

조류계산을 위한 사용자 정의 모델 지향형 그래픽 시뮬레이터 개발

황인준, 김건중, 전동훈, 박현신 신만철, 오성균
 충남대학교 전기공학과 (주)파워이십일

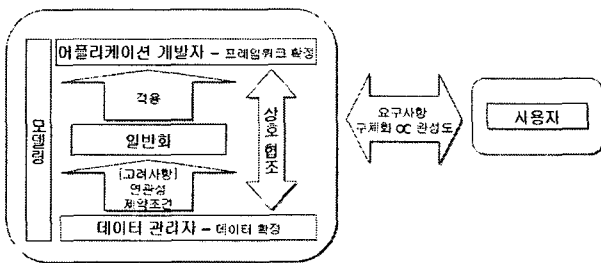
A development of graphic simulator for power flow analysis of user define type

Hwang In Jun, Kim Kun Joong, Jeon Dong Hoon, Park Hyun Shin Sin Man Chul, Oh Sung Kyun
 ChungNam National Univ. Power21 Corp.

Abstract - 현대 과학 기술은 개개의 분야의 고유성과 특성이 점차 모호해지고 여타 분야와의 접촉 내지는 응용의 발판을 넓혀 점차 실용화 하는 추세이다. 특히 경제성과 밀접한 관련이 있는 엔지니어링 분야가 더욱 그러하다. 일상생활에서 사용되는 컴퓨터의 진보는 그간 학문적인 연구에 소요되었던 시간을 단축시켰으며 연산가능 범위를 확대하였고 이것을 바탕으로 우리는 좀더 세밀하고 정확한 진단과 예측이 가능해졌다. 이는 엔지니어들이 갖추어야 할 지식체계에도 영향을 주어 과거 배타적이며 폐쇄적인 업무 성향을 개방적이고 공유 가능한 대상으로 변모가능하게 되었다는 점에서 의의가 크다. 특히 정보 전달의 측면에서 보면 최근 부각되고 있는 그래픽적 요소를 가미한 해석프로그램을 주목해야만 한다. 일반적인 데이터 나열이나 조작 시에 사용되던 텍스트 기법이 다이얼로그 상자를 통해 좀더 인터랙티브한 환경으로 바뀌었고 현재는 이러한 내용을 그래픽적인 환경으로 재해석하고자 하고 있다. 바로 이러한 노력의 일환으로 본 논문 역시 그래픽 시뮬레이터 개발에 관해 다루게 되었다. 본 논문에서는 그래픽이란 주제를 좀더 심층적으로 다루어 사용자 정의 타입의 단계까지 논함으로써 앞으로 나아가야 할 방향을 제시해 보고자 한다.

1. 서 론

그래픽 시뮬레이터 설계 시에는 데이터 관리 영역과 이를 바탕으로 한 어플리케이션 두 가지 측면에서 접근해야 한다. 엔지니어링의 경우 모델의 정식화 내지는 모델링이 해석 전반에 영향을 주므로 더욱 중요하다. 이는 최초 설계 시에 충분한 조사와 검증의 단계가 선행되어야 함을 뜻한다. 데이터 관리자는 문제를 일반화하는 과정에서 적용 범위와 데이터간 상호 연관관계 및 제약조건 등을 고려해야 한다. 좀더 명확하고 분명한 범주 안에서만이 개발 의도와 방향을 주도할 수 있기 때문이다. 어플리케이션 개발자는 사용자의 요구사항 등을 수렴하여 프레임워크를 디자인하고 피드백과 조율을 거쳐 좀더 사용자에게 친숙한 환경을 구축해야만 한다. 그러나 실제 사용자는 자신의 요구사항에 대해 구체적으로 생각하고 있지는 못하며 대개 방법이나 정책상에 의견만을 갖고 있다. 그러므로 본격적인 개발 이전에 피드백이 중요한 이유는 서로 간에 협의와 이해가 이루어지고 차후 프로그램의 만족도와 완성도에 상관관계가 있기 때문이다. 물론 데이터 관리자와 어플리케이션 개발자 양자는 병행하여 진행할 수도 있으나 서로 간에 이해도 또한 내부적으로 중요하다.

