

객체지향 데이터베이스를 이용한 전력조류계산

박지호, 백영식
 경북대학교 전자전기컴퓨터공학부

Load Flow Analysis using Object-oriented Database

Ji Ho Park, Young Sik Baek

School of Electrical Engineering and Computer Science, Kyungpook National University

Abstract - This paper deals with Object-oriented Database(OODB) which is designed for power system analysis, and describes load flow analysis using OODB. Database management systems are widely used and achieve high reliability of data management in the engineering fields. However conventional database systems, relational database system, have disadvantages in application to power system analysis. This paper presents advantages of OODB for modeling complex data of a power system analysis. The OODB supports data modeling requirements of GUI applications and the performance is well acceptable for GUI applications.

1. 서 론

전력계통의 해석분야에서 데이터베이스 관리시스템(DBMS)이 도입되기 전에는 데이터는 파일에 저장되고 그것을 취급하는 것은 응용프로그램을 작성하는 프로그래머의 책임이었다. 즉 필요한 시스템 데이터를 현장에서 직접 수집하고 개발하고자 하는 응용프로그램에 따라서 서로 다른 형태로 직접 입력하였다. 서로 다른 프로그래머에 의해 개발된 프로그램들은 같은 데이터를 서로 다른 방식으로 다루게 되고, 프로그램 사용자들에게 어떤 일치성을 주기가 힘들다. 데이터베이스 관리시스템은 이런 문제점을 해결하기 도입되었는데 관계형데이터베이스(RDB)는 현실에 존재하는 개체를 정의하고 그들의 관계를 최소한의 의미를 가지는 테이블로 구성하고 그 테이블에 있는 열을 연결한 것이다. 객체지향데이터베이스의 장점은 첫째, 관계형 데이터베이스의 단점을 극복할 수 있다. 즉 가변길이의 데이터를 저장할 수 있고, 사용자 정의의 데이터를 사용 가능하다. 둘째, 객체지향프로그램의 장점을 그대로 적용할 수 있다. 즉 다형성, 상속성, 캡슐화등의 장점을 이용할 수 있고, 객체지향방법으로 구성된 응용프로그램, GUI등과 쉽게 연결이 가능하다. 셋째, 관계형 데이터베이스의 장점을 그대로 구현 가능하다. 본 논문에서는 객체지향 데이터베이스를 설계하고, 전력계통의 단선도를 그리고 해석할 수 있는 GUI를 구축하고, 전력조류계산을 OODB를 이용하여 수행하였다.

2. 본 론

2.1 전력계통해석을 위한 데이터베이스

전력계통해석 프로그램에 사용되는 데이터베이스는 고주파해석, 과도현상해석, 전력조류계산 그리고 고장계산과 같은 다양한 응용프로그램을 지원할 수 있어야한다. 이를 위하여 데이터베이스에 저장되는 데이터는 보다 기본적이고 저수준의 데이터로 모델링 되어야한다. 데이터는 많은 응용프로그램에서 다양한 레벨로 유도될 수 있는 최저 레벨의 데이터로 표현되어야 한다. 응용프로그램의 사용자의 관점에서 본 데이터는 하나의 데이터가 하나 이상의 응용프로그램 사용되어야하고, 하나의 프로그램의 결과가 다른 프로그램에서 이용될 수 있어야한다

다. 따라서 각각의 응용프로그램은 데이터를 읽고 계산 결과를 일관성 있는 포맷으로 저장할 수 있어야 한다. 만약 이러한 계통의 데이터를 텍스트 파일로 저장하여 사용한다면 데이터의 양이 매우 큰 경우는 데이터를 읽는 시간이 상당히 길고, 중복된 데이터가 많아진다. 계통해석의 초창기에는 응용프로그램마다 해석에 필요한 데이터가 특별히 고안된 포맷으로 정의되었고, 초기의 전력계통해석의 기초가 되었다. 이 포맷의 장점은 데이터 접근속도가 빠르다는 것이고 단점은 프로그램에서 사용될 때 하나의 포맷에서 배열형태의 데이터로 사용되기 위해서는 데이터 변환이 있어야 하고, 같은 데이터 파일을 이용하여 다른 응용프로그램을 수행할 수 없다.

