

그래픽 인터페이스를 통한 조류계산 구현 I

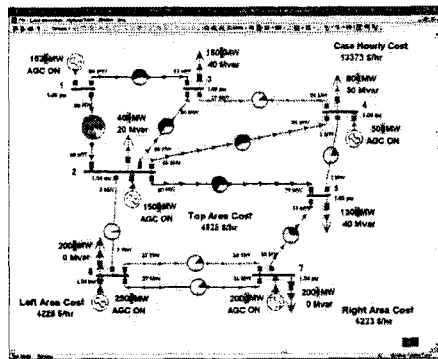
황인준, 김건중, 김규왕 신만철, 오성균
 충남대학교 (주)파워이십일

The realization of Load Flow under Graphic Interface I

Hwang In Jun, Kim Kun Joong, Kim Kyu Wang Shin Man Cheol, O Sung Kun
 Chungnam National Univ. Power21 Corp.

Abstract - In the paper architecture of load flow under graphic interface we mentioned that the define of graphic interface and the way of system modeling. In the view point of engineering, domain define is the step of process which is seek out the general solution. This is a base of program's structure. When making a relationship inner component, relevance and restriction are affected by data type and etc. In this process logical data is realized as a symbol. And system configuration decide that main function and extension by the analysis user's demand and requirements. Specially engineering analysis software has accuracy also through the numerical method. To represent this we need more powerful graphical component. This helps user's accessibility. We can combine IO, numerical library and graphical component as a system element by domain definition. In this paper we will materialize as a step of implementation and decide a direction of programming. In conclusion we will see the analysis software's necessary function to make a better.

전력에 관한 해석 소프트웨어의 현 추세는 강력한 그래픽 기반을 바탕으로 사용자와 인터랙티브한 환경을 구축하는 것이다. 예를 들어 발전기의 동작이나 조류의 흐름을 텍스트상의 수치 데이터만으로 보는 것에서 실시간 확인 가능하도록 한 것이 그 예이다.

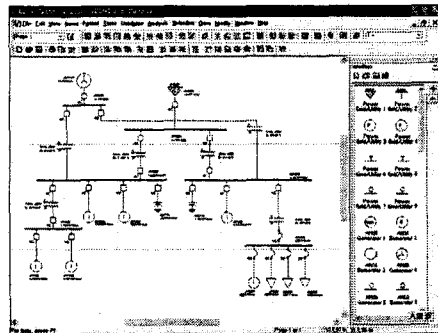


[그림 1] 샘플계통 - PowerWorld 10.0

1. 서 론

그래픽 인터페이스를 통한 조류계산 설계논문에서 그래픽 인터페이스에 대한 정의와 시스템 모델링에 대한 방법론을 살펴보았다. 도메인 분석은 엔지니어링 관점에서 볼 때 제너럴 솔루션(General Solution)을 찾기 위한 과정이며 이를 바탕으로 프로그램의 전반적인 구조를 조직하게 된다. 구성 요소간 관계 조직 시 연관과 제약사항을 분석해 데이터의 타입과 기타 설정을 결정짓게 된다. 이 과정에서 논리적 데이터를 실재화 하는 심벌을 모델링하게 된다. 시스템구성에서는 계획 및 요구사항을 분석하여 주요 기능을 정의함과 아울러 앞으로의 확장성까지 고려하는 단계이다. 특히 계산을 정확성을 기본 바탕으로 하고 있으므로 계산 모듈 즉 수치해석적인 방법이 사용되고 기타의 시나리오 기능을 통해 구체적인 동작상의 기능을 정의하게 된다. 이를 표현하기 위해서 바로 그래픽적인 요소가 가미되어 사용자의 접근성을 돕게 해야 한다. 즉 시스템 구성요소로서 데이터 관련 및 입출력 인터페이스와 수치해석 라이브러리 그리고 그래픽 컴포넌트 등을 도메인 분석 시 작성한 시나리오에 맞게 사용자 인터페이스로 통합 처리하게 된다.

그래픽적인 처리를 응용하면 각 세부 장치의 상태를 원클릭만으로도 핸들링이 가능하다. 특히 차단기나 기타의 장치를 모션에 투입해 활성화 혹은 비활성화 등의 동작을 하는 경우가 대표적이다.



