OODB와 GUI를 이용한 전력계통 고장해석

홍영환* 박지호* 김정년** 백영식*
경북대학교* LG전선**

Fault Anaysis of the Power System by using the OODB and GUI

Hong Young Hwan*, Park Ji Ho*, Kim Jung Nyun**, Baek Young Sik*
KyungPook National University* LG Cable Ltd.**

Abstract - Power system is becoming more and more complex and large. And power consumption increases rapidly according to the industry development.

There is an ample possibility for a fault in proportion to these facts. Therefore power system engineer is requested to abilities, fault analysis and fault prevent.

This paper handling result data by Object Oriented Database(OODB) because OODB has advantages abut flexibleness and recycling.

Besides this paper presents fault analysis program by Graphic User Interface(GUI) to understand easily program using method.

1. 서 론

산업이 발달하고 사람들의 생활 패턴도 변화함에 따라 전력의 요구치가 그만큼 증가하게 된다. 그래서 전력의 공급량도 증가하고 전력 계통 시스템이 전체적으로 더 복잡해짐에 따라 사고의 발생도 높아지고 피해 규모도 점점 커진다. 고장해석은 고장시의 전압, 전류를 정량적으로 분석하여 전력계통, 설비의 영향을 고찰하고 적절한 보호방식을 결정하여 사고의 파급을 최소화한다. 뿐만 아니라 고장을 사전에 방지하는 것을 지향한다.

그리고, 계통의 고장해석을 할 수 있도록 소프트웨어를 개발한다. 어떤 소프트웨어를 개발할 때 업무분석에 시간 할당을 70% 정도 한다. 그만큼 만들고자 하는 작업의 특징과 사용자가 바라는 것을 정확하게 반영해야한다. 전력계통해석 프로그램은 전력계통 자체의 복잡성과 해석 프로그램의 복잡함 그리고 사후 프로그램의 유지보수의 관점에서 객체지향방식으로 개발하는 것이 적합하다. 데이터베이스에도 이러한 소프트웨어적인 경향을 반영하는 것이 객체지향 데이터베이스(OODB)이다. OODB는 객체지향적으로 설계된 소프트웨어와 쉽게 결합이 되고 기존의 관계형 DB의 장점도 그대로 수용가능하다.[1]

본 논문에서는 전력계통의 기본 요소들을 모델링하고 고장 해석을 수행하는데 있어서 먼저 모선사고와 선로사고로 구분하고 각 사고별 종류에 따라 전체 모선의 전압과 전류를 구한다. 프로그램 안에서는 서로 연관되는 변수들을 하나의 변수로 묶어 출력되도록 하기 위해 OODB를 적용하였다. 외부적으로는 그래픽 형태로 프로그램을 나타내어 사용자가 좀 더 쉽게 이해하도록 하였다. 전체적인 개념도는 그림1. 과 같다.

2. 본 론

2.1 고장해석 기법

어떤 전력계통의 시스템이 구축되어 있을 때 고장해석의 첫 단계는 사고의 유형을 규정짓는 것이다. 모선사고와 선로사고로 나누고 하나의 고장 전류가 생기면 전체 시스템의 모든 전압과 전류는 달라지므로 새로운 값을 구해야한다.

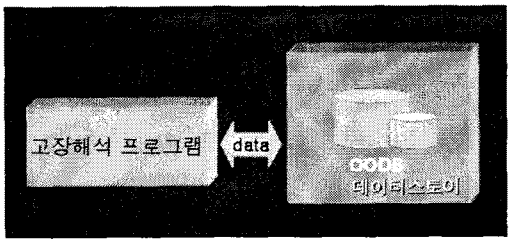

그림 1. 전체 프로그램 개념도
Fig. 1. General idea diagram

선로사고인 경우 그림 2.처럼 선로상의 고장 지점을 중심으로 임피던스의 값이 인접해 있는 모선과의 거리에 비례하므로 새로운 임피던스를 구하기 위해 가상의 새로운 모선을 하나 만든다. 여기서 고장전류 값이 두개 더 추가가 되고 데이터의 테이블로 상에서 행이 더 증가한다.

