

그래픽 인터페이스를 통한 조류계산구현 II

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

The Realization of Load Flow Under Graphics Interface II

Hawing In Jun, Kim Kun Joong, Park Hyun Shin, Shin Man Chul, Oh Sung Kyun
 ChungNam National Univ. Power21 Corp.

Abstract - In this paper, our goal is program's completion which has graphical user interface. During that process we will implement graphic module and apply it for network devices. We will construct integration system between embedded modules and window's components. In conclusion this paper explain you how to make graphical simulator in electrical engineering.

또한 모선과 모선은 선로로 연결된다. 이전 논문에서 그리기 요소로부터 기본 개체를 설계하고 이들의 동작 특성을 정의 하였으므로 이번 논문에서는 이를 응용한 실제 계통 구성 설비들을 아래와 같이 구현해 보았다.

1. 서 론

이전 논문 "그래픽 인터페이스를 통한 조류계산구현 I"에서 개략적인 구현 단계에 대해 언급하였다. 사용 프로그램을 분석해보고 그것들을 구현하는데 있어서 필요한 요소인 그래픽 컴포넌트들의 속성들 그리고 구현방향 등이 그것이다.

이번 논문에서는 좀더 심화된 내용으로 그래픽 모듈에 대해 언급하고 이것을 조류계산에 적용하여 볼 것이다. 또한 사용자 화면과 관련하여 전체 화면을 설계하고 디자인하여 그래픽 프로그램으로 최종 통합할 것이다.

2. 본 론

2.1 그래픽 모듈 개발

그래픽적 요소는 일반적인 명령어 방식의 개념을 시각화하여 현실적이고 직관적인 정보전달을 목표로 하는데 효과적이다. 따라서 맑고 표현해야만 하는 어떤 사물 혹은 기능을 부가적인 설명이 필요 없이 개인 스스로가 상당부분 이해하고 예측이 가능한 환경으로 만들 수 있다. 바로 이점이 그래픽적 요소를 도입해야만 하는 이유이다. 특히 엔지니어링 소프트웨어의 경우 주어진 문제에 대한 해석결과는 수치적인 연산결과를 바탕으로 하므로 비전문가가 보기에는 알 수 없는 수들의 연속으로 생각하기 쉽다. 하지만 사실은 그 속에서 규칙이나 특성을 분석하고 이를 제어할 수 있도록 해를 찾아가는 일련의 과정을 도출해야만 하는 경우이므로 더욱 그래픽적 요소를 도입해야만 한다.

계통해석 프로그램은 $Ax=B$ 라는 연산에서 x 를 구하기 위한 행렬연산을 수없이 반복하게 된다. 이때 네트워크 방정식을 구성하는 요소들을 실제범위로 확장해 보면 연산에 사용되는 입력 변수들은 상당히 많을 것이다. 이러한 것들을 임의이 조직화하여 나타낸다는 것은 쉽지 않은 일이다. 따라서 그래픽 환경에서는 이러한 입력과 일 동을 네트워크 다이어그램으로 대체해 표시하고 각 해당 장치들을 심비로 상징화하여 속성들을 표시하고자 한다.

2.1.1 네트워크 모듈 개발

계통을 구성하는 설비들은 모선을 중심으로 배치된다.

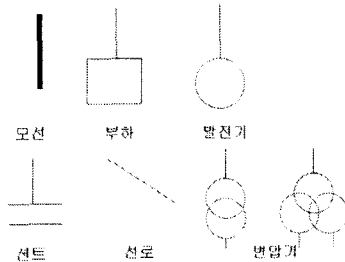


그림 1 네트워크 설비들

그림과 같이 실제 네트워크 모듈은 의외로 단순한 그리기적 요소들의 조합이다. 단지 이들을 그룹으로 표현한 것에 지나지 않는다고 보아도 무방하다. 그러나 계통에 매핑 되는 특정 장치의 데이터 항목들을 속성으로 포함하므로 실제 계산에 사용이 가능하다는 점이 일반 그리기 개체들과 다르다.

참고로 각 계산 속성 데이터는 현재 PSS/E의 *.raw파일용 모델로 하였다.

또한 그리기 속성들로 색상과 경계선 등의 스타일을 설정할 수 있으므로 상태체크를 별도로 필요로 하지 않고 이상의 속성들의 변화로 알림기능이 가능하다.

