

객체지향 모델링에 기반한 발전소 시뮬레이션 툴 개발

전상규*, 손기현**

Development of a Power Plant Simulation Tool Based on Object-Oriented Modeling

Sang Gyu Jeon, Gihun Son

Abstract

A power-plant simulation tool has been developed for training the plant operators and testing a plant control system. The simulation tool is composed of a graphic editor, a component model builder and a system simulation solver. Such new programming techniques as object-oriented modeling and GUI(Graphical User Interface) are employed in developing the simulation tool. The graphic editor is based on the OpenGL library for effective implementation of GUI while the component model builder is based on object-oriented programming for efficient generalization of component models. The developed tool has been verified through the simulation of a real power plant.

Key Words: GUI(Graphical User Interface), Object-Oriented Modeling, Power Plant Simulation.

* 서강대학교 기계공학과 대학원

** 서강대학교 기계공학과

1. 서론

발전소는 화석에너지 또는 원자핵에너지 등을 이용하여 급수를 고온, 고압의 증기로 변환하여 터빈을 구동시킴으로써 전력을 생산하는 시스템이다. 이러한 발전계통에는 열교환기, 펌프, 밸브, 탱크 등의 컴포넌트가 복잡하게 연결되어 있고 발전소마다 그 컴포넌트와 그 연결 상태가 모두 다르기 때문에 수작업으로 각각 코딩하는 것은 매우 비효율적이다. 따라서 이러한 발전소의 복잡한 계통의 효율적인 시뮬레이션을 위하여 GUI(Graphical User Interface)에 기반을 둔, PRO_TRAX, US3(Unix Simulation Software System), MMS(Modular Modeling System) 등의 시뮬레이션 툴들이 국외에서 개발되었다[1~3].

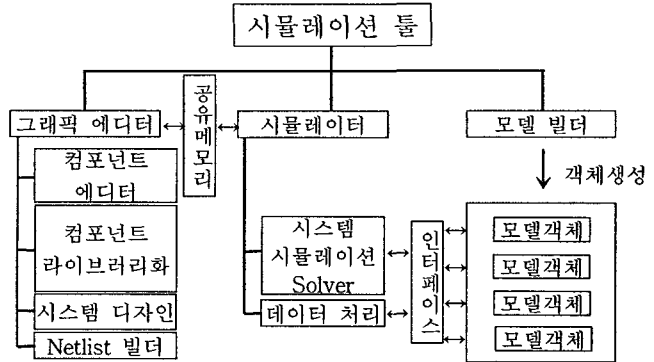
현재 발전소에서 운용되고 있는 기존 국외 시뮬레이션 툴은 구조적인 문제와 마케팅 전략으로 인해 운용 시에 컴포넌트의 확장이 어렵다는 제약이 따른다. 또한 숙련된 사용자가 새로운 모델을 개발하여 시뮬레이션 툴에 적용하려고 해도 개발사에 제품 개발을 요청하여 개발하지 않는 이상 시뮬레이션 툴에 적용할 수 없는 한계를 가지고 있다.

본 논문에서는 객체지향 모델링을 통해 기존 시뮬레이션 툴이 갖고 있던 시뮬레이션 툴의 확장성의 문제를 해결한 방법과 사용자가 시스템의 디자인으로부터 시뮬레이션을 실행하여 데이터를 얻는 과정에서 기존 시뮬레이션 툴과 차별화시켜 중간 단계의 과정을 거치지 않도록 한 방법에 대해 설명하고, 실제 호남화력발전소의 급수계통 데이터를 이용하여 개발된 시뮬레이션 툴을 검증하였다.

2. 시뮬레이션 툴의 구성

시뮬레이션 툴은 <그림1>에서 나타난 바와 같이 그래픽 에디터와 시뮬레이터, 모델 빌더로 구성되어있다. 그래픽 에디터와 시뮬레이터는 각각 별도의 프로그램으로 개발하였고 모델 빌더는 시뮬레이터 프로그램에 포함하였다. 그래픽 에디터 기능의 효과적인 구현을 위해서는 시뮬레이터 개발과는 구분되는 전문적인 지식과 능력이 요구되기 때문에 그래픽 에디터와 시뮬레이션 Solver 간의 프로