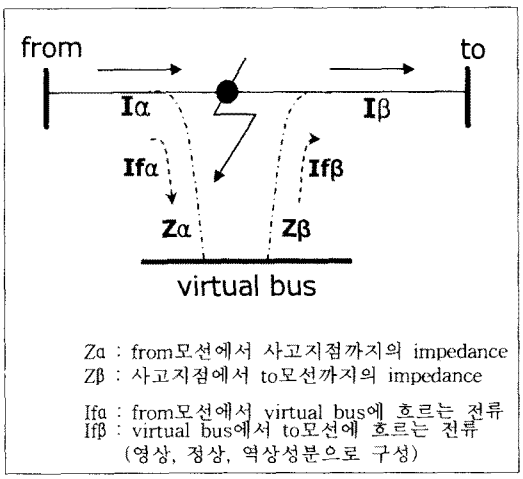

그림 2. 선로사고 지점의 가상 변환
Fig. 2 Virtual transform at line fault point

계통에서 고장해석은 불평형 전류, 전압값을 가진다. 그래서 대칭적인 성분, 즉 영상·정상·역상의 값으로 분해하고 다시 합성하는 대칭좌표법을 사용한다. 본 논문에서도 모선과 선로의 전압, 전류를 sequence, phase로 나타내고 사고의 형태에 따라 각각의 구하는 식을 달리 적용했다. 사고의 형태는 사고의 유형과 구별되는데 사고의 유형은 사고를 모선사고와 선로사고로 나누지만 사고의 형태는 3상 지락사고, 1선 지락사고, 2선 지락사고, 2선

간 사고로 나누고 각각의 전압, 전류 구하는 식이 영상, 정상, 역상마다 다르다.

Impedance는 영상과 정상은 계산해주고 역상은 정상과 같은 값을 사용한다. 각각의 사고형태에 따른 전압과 전류는 다음과 같다.[2,3]

❖ 3상사고

$$I_f^0 = I_f^2 = 0 \quad I_f^1 = \frac{E}{Z_f + Z^1}\left(E = 1, Z_f = 0\right) \tag{1}$$

$$V^0 = V^2 = 0 \quad V^1 = E - \left(Z^1 \times I_f^1\right) \tag{2}$$

❖ 1선 지락사고

$$I_f^0 = I_f^1 = I_f^2 = \frac{E}{\left(Z^0 + Z^1 + Z^1 + 3 * Z_f\right)} \tag{3}$$

$$V^0 = -Z^0 * I_f^0 \qquad V^1 = E - Z^1 * I_f^1$$

$$V^2 = -Z^1 * I_f^2 \tag{4}$$

❖ 선간사고

$$I_f^0 = 0$$

$$I_f^1 = -I_f^2 = \frac{E}{(Z^1 + Z^1 + Z_f)} \tag{5}$$

$$V^0 = 0 \quad V^1 = E - Z^1 * I_f^1 \quad V^2 = -Z^1 * I_f^2 \tag{6}$$

❖ 2선 지락사고

$$I_f^0 = \frac{-Z^1 * E}{(Z^1 * Z^1 + 2 * Z^1 * (3Z_f + Z^0))}$$

$$I_f^1 = \frac{(Z^1 + 3Z_f + Z^0) * E}{(Z^1 * Z^1 + 2 * Z^1 * (3Z_f + Z^0))}$$

$$I_f^2 = \frac{-(3Z_f + Z^0) * E}{(Z^1 * Z^1 + 2 * Z^1 * (3Z_f + Z^0))} \tag{7}$$

$$V^0 = -Z^0 * I_f^0 \qquad V^1 = E - Z^1 * I_f^1$$

$$V^2 = -Z^1 * I_f^2 \tag{8}$$

전체 계산되는 프로그램의 Flow Chart는 그림3.과 같다.

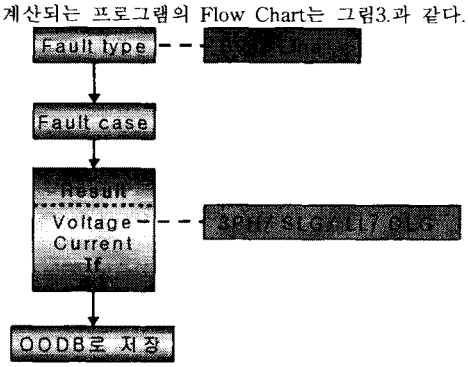

그림 3. 프로그램 순서도
Fig. 3. Program flow chart

위의 그림에서 Bus타입이 되면 Z0와 Z1을 구하고 Line 사고일때는 사고지점(입력되는 값)을 기준으로 임피던스를 그림 2.처럼 나누고 line data는 two line이 추가된다.