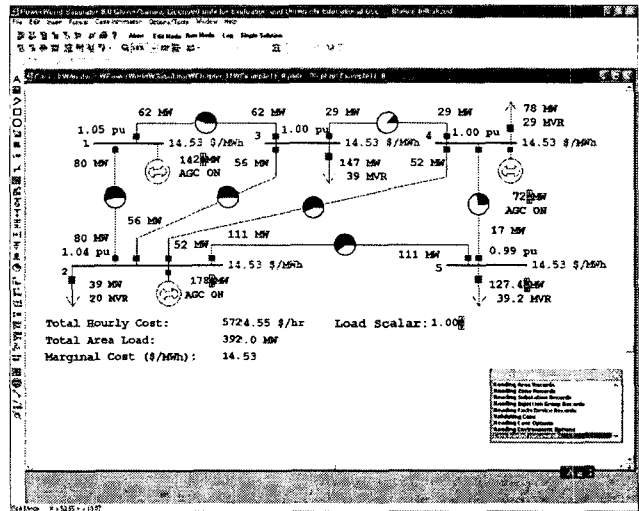


〈그림 1〉 시뮬레이터 설계 시 고려해야 할 과정

이러한 설계과정에서 참고 할 것이 바로 상용 프로그램 분석이다. 기존 상용 프로그램의 기능들과 구현 수준을 바탕으로 현재 진행하는 개발 시뮬레이터의 기본 기능 등을 정의하고 결과 데이터를 어떻게 보여줄지 혹은 앞서 정의한 모델링과의 차이점을 통해 추가로 필요한 정보가 무엇인지에 대한 조사활동이 필요하다. 이로부터 완성도와 성능을 비교가능하게 된다. 검증 시에는 실제 상용 프로그램을 우리가 정의한 문제에 적용해보고 동일한 조건에서 실시하며 이에 대한 차이점이 바로 프로그램의 장점이자 특징이 된다.

이상의 내용을 전제로 본론에서는 그래픽 시뮬레이터의 특징 및 장점, 구성 그리고 사용자정의 기능의 필요성과 적용에 관해 살펴보고 그러한 기능을 프로그램에 통합하고 최종적으로 조류 계산에 적용해보도록 하겠다.

1) 요구사항에서 발생하는 예외유형으로든 틀린 요구, 일관되지 않은 요구 혹은 호환성이 없는 요구, 불명확한 요구, 대상 범위 이외의 요구, 빠진 것이 있고, 불완전한 요구와 부적당한 요구 그리고 Type 오류 등이 있다.

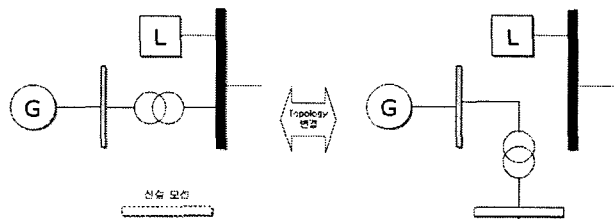


〈그림 2〉 상용 프로그램 예 - PowerWorld Simulator

2. 본 론

2.1 그래픽 시뮬레이터의 특징 및 장점

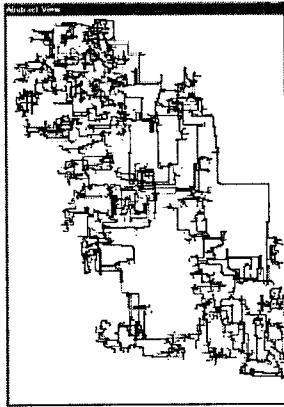
그래픽을 사용 시 다음의 특징과 장점을 얻을 수 있다. 첫 번째로 개념을 실제화 하여 관계를 파악하기 쉽다. 가령 변전소를 신설해 계통에 추가하는 경우를 모델링한다고 하자. 계통에 변전소는 모선과 같은 역할을 담당하고 있다. 즉 변전소를 신설할 경우는 주변 변전소로부터 도플로지가 수정 및 변경되어 인접 변전소에서 신설 측으로 반영이 되어야 한다. 이때 변압기와 기타 설비가 이동(Move) 또는 분리(Split)한다고 한다면, 기존에 대화창이나 텍스트 방식의 경우 각각의 정보들이 종류별로 산재해 있기 때문에 전체적인 연결 관계를 이미 알고 있지 않은 경우는 각각 해당 분류로부터 찾아야만 한다. 모선이나 기타 설비들에 넘버링 규칙에 경우 현재는 미리 약속으로 일정 번호 대를 발전모선으로 지정한다든지 혹은 지역적으로 분류 한다든지 등과 같은 방법을 사용하여 왔다. 이러한 노력은 존재해 있는 리스트 중에서 인의의 설비를 빨리 찾기 위한 노력을 반영한 것이다. 그러나 그래픽적인 계통관계를 그릴 수 있다면 위와 같은 내용을 파악하는데 드는 노력을 최소화 할 수 있다. 실제 배치에 관련된 내용을 수정할 때 계통도를 보고 데이터를 찾아서 수정하는 2단계 작업이 아닌 바로 그래픽 상에서 보고 수정하는 방법이 가능하다. 따라서 사용자는 계통에 구성이나 상태를 파악하기 위한 노력을 줄이고 좀더 효율적인 네트워크 모델을 구성하도록 집중이 가능하게 된다.