2.1.2 네트워크 모듈의 고려 요소

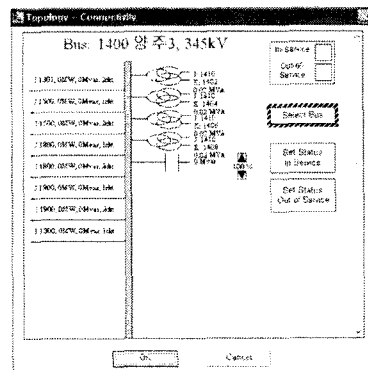


그림 2 Topology - Dis/Connection

네트워크 모델의 경우 실제 배치를 해보면 중심 모선을 기준으로 위치한다. 따라서 각 장치 요소간 배치도 고려해주어야 할 사항이다. 화면에서 1픽셀의 오차도 커다란 오류처럼 민감하게 보이기 때문이다. 그리고 모선 토폴로지 등과 같이 모선 변경시에도 위치 정보를 고려해 주어야 한다.

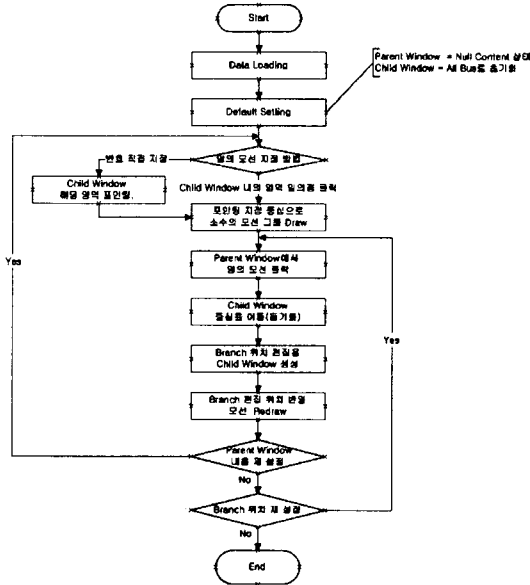


그림 3 권역별 인접 모선 보기 순서도

또한 그리기 편집시마다 모선과 연결된 세부 장치들의 계산 데이터에도 영향을 주도록 해야 한다. 왜냐하면 대부분의 네트워크 설비들은 자신과 연결된 모선 번호를 기준 정보로 설정하고 있으므로 만약 모선 변경시 모선 번호 속성을 변경하지 않으면 그림과 실제 데이터가 불일치하므로 문제가 발생한다. 대표적으로 서브시스템을 구성하는 경우가 그러하다.

2.2 그래픽 모듈의 조류계산에 적용

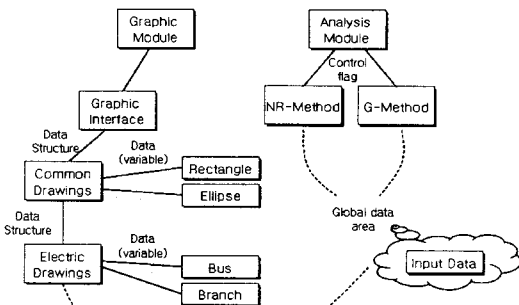


그림 4 그래픽/계산 모듈 결합도

위 그림과 같이 지금까지는 그래픽 관련 모듈과 계산 관련 모듈을 서로 분리하여 생각했다. 그러나 조류계산을 그래픽적으로 표현하기 위해서는 계산 모듈로부터 산출된 결과를 그래픽 모듈로 넘겨주는 연계 과정이 필요하다.

그래픽 조류계산이란 정의를 먼저 내려보자. 물리적인 계통데이터를 기반으로 한다고 생각했을 때 역시 이를 반영하는 뷰어를 그래픽으로 처리한다는 결론이 나온다. 따라서 *raw, 파일과 같은 데이터를 네트워크 다이어그램으로 표시할 수 있어야 하고 연산 수행 후 결과를 반영하는 기능이 필요하다. 조류계산 결과는 조류의 흐름과 방향 그리고 설정한계를 넘어선 오버플로 등을 체크하는 기능을 갖추고 있어야 한다. 이러한 메시지에 대한 알람은 개체의 스타일 변화로써 표시 가능해야 한다.

또한 모선 편집 시 데이터 변경이 화면에 동기화 되어야 하고 서브시스템을 갖추어 볼 수 있는 기능 등도 필요하다.