2.2 OODB로의 데이터처리

데이터베이스로 설계할 때 현실세계를 반영하는 대상을 실체(Entity)라고 하며 그 실체가 가지는 특징들을 속성(attribute)이라 한다. 하나이상의 실체 사이에는 관계가 이루어져 있고 일대일, 일대다, 다대다의 형태로 나뉜다. 관계형 DB는 객체와 객체 또, 그 사이의 관계를 중심으로 두기 때문에 데이터들에서의 좌우로 신속하게 검색할 수 있고 중복되는 데이터들을 줄일 수가 있다. 데이터베이스 분석을 할 때 객체와 관계의 관리를 쉽게 나타낸 것이 E-R diagram이다[1,4]. 그림 4.은 Bus Entity 하나에 여러 속성들을 나타내고 있으며 이 Entity를 구분짓 수 있는 기본키도 설정되어 있다. 그리고 테이블에 데이터가 저장 될 때는 아래 그림처럼 그려질 수 있다.

고장해석의 데이터들을 Object Oriented Database로 구현하고자 한다면 데이터뿐만 아니라 데이터를 특징짓는 함수까지 처리할 수 있어야한다. 그림 5.는 데이터만 처리했을 때와 데이터와 데이터형을 같이 저장하는 방식을 비교해주는 그림이다. 관계형 데이터베이스는 지정한 칼럼의 수가 고정되어 있어서 유동적이지 못하다. 하지만 OODB로 데이터를 설계하면 필요한 데이터만 처리하기 때문에 속도도 빨라지고 메모리 용량도 줄어든다.[4]

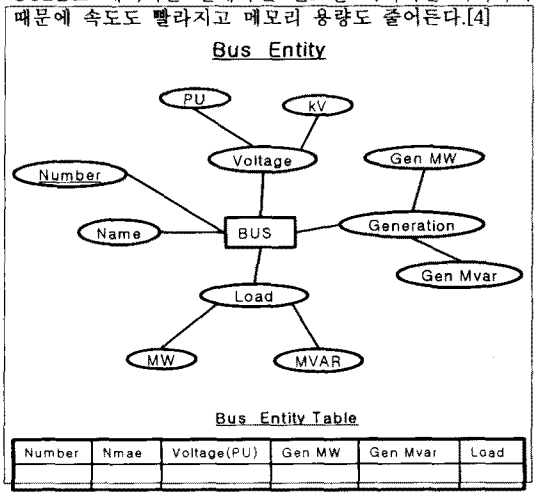

그림 4. E-R diagram과 table 예
Fig. 4. example of the E-R diagram & table

고장해석 프로그램에 그림5.의 OODB개념을 적용하면 사고의 형태에 따라 데이터 구조가 달라질 수 있음을 볼 수 있다. 선로사고일 경우 그림 2.에서 보듯이 임피던스가 고장지점을 중심으로 나누어지고 가상의 모선이 생성되므로 I도 두개 값이 더 추가가 된다. 그러나 모선사고일 경우는 임피던스를 분리하지 않아도 되지 때문에 Zα, Zβ는 불필요한 데이터이다. 즉, 칼럼의 수를 줄일 수 있다.

고장해석을 위한 프로그램에는 상황에 따른 결과값을 계산하는 부분도 있어야 되지만 table에 저장하는 부분도 있어야된다. data저장 설계를 OODB로 하기 때문에 이를 구현하는 명령어들도 따로 만들 수 있다.

사용자가 필요한 데이터들을 WriteObject라는 객체에 등록시켜주고 값을 받을 때는 ReadObjecct에 받는 데이터이름을 입력한다. 그리고 OODB로 데이터를 저장하는 클래스이름을 고장 계산하는 프로그램에 등록시켜준다.

※ : data가 없음.불필요한 메모리

Id	Char형 Data	Double형 Data	Int형 Data
1	A	※	E
2	※	C	※
3	B	D	※

(a) Rational Database형으로 저장

Id	Data	
1	Char형 A	Int형 E
2	Double형 C	
3	Char형 B	Double형 D

(b) Object Oriented Database형으로 저장

그림 5. RDB와 OODB구조 비교
Fig. 5. Comparison of RDB and OODB structure

2.3 전력계통 고장해석을 위한 GUI구현

컴퓨터가 발달함에 따라 프로그램 구축 환경이 GUI를 기반으로 하는 윈도우 환경으로 변해가고 있다. 사용자는 그림이 주는 이점 때문에 좀 더 쉽게 프로그램을 사용할 수가 있다. DOS환경과 비교해볼때 입력되는 값들도 동시에 여러 값들을 쉽게 입력할 수 있고 명령어들이 나타나는 것을 클릭하거나 버튼을 클릭함으로써 작업을 수행할 수 있다. 또 GUI의 이점은 데이터들을 변환할 때도 쉽게 할 수 있다.