〈그림 3〉 토폴로지 변경 예

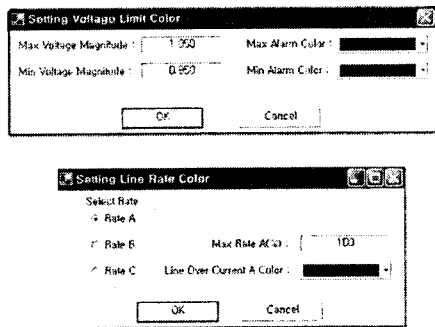
두 번째로는 분명하고 효과적인 정보전달을 통한 시스템 통합이다. 먼저 시스템 통합의 범위를 분명히 해야 할 것이다. 여기서 말하는 시스템이란 네트워크를 이루는 설비들을 통칭하여 일컫는 것이므로 서브시스템과 같은 소규모로부터 실제계를 모두 적용할 수 있다. 먼저 서브시스템의 경우를 보

면 이는 부분적인 해석능리를 검증해 보고자 구성하는 예제 계통으로 모델링된 장치들의 종류에 비례하여 설비들을 구성할 수 있다. 무엇보다 모션하나하나 사용자가 만들어서 구성하여야 하므로 계통의 안정성을 판별하는 기능이 필요하다. 이전의 경우 파일 기반으로 이러한 작업을 진행하였으므로 프로그램에서 파일을 읽어와 일일이 세부 내용과 연관관계를 지정해주고 저장해주어야만 했다. 그러나 필요설비를 그래픽적으로 모델링해 적용한다면 화면에 보이는 내용이 바로 구성도가 되고 나아가 이를 저장 시 이전처럼 매뉴얼 방식이 아닌 자동화 방식을 통해 데이터를 별도로 주작할 필요가 없다. 즉 데이터 관리 측면에서 사용자가 내부적인 처리를 고려하지 않아도 된다. 전체 계통 시스템의 경우 신설과 변경, 삭제 등을 통해 기존 구성을 편집하는 동작을 필요로 한다. 기존 방식의 경우 이러한 조작을 수행 후 해석기능을 통해 변경된 결과를 선택 모션에 한해 제한적으로 볼 수 밖에 없었다. 그러나 그래픽 시뮬레이터에서는 인접하는 부근에 설비를 포함한 계통 전체까지 한번에 볼 수 있으므로 모의 하고자 하는 내용을 더욱 다양한 측면에서 분석할 수 있다.



〈그림 4〉 계통 전체보기 예

세 번째로는 앞서 말한 실제화하고 시스템통합에 적용하는 가운데 나타나는 특징으로 상태변수를 들 수 있겠다. 현재는 해당 설비가 서비스 상태인지 아닌지만을 이진 형식으로 구분하고 있으나 그래픽 요소를 가미하면 생생한 그러한 내용을 대신할 뿐만 아니라 최대/최소값 체크와 같은 유효성도 고려할 수 있다.



〈그림 5〉 속성과 연계된 리미트 체크 기능 예

2.2 그래픽 시뮬레이터의 구성

본 그래픽 시뮬레이터는 일반적인 비즈니스 계열의 내용이 아닌 엔지니어링 해석을 기본으로 한다. 따라서 사용자가 다루는 어플리케이션의 프레임워크 외에 연산 알고리즘이 추가되는데 좀더 세분화 하여 살펴보면 다음과 같은 모듈들을 필요로 한다.

$$X(x, y) \rightarrow T(X), X'(x', y') = (x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta)$$