그래밍 영역은 확연히 구분되고 서로의 영역



<그림1> 시뮬레이션 툴의 구성도

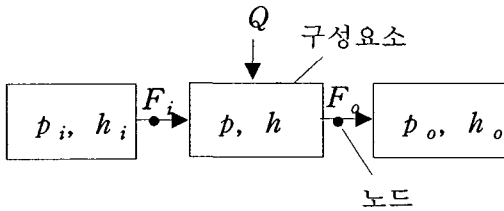
에 대해 모두 이해하는 데는 어려움이 따르기 때문이다. 따라서 개발과정과 향후 업그레이드 과정에서의 효율적인 작업을 위하여 그래픽 에디터와 시뮬레이터를 각각 서로 다른 프로그램으로 개발하였고 공유메모리를 통해 사전에 정의된 형식으로 데이터를 교환하도록 하였다. 또한 시뮬레이션 Solver와 모델 객체 사이의 데이터 교환은 별도의 인터페이스를 정의하여 일반적인 방법으로 데이터를 교환할 수 있도록 하였고, 향후 컴포넌트를 추가할 때는 모델 빌더와 컴포넌트 라이브러리에 컴포넌트를 추가함으로써 전체 프로그램의 변경이 없이도 컴포넌트가 추가되도록 하였다.

3. 시뮬레이션 Graphic Editor

발전소는 다양한 컴포넌트들이 복잡하게 연결되어 있기 때문에 시뮬레이션을 위해서는 발전계통의 다양한 컴포넌트들을 배치하고 연결하는 과정이 필수적이다. 따라서 시뮬레이션의 효율적인 작업을 위해서는 GUI 방식의 그래픽 에디터가 필수적으로 요구되고 현재 사용되고 있는 시뮬레이션 툴들이 GUI 방식을 사용하고 있다. 따라서 본 연구에서도 발전소 시뮬레이션만을 위한 그래픽 에디터를 개발하게 되었다. 최근 시뮬레이션 소프트웨어의 개발경향이 사용자 인터페이스에 단순히 데이터를 보여주는 기능을 제공하던 것에서 벗어나 애니메이션 기능을 추가하는 등의 비주얼한 요소에 중점을 두고 있기 때문에 3D 그래픽을 효율적으로 구현할 수 있는

OpenGL library를 이용하여 향후 그래픽 에디터의 애니메이션 기능을 업그레이드하는데 용이하도록 하였다.

4. 열수력 모델의 개괄적 검토



<그림 2> 구성요소간의 연결 상태

<그림 2>는 열교환기, 펌프, 밸브, 탱크등과 같은 발전소 컴포넌트의 연결 상태를 나타내며, 노드는 각 구성요소를 연결하는 점으로 정의되었다. 구성요소의 열수력 현상을 지배하는 방정식은 질량, 에너지, 운동량 보존 방정식이며, 다음의 형태로 표현할 수 있다.

$$\frac{d}{dt} M = F_i - F_o \quad (1)$$

$$\frac{d}{dt} Mh = F_i h_i - F_o h_o + Q + V \frac{dp}{dt} \quad (2)$$

$$F = a \Delta p + b \quad (3)$$

여기서, M, F, h, Q, p 는 구성요소의 질량, 질량 유량, 엔탈피, 열량, 압력이며, a, b 는 압력과 유량의 관계를 결정하는 계수이다. F, h, p 는 위의 미분 방정식으로부터 얻어진다[4]. 또한 나머지 변수는 EPRI의 RETRAN 매뉴얼에서 제공된 상태방정식과 열전달 관계식을 이용하여 구할 수 있다[5].

5. 시물레이션 Solver

5.1 Netlist를 이용한 구성요소 모델의 계통화

앞에서 언급한바와 같이, 발전계통의 시물레이션은 수작업이 비효율적이기 때문에 GUI를 이용하는 모델링이 필요하다. 먼저, Library에 등록된 펌프, 밸브 등의 구성요소를

배열하고 연결해야 한다. 실제 모델링에서는 이러한 구성요소간의 연결상태를 나타내는 Netlist가 필요하다. 즉, 에너지 보존식으로부터 엔탈피를 구하거나 압력방정식으로부터 압력을 구하기 위해서는 구성요소-구성요소간의 연결 상태를 알아야하며, 이는 노드-구성요소간의 Netlist를 생성하는 그래픽 에디터로부터 받는다. 구성요소-구성요소간의 연결 정보로부터 에너지 보존식은 어려움 없이 풀 수 있다. 그러나, 압력방정식을 풀기 위해서는 다음과 같은 행렬연산이 필요하다.