[그림 2] 샘플계통 - EDSA

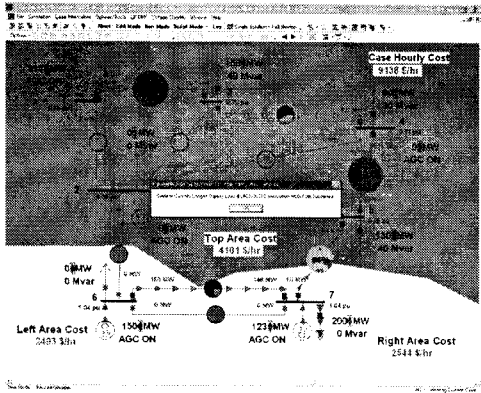
본 논문에서는 이를 더욱 구체화하는 구현의 단계로서 기존 상용 툴들을 분석하고 구현방향을 설정해 실제 구현을 해보고자 한다. 또한 결론으로 해석 소프트 및 기타 범용 적으로 확장 가능한 프로그램이 갖추어야 할 기능으로서의 개선의 여지 등을 알아보려고 한다.

그래픽적 요소를 통하면 [그림 1]과 [그림 2]에서와 같이 실제계통 운영에 필요한 소규모 단위 샘플계통을 구성하고 조작이 편리하다. 계통 해석 시에도 계통 전반에 걸친 내용을 한눈에 볼 수 있고 리미트(limit) 등의 체크기능을 색상의 변화로 알려주므로 운영 계획단계에서도 일일이 수작업을 통해 특정 지역을 열람했던 것과는 상당히 진보된 방식으로 작업을 할 수 있다. 이미 이와 같은 사용자 편리성은 상용 프로그램에서 제공되고 있는

2. 본 론

2.1 기존 상용 프로그램 분석

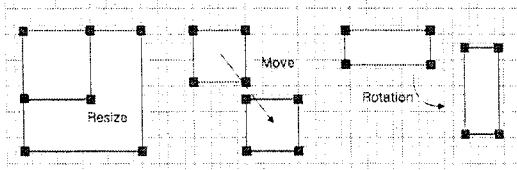
기능들이다.



[그림 3] Blackout 메시지 - PowerWorld 10.0

2.2 구현 방향

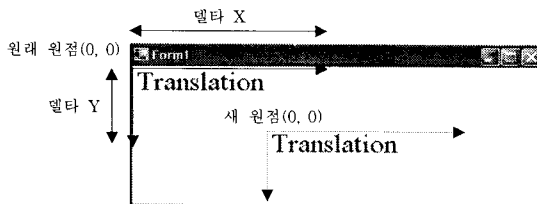
구현은 그래픽 컴포넌트를 구비하는 일련의 과정을 말한다. 그래픽 심벌에 해당하는 컴포넌트들의 경우 그 유 동작 특성을 갖게 된다. 예를 들어 화면에 배치 후 크기를 조절한다든지(resize) 이동한다든지(move) 혹은 회전(rotation)을 하는 식의 조작이 가능해야 한다. 이는 프로그램의 각 객체간 상속성에서 볼 때 기본 클래스에서 정의하면서 서브 클래스에서 적절한 재정의가 이루어지도록 해야 한다.



[그림 4] 그래픽 컴포넌트의 기본 동작들

또한 일반적으로 지원되는 순수 그래픽 컴포넌트에는 다음과 같은 것들이 공통적으로 구현되어야 한다. 프로그램 상에서는 이들을 열린 도형과 닫힌 도형으로 구분하는데 열린 도형으로는 line, polyline, freeline, arch가 있고 닫힌 도형으로는 rectangle, ellipse, circle 등이 있다. 이들 객체의 물리적 동작을 정의하기 위해서는 행렬 연산을 통한 수학적 모델링 과정을 수식으로 정립해야 한다.

예를 들어 이동 변환과 회전 변환의 경우를 보자. 이동 변환은 그리기 연산의 결과를 지정된 수치들만큼 좌, 우, 상, 하로 이동시키는 역할을 한다.



