도

공유메모리에 상주하는 데이터베이스에는 모든 응용프로그램에서 공통으로 사용되는 데이터들이 전부 포함되어 있으나, 특정 응용프로그램에서는 데이터베이스의 모든 데이터를 필요로 하지 않으므로 필요한 데이터만을 별도로 정의해서 사용할 필요가 있다. 별도로 정의된 데이터는 응용프로그램내의 로컬메모리로 정의한다. 이는 공통 데이터베이스의 전체 데이터구조 정의와 마찬가지로 로컬메모리에 해당하는 데이터구조를 별도로 정의하는 것이다. 이를 위해 프로그램 개발자는 글로벌 헤더파일과 로컬 헤더파일을 별도로 작성하여 코드에 포함시켜야 한다.



〈그림 4〉 응용프로그램 인터페이스(API)

• 글로벌 헤더파일: 공유메모리에 상주하는 공통 데이터베이스의 테이블구조 정의 및 데이터베이스 관련 API함수를 정의한 파일
• 로컬 헤더파일: 응용프로그램에서 사용하기 위한 로컬메모리에 대한 정의가 기술되어 있는 파일
• API: 응용프로그램에서 사용되는 함수들을 정의한 라이브러리로, 프로그램 개발자는 필요한 API함수 호출을 통해 데이터베이스의 물리적 구조에 대한 여러 가지 정보를 리턴 받는다.

응용프로그램을 위한 입력데이터 처리는 API함수가 아닌 프로그램 개발자가 작성하는 함수에 의해 이뤄진다. 프로그램 개발자는 API함수 호출을 통해 접근하고자 하는 데이터베이스의 테이블 구조와 관련한 정보를 얻어온다. 다시 말해서 프로그램은 API함수를 통해 데이터베이스의 특정 테이블 레코드 길이나 테이블 메모리 포인터와 같은 정보를 읽어서 사용자가 개발한 입▪출력데이터 처리 함수로 넘겨주면, 함수는 필요한 데이터를 테이블단위로 공유메모리에서 로컬메모리로 옮겨와서 데이터 연산처리를 수행한 후에 필요할 경우 결과데이터를 공유메모리의 해당 테이블에 기록하는 작업을 수행하게 된다(그림.4).

2.5 통합 EMS 응용프로그램 검증

API 헤더파일과 라이브러리를 포함하여 프로그램이 완성되면 데이터베이스 기반위에서 구동할 수 있는 환경은 만들어진 것으로 볼 수 있다. 즉, 기존의 EMS 환경에 새로운 응용프로그램의 통합과정을 마친 것이다. 마지막으로 프로그램 검증을 위해서 데이터베이스의 데이터를 채우고 데이터를 바꾸어가면서 시험절차에 따라 기능과 사용자 화면의 출력데이터를 확인하고 요구사항 대비 미비한 부분은 설계단계에서 재검토를 통해 보완이 이뤄질 수 있도록 한다.

3. 결　　론

지금까지 한국형 EMS(K-EMS)의 베이스라인 응용프로그램 개발경험을 근거로 전력계통 데이터베이스 기반에서 구동되는 EMS 응용프로그램 개발 프로세스와 각 부분에 있어 고려할 사항들에 대해 개략적으로 기술하였다. 본 논문에서는 한국형 EMS 개발과제를 위한 응용프로그램의 추가개발과 특정한 포맷의 파일기반의 전력계통 해석도구 및 개발환경에 익숙한 전력계통 엔지니어에게 데이터베이스를 활용하는 응용프로그램 개발에 필요한 방법론을 제안하고자 데이터 모델링 그리고 응용프로그램과 데이터베이스 인터페이스 부분에 초점을 두고 자세히 설명하고자 하였다. 논문에서 언급한 개발 프로세스를 통해 개발된 응용프로그램으로는 현재 자동발전제어, 경제급전, 토폴로지처리, 상태추정 그리고 급전원 조류계산이 있으며 향후 상정고장해석, 최적조류계산, 전압계획 등이 추가로 개발될 예정이다.

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

[참 고 문 헌]

[1] 한국전력거래소 KEMS 개발실, "K-EMS 기술지침서07-01", 2007
[2] 허성일, "한국형 EMS의 계통해석 응용프로그램을 위한 DB 및 S/W 구조설계에 관한 연구" 대한전기학회 하계학술대회, 2007