2.2 관계형데이터베이스와 객체지향형 데이터베이스.

그림1은 관계형 데이터베이스를 이용한 전력계통 데이터의 표현을 나타낸 것이다. 데이터 베이스가 공학적인 문제에 응용될 때는 사용자는 디스크에 저장된 데이터를 메모리에 불러들이고, 그 데이터를 필요한 시간만큼 사용하고 수정된 데이터를 다시 디스크에 저장하게된다. 전력계통해석 응용프로그램이 주어진 시스템을 해석하기 위해서는 관계형데이터베이스에 저장된 레코드 형태의 데이터를 읽고 해석에 필요한 형태로 컴퓨터 메모리구조에 맞게 데이터를 포인터 형태로 재구성한다.

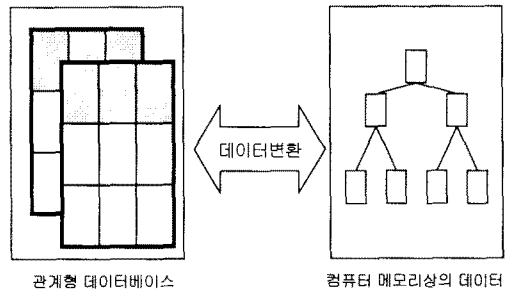
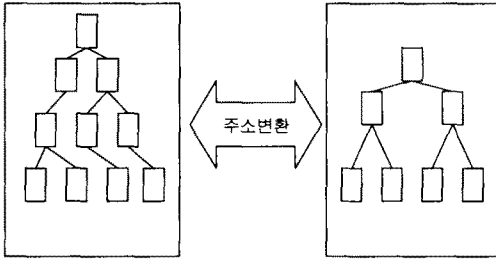


그림 1 관계형데이터베이스의 전력계통 데이터 표현

그림2는 객체지향데이터베이스를 이용한 전력계통 데이터의 표현을 나타낸 것이다. 객체지향 데이터베이스는 데이터의 저장구조를 컴퓨터 메모리상의 데이터 구조와 동일한 트리구조를 가진다. 각 객체들간의 포인터 연결구조에 데이터 베이스에서 유지하고, 메모리에 불러들일 때 이 구조의 포인터 연결만을 재구성하면 된다. 데이터 베이스에서 C언어 자료구조를 그대로 유지하고 있다. 따라서 응용프로그램을 작성하는 사람은 데이터베이스를 하나의 큰 가상메모리처럼 사용할 수 있다.



객체지향 데이터베이스 컴퓨터 메모리의 데이터
 그림 2 객체지향데이터베이스의 전력계통 데이터 표현

2.3 전력계통의 해석

2.3.1 전력계통의 객체지향적 표현

전력계통은 그림 3과 같이 모선, 선로, 모선과 모선사이의 선로에 설치된 장치들, 부하, 모선에 설치된 장치들의 집합으로 표현될 수 있다. 선로에 설치된 장치는 단로기, 차단기, 변압기등이고 모선에는 Shunt 장치를 들 수 있다. 전력계통의 구성요소들을 객체로 모델링하여 이들의 연결관계를 계통의 구성에 맞게 맺어주면 전력계통의 표현이 완성되는 것이다. 객체지향적 모델링의 기본개념에 충실하게 시스템의 실 객체의 독립된 단위를 클래스로 모델링하여야 한다. 따라서 기본적으로 계통요소를 표현하는 데이터는 각 요소를 모델링 하는 객체단위로 분리된다.