구체적인 프로그램 GUI를 만들기 위해 먼저 계통의 기본적인 요소들을 그릴 수 있는 메인 화면(그림 6.)이 있고 각 요소들의 데이터를 입력할 때는 오른쪽 마우스 버튼을 누르면 그림 7.과 같은 데이터 입력창이 뜨도록 만든다.

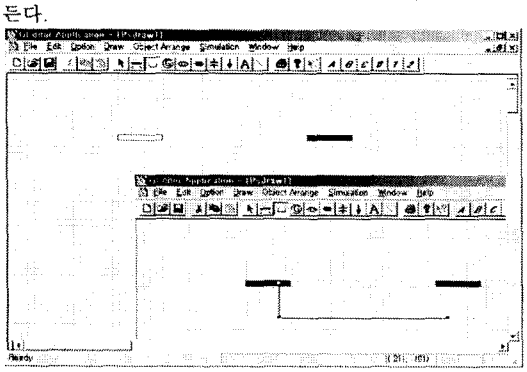

그림 6. 고장해석 프로그램 메인화면
Fig. 6. Main screen of fault analysis

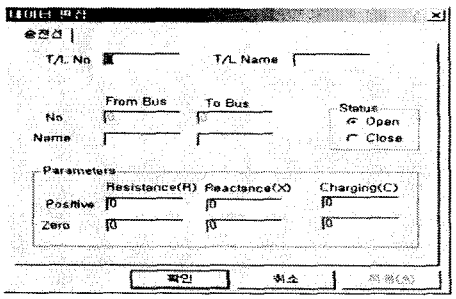

그림 7. 데이터 입력창
Fig. 7. Data input window

고장해석을 시작하면 고장해석의 입력데이터 창이 그림 8.처럼 만들어서 DOS환경보다 좀 더 쉽게 값을 입력하도록 한다.

고장해석이 실행되면 결과값을 화면으로 표현(그림 9.)할 수도 있고 Text파일로도 출력 할 수가 있다.

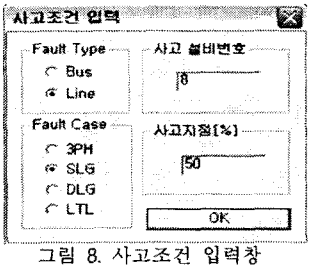

그림 8. 사고조건 입력창
Fig. 8. Fault option input window

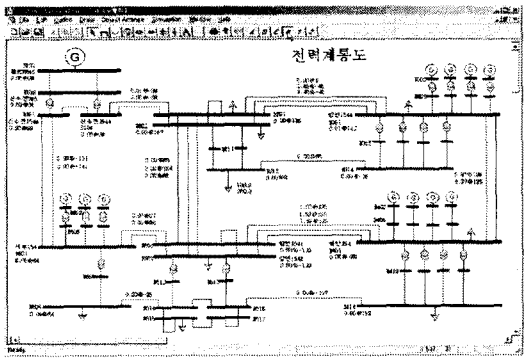

그림 9. 사고조건 입력창
Fig. 9. Fault option input window

3. 결 론

본 논문은 전력계통의 고장해석에 OODB와 GUI를 결합하여 데이터 처리를 기존의 관계형 데이터베이스의 단점을 다음과 같이 개선하였다.

첫째, GUI를 사용하여 사용자가 쉽게 프로그램을 사용할 수 있도록 하였다.

둘째, OODB를 사용하여 데이터 처리의 효율성과 재사용가능성을 높였다.

그리고, OODB의 형태로 저장하는 것은 처음부터 OODB의 형태로 프로그램화 되지 않아도 필요한 결과값들을 선별하여 함수에 등록시켜주고 함수를 수행하는 객체를 고장해석 프로그램에 클래스 등록해주므로 가능하게 한다.

[참 고 문 헌]

[1] 황규영, 홍의경외, "데이터베이스 시스템", 2002
[2] Hadi Saadat, "Power System Analysis", 1999, pp 400-453
[3] J.Duncan Glover, Mulukutla S. Sarma, "Power System Analysis and Design", 2002, pp 396-425
[4] 김연홍, 우성미, 문택근, "데이터베이스 모델링", 2002
[5] 이상엽, "Visual C++ Programing Bible ver. 6.x", 2003
[6] Nutakki D. Rao, "Advanced Applications of a Relational Database", IEEE Transaction on Power System, Vol.5, No.1, Feb. 1990, pp 338-345
[7] M. Foley, A. Bose, W. Mitchell and A. Faustini, "An Object Based Graphical user Interface for Power System", IEEE Transactions on Power System, Vol. 8, No.1, Feb. 1993, pp 97-104