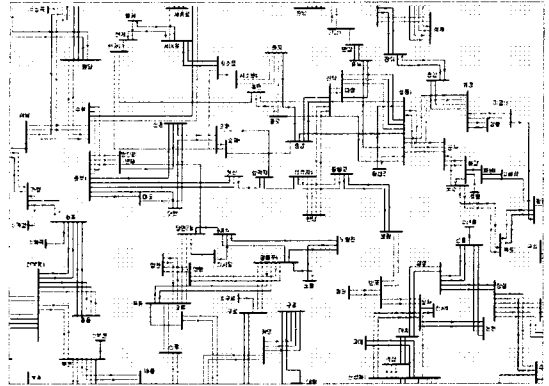


그림 5 실제용을 적용해 조류계산을 반영한 예

2.3 사용자 화면 설계

사용자 화면을 설계하자면 어떤 정보를 어떻게 보여줄지에 대한 고려를 해봐야 한다. 그리고 거기에 개발자가 어떻게 반영하겠다는 정책을 더해 프로그램이 만들어진다.

따라서 본 프로그램은 아래 그림과 같이 만들어 보자 한다. 우선 메뉴와 툴바를 상단에 배치하고 이를 통해 명령어나 기능 등을 조작하도록 한다. 그리고 중앙에는 네트워크 다이어그램을 그리기 위한 캔버스 역할의 드로잉 시트를 배치할 것이다. 또한 각종 뷰어를 두어 이들을 통해 사용자 편의의 정보를 돕도록 할 것이다.

예를 들어 프로젝트 기능을 두어 2분화된 그래픽 데이터와 계통 데이터를 관리하도록 하고 더불어 연산 결과와 이를 반영하는 드로잉 시트의 저장 및 불러오기 가능한 환경을 만들도록 한다. 불러온 계통데이터를 모션중심으로 조작적으로 보여주기 위한 뷰어로서 버스트리스트 창을 두어 활용하도록 한다.

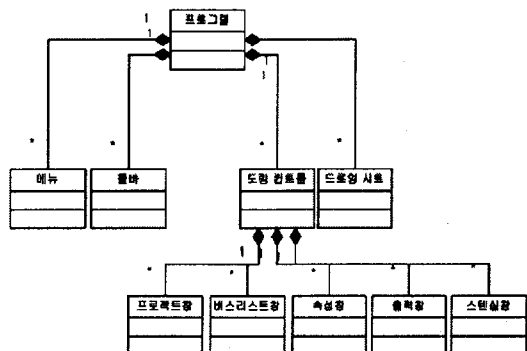


그림 6 사용자 인터페이스 구성도

그리고 그래픽 요소를 선택 혹은 스타일을 편집하기 위한 속성뷰어로서 프라퍼티창을 만들도록 한다. 계산 결과나 프로그램 전반적인 메시징의 출력을 위한 아웃풋 창도 필요하다. 또한 사용자는 자신이 정의한 모델을 등록하고 이를 서브시스템에 사용함으로써 다양한 사용자 정의 모델을 지원하도록 하는 스텐실창을 두도록 할 것이다. 마지막으로 사용자가 조작한 내용을 기록하는 히스토리 창을 두어 동일명령어 반복시 선택만으로 재실행하는 동작을 부여할 것이다.

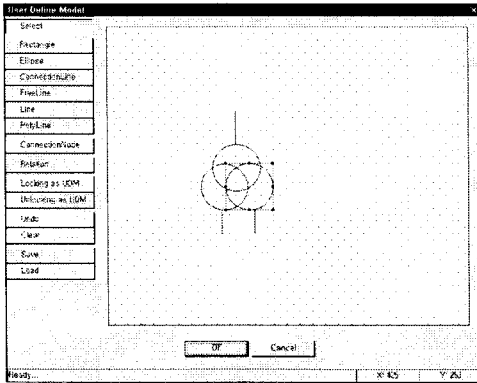


그림 7 사용자 정의 모델러-변압기 생성 예

2.4 전체 프로그램 통합
시스템 통합은 아래와 같은 계층적 구조를 이해해야만 한다.

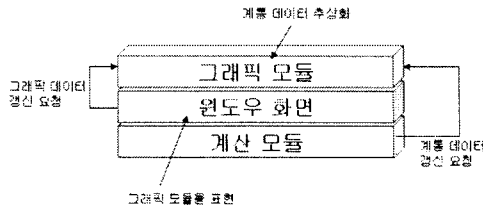


그림 8 내부모듈과 UI의 관계

계통 데이터 표현을 중심으로 볼 때 최하층 부는 역시 실제 계산 데이터가 존재하고 이 모듈은 계속적인 연산을 담당하면서 데이터를 갱신한다. 그다음에 사용자에게 편집 및 뷰어의 역할을 하는 윈도우 인터페이스이다. 여기에 위에서 설계했던 컨트롤과 창들이 배치된다. 또한 윈도우 인터페이스는 그래픽 모듈을 표현하는 캔버스 역할을 담당한다. 마지막으로 그래픽 모듈은 그리기 요청이 캔버스로부터 요청될 때 마다 윈도우의 그리기 자원으로 자신을 그리게 된다.