[그림 5] 이동 변환

이동 변환은 그리기 연산이 적용될 좌표계의 원점을 이동시키는 것으로 간주될 수 있다. 그리기 연산은 원래의 축들과 평행한 축을 따라서 적용된다.

$$\begin{aligned} X(\text{페이지 좌표}) &= X(\text{영역 좌표}) + \Delta X \\ Y(\text{페이지 좌표}) &= Y(\text{영역 좌표}) + \Delta Y \end{aligned}$$

[수식 1] 이동 변환 식

이러한 이동 변환은 가상공간에 대해 그리기 연산을 적용한다는 점을 신경 쓸 필요 없이 좌표계에 독립적인 방식으로 그릴 수 있게 된다.

회전 변환의 경우 다음 식을 따른다.

$$\begin{aligned} X(\text{페이지}) &= X(\text{영역}) \times \text{Cosine}(\text{각도}) + Y(\text{영역}) \times \text{Sine}(\text{각도}) \\ Y(\text{페이지}) &= -X(\text{영역}) \times \text{Sine}(\text{각도}) + Y(\text{영역}) \times \text{Cosine}(\text{각도}) \end{aligned}$$

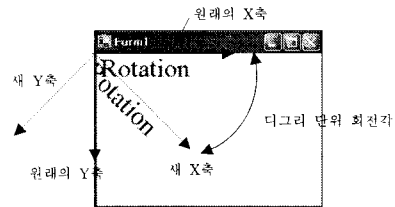
[수식 2] 회전 변환 식

이러한 변환을 행렬연산으로 표현하면(유도 생략) 변환된 점의 최종적인 X, Y 좌표는 점을 의미하는 3×1 벡터와 변환을 의미하는 3×3 행렬의 곱으로 표현된다.

$$\begin{bmatrix} X_T & Y_T & 1 \end{bmatrix} = \begin{bmatrix} X_r & Y_r & 1 \end{bmatrix} \begin{bmatrix} L_1 & L_2 & 0 \\ L_3 & L_4 & 0 \\ T_1 & T_2 & 1 \end{bmatrix}$$

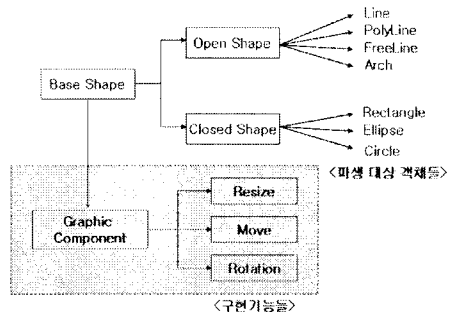
[수식 3] 행렬을 이용한 Rotation

여기서 X_0 와 Y_0 는 원래의 좌표들을 뜻하고, 3×3 행렬은 영역 변환을 수행하는데 필요한 모든 정보를 담는다. 그리고 X_T 와 Y_T 는 변환 후의 페이지 좌표들이다. 이를 응용해 대칭 변환(뒤집기 변환)도 가능하다.



[그림 8] 회전 변환

그래픽 컴포넌트는 자신만의 상태를 나타내는 속성을 갖고 있어야 한다. 이는 경계선에 관련된 색상 굵기 그리고 선 타입과 내부 색상 그리고 텍스트의 크기 스타일 상태 등을 표시하는 역할을 한다.



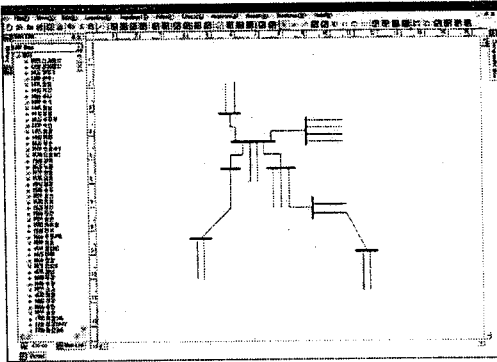
[그림 9] 기본형에 대한 그래픽 객체들

그래픽 객체는 단일 객체만 다루는 것이 아니라 여러 컴포넌트를 하나의 그룹처럼 지정해 범주화 하는 기능이 필요하다. 물론 그룹을 지정할 경우도 크기조절과 이동

그리고 회전 등의 기능이 제공되어야 한다.