$$A \vec{p} = \vec{q} \rightarrow \vec{p} = A^{-1} \vec{q} \quad (13)$$

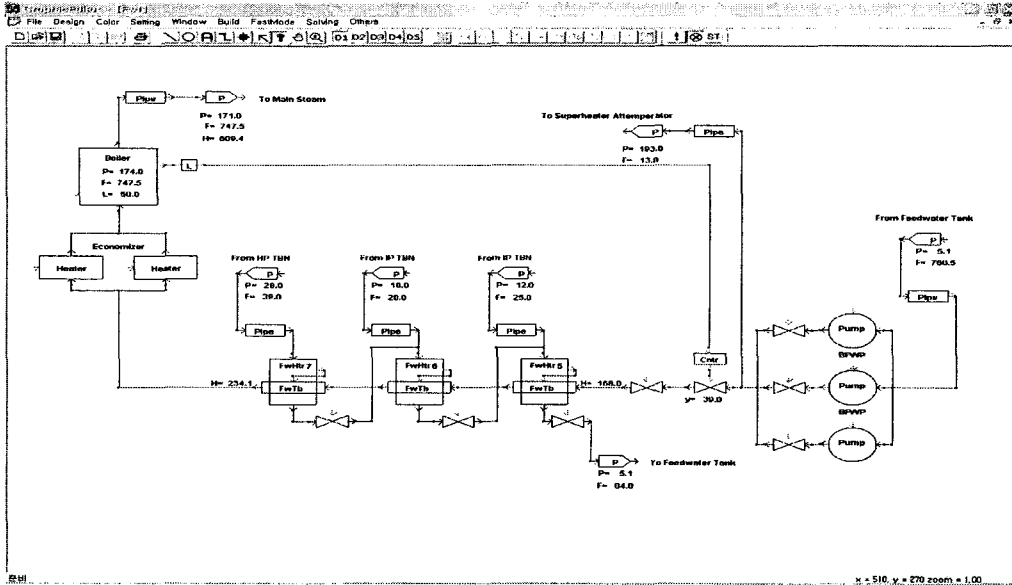
본 연구에서는 기존에 알려진 Gauss Elimination 방법을 사용하되 계산의 효율성을 높이기 위해 행렬 A 의 영에 해당하는 부분은 연산이 되지 않도록 하였다.

5.2 객체지향 기법을 이용한 모델 빌더

발전계통의 시물레이션은 많은 수의 모델을 필요로 하며 실제 시스템에 근접한 시물레이션 결과를 얻기 위해서는 좀더 정확한 컴포넌트의 모델링이 요구되기 때문에 여러 전문가들이 분담해서 자신의 분야에 속한 모델 빌더를 만들어 통합하는 과정이 필수적이다. 이러한 통합 과정을 효과적으로 하기 위하여 객체지향 언어인 C++를 이용하여 각각의 모델에 필요한 변수와 함수를 class의 멤버로 선언하여 객체를 대상으로 개발함으로써 모델 빌더 간 독립적인 개발과 통합이 용이하도록 하였다. 또한 객체지향 기법을 적용한 모델 빌더를 구현함으로써 소스코드의 변경 없이 추가된 컴포넌트 클래스를 포함하여 컴파일함으로써 시물레이션 툴에 새로운 컴포넌트를 추가할 수 있도록 하여 시물레이션 툴의 확장성을 향상시킬 수 있었다.

5.3 모델 간 인터페이스 구축

시물레이션 Solver는 에너지 보존식으로부터 엔탈피를 구하거나 압력방정식으로부터 압력을 구하는 행렬계산을 필요로 한다. 이러한 연산을 수행하는데 사용되는 데이터는 각각의



<그림3> 호남화력 급수계통의 시뮬레이션 결과

컴포넌트 모델로부터 얻어 와야 하는데 각각의 컴포넌트 모델이 모델 빌더 class의 객체로 생성되기 때문에 시뮬레이션 Solver가 각

각의 모델 객체에 접근할 수 있는 일반적인 인터페이스가 요구된다. 따라서 Component Object Model(COM)에서 사용하는 인터페이스 구축 방식을 적용하기 위하여 인터페이스를 아래와 같이 모든 모델 빌더에 공통적으로 사용되는 함수를 가상함수로 가지는 추상 class로써 구성하였다[6].

```
class CBase
{
public:
    virtual void Input(int, int, double)=0;
    virtual void Input(int, double)=0;
    virtual void Input(int, CString)=0;
    virtual void Save(CArchive&)=0;
    virtual void Load(CArchive&)=0;
    virtual double Output(int, int)=0;
    virtual CString Output(int)=0;
    virtual CString VarName(int)=0;
};
```