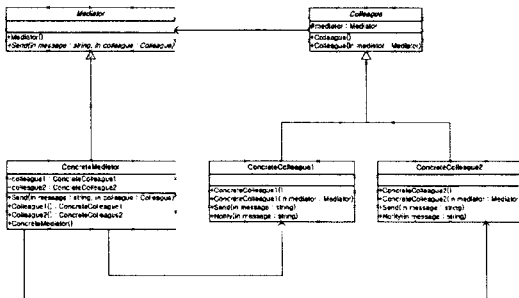


그림 9 시스템 통합 연결자 패턴

성능면에서 화면 처리 요구사항 즉 그래픽 처리기능이 많아질 경우 중간 계층인 윈도우 화면에서는 퍼포먼스 코스트가 많이 요구된다. 이는 사용자의 대부분의 그래픽 편집동작이 마우스 이벤트에서 발생한다고 볼 때 그리기 경우에 수가 많아질수록 물리적으로 요구되는 구원량이 한곳에 집중되는 구조적 비대칭상을 초래하기 때문이다. 따라서 이를 적절히 분산함으로써 성능향상을 꾀하는 구조로 변경해야만 하는 이유가 여기에 있는 것이다. 위 그림 9가 바로 그러한 내용을 설명하고 있다. 요약하자면 그래픽 모듈과 윈도우 화면과의 중간에 층을 하나 두어 중간 매개체 내지는 분산자 역할을 하도록 하는 것이다.

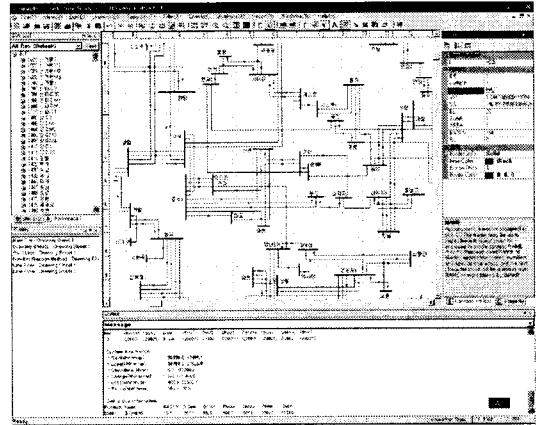


그림 10 메인 화면

3. 결 론

본 논문은 이전에 "그래픽 인터페이스를 통한 조류계산구현 I"에서 개발된 그래픽 기본 컴포넌트와 계산 모듈을 확장 사용하여 그림10과 같이 하나의 어플리케이션을 완성하였다. 그 과정으로 그래픽 모듈을 네트워크 타입으로 적용해 보았고 이로부터 정의된 설비 심벌들과 계산 모듈을 연계시켜 조류계산후 계통의 전체도를 나타내어 보았다. 여기에 윈도우 환경의 사용자 화면을 설계하고 전체프로그램에 적용하였으며 이때 내부적인 모듈간 상호 연계성과 균형적인 조직화가 수반되어야함을 언급하였다.

차후 본 프로그램은 실무 적합성을 검증하고 이에 대한 피드백을 기반으로 기능 추가 등의 개선을 해야만 한다. 따라서 이에 대한 연구가 계속되어야 하겠다.

[참 고 문 헌]

- [1] 신만철, 김건중, 엄제선, 이병일, 박철우, 장경철, "Graphic-based Power System Simulator 소개", 대한전기학회 전력기술부문의 추계학술대회 논문집, p.133-136, 2001년 11월
- [2] 신만철, 김건중, 박철우, 박원경, 이병일, "Graphic-based Power System Simulator의 대규모 전력계통 적용에 관한 연구", 대한전기학회 하계학술대회 논문집, p.50-52, 2002
- [3] 황인준, 김건중, 김규왕, 신만철, 오성균, "그래픽 인터페이스를 통한 조류계산 설계", 대한전기학회 하계학술대회 논문집, p.126-128, 2004
- [4] Eric White, "GDI+ Programming", Wrox, 2003
- [5] 황인준, 김건중, 김규왕, 신만철, 오성균, "그래픽 인터페이스를 통한 조류계산구현 I", 대한전기학회 추계학술대회 논문집, p.220-222, 2004
- [6] ROGER S. PRESSMAN "SOFTWARE ENGINEERING", McGraw-Hill, 2003