2.3 실제 구현

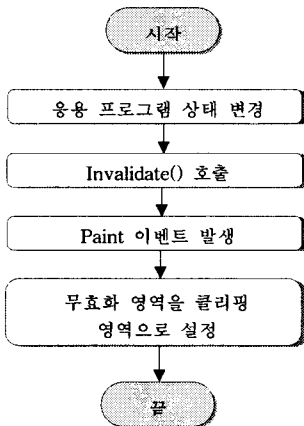
구현 방향에 따라 모선과 선로를 구현해 보았다. 본 샘플 프로그램의 경우 실계통을 읽어들이어 모선을 구성하도록 하였으며 아래와 같은 서브 계통을 구성해 보았다. 이를 위해서는 실계통에 대한 정보를 구성해야 하는데 모선의 유닉성은 모선 번호를 통해서 구분하였고 선로의 경우 from측과 to측의 연결모선 번호로 구분하여 데이터를 구성했다. 기타 동일 선로의 경우 circuit ID로 구분 지었다.



[그림 10] 모선만을 구현한 예

모선의 경우 하위 선로들만을 구성하도록 제한하였으며 양 모선을 시작점과 끝점으로 연결한 선로는 검은색 색상으로 속성을 표시하고 그렇지 않은 미 연결 선로는 붉은 색상으로 나타내도록 설정해 사용자에게 메시지를 전달하도록 하였다. 모선 및 선로의 경우 이동 및 회전 그리고 리사이즈가 가능하도록 구현하였으며 선로의 경우 절점을 주어 꺾은선과 사선이 모두 가능한 배치를 하도록 지원했다.

소규모 계통을 그래픽으로 처리할 경우와 실계통을 대상으로 할 경우는 기술적으로 다음과 같은 처리가 필요하다. 한정된 컴퓨터의 자원으로 모든 계통을 메모리에 담아 두기에는 소모적인 부분이 많기 때문이다. 이는 화면상에 보이는 부분(View port)과 보이지 않는 부분을 구별함으로써 그래픽 엔진의 퍼포먼스 부하를 줄인다.



[그림 11] 무효화 작동 순서도

무효화란 더 이상 유효하지 않게 된 그래서 다시 칠해야 할 부분을 알려주는 것이다. 윈도우에서 무효화는

Invalidate 메서드를 사용해 호출된다. 그래픽 동작의 성능 향상을 위해서는 무효화와 클리핑 영역을 설정해주는 것이 필수적이다. 대규모 시스템을 구현하는데 있어서 윈도우 환경을 채택한 경우라면 반드시 고려되어야 할 사항이다.

3. 결 론

본 논문에서는 지난 “그래픽 인터페이스를 통한 조류계산 설계”에 이어 구현과정을 알아보았다. 기존 상용 프로그램의 제공 기능 및 수준 그리고 이를 위한 구현 방향으로 그래픽 컴포넌트가 갖추어야 할 기본 기능을 알아보았다. 최종적으로는 이를 바탕으로 한 실제 구현의 단계까지 보였다.

차후 개선할 점은 현재 프로그램의 기본 계통 컴포넌트가 완벽히 갖추어진 상태가 아니므로 그러한 요소부터 모델링을 하도록 해야 할 것이며 해석 연산 후에 디스플레이 기능을 추가해야 할 것이다. 또한 실무에서 필요로 하는 리포트 기능을 갖추는 방향으로 연구가 필요하다.

[참 고 문 헌]

- [1] 신만철, 김건중, 엄재선, 이병일, 박철우, 장중철, “Graphic-based Power System Simulator 소개”, 대한전기학회 전력기술부문의 추계학술대회 논문집, p.133-136, 2001년 11월
- [2] 신만철, 김건중, 박철우, 박헌경, 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] Eric Lengyel, “Mathematics for 3D Game Programming and Computer Graphics Second Edition”, CHARLES RIVER MEDIA, 2004