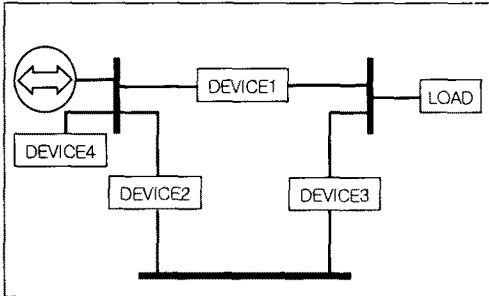


그림 3 전력계통의 구성

그림 4는 모선객체의 구성을 나타낸 것이다. 모선객체는 고유 아이디를 가지고 있으면서 모선에 연결된 가능성이 있는 발전기, 부하, 차단기 그리고 선로객체에 대한 포인터를 가진다. 모선 객체에서는 발전기객체, 부하객체 그리고 차단기 객체를 양방향 포인터 변수로 관리한다. 이런 구조의 장점은 각각의 객체가 독립성을 유지하면서도 상호관계를 유지한다는 것이다. OODB는 이러한 구조를 변경없이 저장하고 관리할 수 있다.

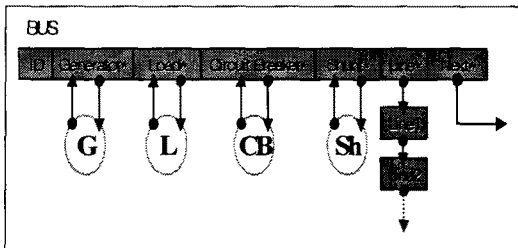


그림 4 모선객체의 구성

그림 5는 선로객체의 구성을 나타낸 것이다. 선로객체는 아이와 모선객체, 변압기 객체를 변수로 가진다. 변수 Bus는 현재의 선로가 연결된 두 개의 모선정보를 포인터로 가진다. 모선과 선로 객체는 양방향 연결구조이다. 변압기가 존재하지 않을 때는 변압기 변수는 널 포인터가 된다. 선로객체도 시스템에 존재하는 다른 선로와 포인터로 연결되기 위한 Next 포인터를 가지고 있다.

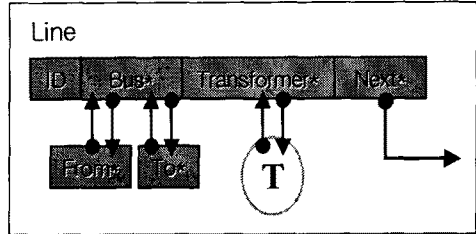


그림 5 선로객체의 구성

2.3.2 전력계통 구성요소의 OODB저장구조

실제된 객체가 OODB에 자신의 데이터를 저장할 수 있는 기능을 가지기 위해서는 각 객체가 객체의 지속성을 가져야한다. 사용자의 클래스가 지속성을 가지기 위해서는 사용자 정의 클래스가 지속기반클래스 Persist에서 파생되는 클래스 구조가 되도록 하고, 지속기반클래스에서 가상함수로 정의되어 있는 readObj()와 writeObj() 함수에 대한 처리루틴을 만들면 된다. 객체지향데이터베이스에서 객체가 지속성을 가진다는 것은 객체를 디스크에 저장하고 다시 불러오는 것을 말하는 데 객체는 생성자가 소멸된 후에도 존재해야되고 객체의 위치는 객체가 생성된 메모리의 주소공간으로부터 디스크로 이동한다. 따라서 객체에게 지속성을 부여하기 위해서는 지속기반클래스를 설계해야된다. 지속기반클래스의 기능은 객체의 데이터 멤버를 디스크에 쓰기, 클래스가 디스크에 쓰여진 데이터 멤버를 읽어서 클래스의 데이터 멤버를 생성하기 그리고 클래스가 데이터 멤버의 크기와 위치를 알 수 있도록 하는 것 등이다.

```

class Persist
{
    virtual writeObj();
    virtual readObj();
    :
};
class Bus: public Persist
{
    data objects;
    void writeObj();
    void readObj();
    :
};
  
```

그림 6 지속기반객체의 상속

2.3.3 전력계통 해석을 위한 MMI구현

마이크로소프트(MS)사에서 제공하는 MFC(Microsoft Foundation Class)는 윈도우 운영체제를 위해 MS사 자체에서 개발된 OOP를 근간으로 하는 윈도우용 C++ 라이브러리로서 윈도우용 프로그램을 개발하기 위한 편리한 클래스들을 제공한다. 본 논문에서는 MFC를 이용하여 전력계통의 온라인 안정도 해석을 위한 MMI를 구현한다. 주요 기능은 주화면에 전력계통을 직접 그림 수 있는 기능과 그려진 시스템에 마우스를 이용한 사건처리