각각의 컴포넌트 모델 빌더는 인터페이스 class로부터 상속을 받아 인터페이스를 외부에 노출하도록 함으로써 시뮬레이션 Solver에

서 가상함수 테이블을 통하여 모든 컴포넌트 모델 객체에 접근할 수 있게 된다. 즉 시뮬레이션 Solver에서 하나의 인터페이스 class의 포인터를 통해 모든 객체에 정의된 멤버함수

를 호출함으로써 각 객체에서 제공하는 메소드를 이용하여 계산에 필요한 데이터를 처리하도록 하였다. 또한 기존 시뮬레이션 툴과 차별성을 위하여 그래픽 에디터가 제공하는 Netlist와 일반화된 인터페이스를 이용하여 그래픽 에디터에서 디자인한 시뮬레이션 결과를 중간 단계를 거치지 않고 바로 확인할 수 있도록 하였다.

6. 시뮬레이션 결과

위에 설명한 발전소 시뮬레이션 툴을 이용하여 호남 화력의 급수 계통을 시뮬레이션 한 결과를 <그림3>에 나타내었다. 열수력 모델에 필요한 입력 데이터는 각 컴포넌트에 배정된 대화상자를 통하여 입력되었다. 급수계통의 시뮬레이션에서, 압력방정식의 경계조건으로 급수탱크의 압력과 주증기의 압력을 선택하였다. <그림3>에서 각 구성요소에 출력된 숫자는 드럼의 수위를 50%로, 압력을 174kg/cm²으로 유지했을 때 유량(F), 엔탈피(H), 압력(P), 수위(L), 밸브의 개도(y) 등의 계산결과를 나타낸다.<그림3>에 나타난 바와 같이, 정

상상태에서 급수탱크로부터 입력되는 유량은 주증기로 출력되는 유량과 같다. 이는 질량 보존을 지배하는 압력방정식이 바르게 풀렸음을 증명한다.

7. 결론

본 연구에서는 발전 시스템의 컴포넌트를 객체지향 기법을 이용하여 개발함으로써 시뮬레이션 툴의 확장이 용이하도록 하였고 시뮬레이션 Solver에서 일반적인 방법으로 접근할 수 있는 인터페이스와 그래픽 에디터에서 제공하는 Netlist를 이용하여 시스템 디자인에서 시뮬레이션 데이터를 얻기까지 중간 단계를 거치지 않고 바로 할 수 있도록 하였다. 또한 호환화력의 급수계통을 시뮬레이션 함으로써 발전소 시뮬레이션 툴의 기능이 정상적으로 수행됨을 증명하였다.

추후에 그래픽 에디터의 애니메이션 기능과 다양한 에디팅 기능을 추가하고, 모든 컴포넌트에 COM(Component Object Model)을 적용하여 컴포넌트 추가 시 추가된 컴포넌트만을 컴파일해서 추가하도록 함으로써 시뮬레이션 툴의 확장성 개선하여 시뮬레이션 툴의 기능을 업그레이드 하고자 한다.

참고문헌

- [1] US3, "User Guide Release 5", S3 Technologies, Columbia, Maryland, 1994
- [2] ProTRAX, "Analyst's Instruction Manual", Vol. 1 and 2, TRAX Corporation, Forest, Virginia, 1994.
- [3] MCKim C. S. and M.T. Matthews, "Modular Modeling system Model Builder", Intersociety Energy Conversion Engineering Conference, Washington, D.C., 1996.
- [4] 손기현, 조병학, 김동욱, 이용관, "발전소 시뮬레이션 툴을 위한 열수력 모델 구현", 한국시뮬레이션학회 추계학술대회논문집, 1998, p95~99.
- [5] McFadden, J.H. et. al., "RETRAN-02-A Program for Transient Thermal-Hydraulic

Analysis of Complex Fluid Flow Systems, Volume 1; Theory and Numerics(Revision 5)", EPRI NP-1850-CCM, 1992.

- [6] 김응연, "COM Bible", 1판, 삼양출판사, 2001.