∴ Linear transformation $T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$

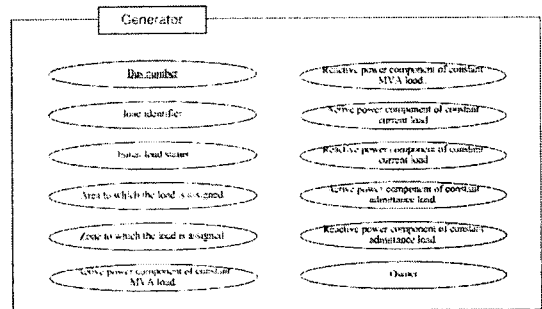
$$T(X) = X'$$

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

〈그림 6〉 그래픽 모듈 회전 예

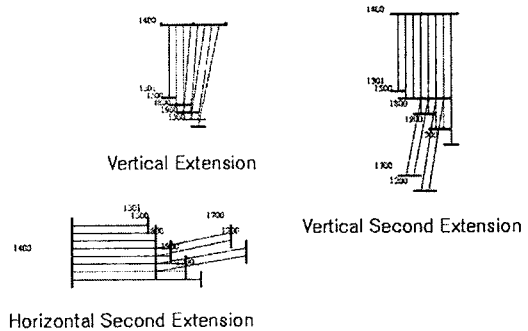
그래픽 모듈의 경우 계통 그리기가 가능해야 한다. 이는 일반 그리기 기능의 모듈로부터 확장하여 얻을 수 있다. 가령 그리기의 기본적인 도형인 사각형, 원, 선, 점등을 조합해 네트워크 시스템에 사용되는 기기들과 장치

들을 형상화하는 작업이 필요하다. 물론 사용자 정의 기능에서도 필요한 기능이며 사용자는 그러한 기본 기능으로부터 그래픽 모델링을 하게 된다. 계통 설비들의 경우는 네트워크 상에 연산 데이터를 부가정보로 갖게 된다. 입력 파일들로부터 부여된 값들을 보여주고 편집 가능하도록 해야 한다. 그리고 각각의 모듈에 대한 기본동작을 제어할 수 있는 기능들을 포함해야 한다. 모션의 경우 모션이동, 모션분리, 모션결합 등의 토폴로지 특성을 갖는다. 동일하게 선로, 발전기, 부하, 선트, 변압기 등도 각 특성을 정의해 부여해준다.



〈그림 7〉 그래픽 데이터의 계산데이터 속성 - 발전기 예

또한 그래픽 모듈의 경우는 모든 조작이 마우스를 통해 이뤄지므로 사용자 편의 기능을 도울 수 있으면 더욱 편리하겠다.

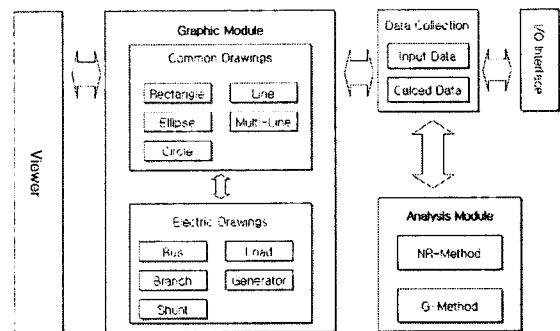


〈그림 8〉 사용자 편의 기능 - 모션의 확장 예

이러한 그래픽화가 가능하도록 하기 위해서는 데이터 관리측면도 고려해야 한다. 데이터의 구조는 그래픽 모듈을 저장하는 부분과 계산 데이터를 저장하는 부분으로 나누어 생각할 수 있다. 전자와 후자는 하나의 통합 파일로 관리하거나 나누어 관리하는 방법이 있겠다. 어느 것이든 간에 입출력 모듈을 통해서 하나의 모델을 구성하는 요소들로 이루어져야 한다.

그리고 해석을 위한 모듈이 필요하다. 본 논문에서는 조류계산을 위한 일반적인 해석 기법으로 뉴턴-랩슨 방법과 가우스-사이드 방법을 적용하겠다. 이들 방법의 구체적인 알고리즘 전개는 본 논문에서 제외하겠다. 그러나 일반적으로 요구하는 해석의 요건은 수렴성과 속도에 초점을 맞추어야 한다. 수렴성은 주어진 데이터가 원하는 제어범위에 들도록 하는 제어변수들과도 관련이 깊으며 이를 세분화하는 것은 계통에 그래픽 모델링과 상관관계가 있다. 또한 수많은 반복적인 연산을 수행할 경우 결과를 확인하는 시간이 바로 프로그램 신뢰도와 직결된다.

마지막으로 뷰어가 필요하다. 뷰어는 전체 그래픽 모듈을 하나의 화면으로 보여주는 일종의 캔버스 역할을 하게 된다.



〈그림 9〉 내부 모듈 관계도