로 시스템 데이터를 직접 입력하고 전력조류 계산을 행하는 기능이다. 메인 메뉴는 크게 파일, 편집, 보기, 그리기, 해석 그리고 창의 메뉴로 구성된다. 파일 메뉴는 새 파일, 열기, 닫기, 저장, 다른 이름으로 저장, 프린터 미리보기 그리고 프린터설정으로 구성된다. 편집 메뉴는 실행취소, 잘라내기, 복사, 삭제, 전체선택 그리고 새 객체삽입으로 구성된다. 보기 메뉴는 안내선, 바탕색, 객체들 보기, 툴바 그리고 상태바로 구성된다.

전력계통을 해석하기 위해서는 우선 GUI에서 사용자가 해석하고자 하는 전력계통의 단선도를 직접 그리고 시스템의 모선, 발전기, 선로, 변압기등의 구성요소들의 데이터를 직접 화면에서 입력해야 한다. 그림 7은 GUI 화면에서 시스템의 발전기 데이터를 입력받기 위한 입력창을 보여준다. 그림 8은 선로데이터를 입력하기 위한 입력창이다. 입력된 데이터는 모두 OODB에 저장되고, 이후 시스템 해석의 모든 데이터는 OODB에서 관리된다. 그림9은 임의의 예제시스템의 전력조류계산 결과를 화면상에 표시한 것이다.

그림 7 발전기데이터 입력창

그림 8 선로데이터 입력창

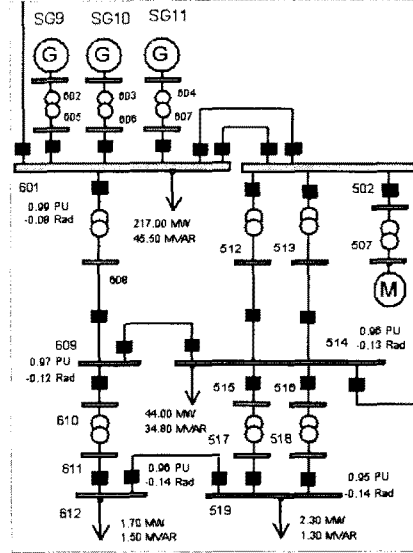


그림 9 전력조류 계산결과

3. 결 론

본 논문에서는 전력계통 해석을 위한 객체지향 소프트웨어를 설계하고, 편리한 그래픽 환경을 구축하여 화면과 데이터베이스를 연동시켜 시스템의 데이터 관리를 효율적으로 할 수 있는 도구를 만들었다. 전력계통과 같이 매우 복잡한 시스템의 해석을 위한 소프트웨어의 개발은 객체지향기법이 가장 좋다. 따라서 이와 연동되는 데이터베이스를 객체지향적으로 설계하여 시스템해석의 효율을 더욱 높였다. OODB는 객체에 관련된 데이터를 하나로 모아서 관리하기 때문에 복잡한 시스템의 데이터 관리에 더욱 효과적이다. 전력계통 해석을 위한 소프트웨어들이 현재의 경향인 객체지향적으로 설계된다면 OODB는 더욱 각광을 받을 것이다.

[참 고 문 헌]

- [1] Harrington, "object-oriented Database Design", Morgan Kaufmann, 2000.
- [2] Scott Danforth, "Integrating Object and Relational Technologies", 1992 IEEE
- [3] Fu Shuti, "An Approach of Object Oriented Database for Open Energy Management System", IEEE TENCON'93
- [4] Jen-Hung Chen, "Industrial Power System Analysis with Database Access", IEEE Trans. on Industry Applications, VOL. 36, NO. 5. 2000
- [5] 황규영, "데이터베이스시스템", 영한출판사, 2002년
- [6] 오명희, "C로 만드는 데이터베이스", 도서출판 삼각형, 1996년
- [7] 김연홍, "데이터베이스 모델링", 프리렉, 2002년