

## 자바 애플릿을 이용한 블록 다이어그램식 GUI 개발

황성환\*(KIMM), 한형석(KIMM), 이재경(KIMM), 김동성(KIMM)

### Development of a Block-diagram type GUI in JAVA Applet

S. H. Hwang(KIMM), H. S. Han(KIMM), J. K. Lee(KIMM), D. S. Kim(KIMM)

#### ABSTRACT

Due to advances in information technology, even Engineering Simulations can now be performed in web-based JAVA through an integrated operating system called Virtual Machine and the use of byte code, which guarantees the compatibility of identical codes in every computing system and makes it suitable for web-based simulation system development. This paper introduces an implementation embodied in JAVA Applet that allows a block-diagram type GUI that runs in a web browser for use in the dynamics simulation modeling of powertrains of vehicles and multi-body systems. This system is not restrained by any of the H/W and S/W in the user's computer, so that it has the advantage of providing a GUI that allows web-based block-diagram type modeling.

**Key Words :** Web-based Simulation(웹기반 시뮬레이션), JAVA Applet(자바 애플릿), DB(데이터베이스)

#### 1. 서론

정보 기술의 발전으로 공학적 시뮬레이션도 웹 기반으로 많이 수행되고 있다. 웹의 개념이 처음 제시되고 확장되어 가던 초창기에는 인터넷의 느린 속도로 인한 데이터 전송 문제, 각각의 컴퓨터 사양과 환경의 다양성, 그리고 보안 문제 등의 공학적 시뮬레이션을 하기에는 많은 문제점들을 안고 있었다. 느린 데이터 전송 속도 문제는 전송 매체의 발달로 해결되고 있으며 보안과 호환성 문제는 소프트웨어적인 측면에서 해결해야 할 문제이고, JAVA<sup>1,2</sup>는 이러한 문제점을 해결할 수 있는 대안으로 제시되었다. JAVA는 메모리에 대한 직접 접근, 포인터 연산을 허용하지 않고 Encryption이라는 암호화 기법을 사용하여 한층 강화된 보안 체계를 유지한다. JAVA는 프로그램 실행 시 바이트코드를 사용하는데, 바이트코드 번역 시 JAVA 규약에 위배되는 사항이 있을 때에는 실행이 불가능하게 되어 있으며, 내부적으로 Class Loader가 파일 시스템과 네트워크를 분리한 'Name Space'라는 개념을 가지고 보안을 유지한다. 또한 Virtual Machine이라는 환경에서 실행되도록 만들어진 JAVA는 Virtual Machine이 구축되어 있는 모든 종류의 컴퓨터에서 같은 동작을 보이도록 만들어졌다.<sup>3,4</sup> 본 논문에서는 자동차

동력 전달계의 성능 시뮬레이션 모델링에 사용되는 블록 다이어그램식 GUI를 JAVA Applet으로 구현한 O-DYN/Modeler를 소개하고 시스템 아키텍처와 적용 사례를 소개한다.

#### 2. 시스템 구성

##### 2.1 시스템 아키텍처

Fig. 1은 본 논문에서 소개하는 O-DYN/Modeler의 시스템 구조를 보여주고 있다. 시스템은 O-DYN/Modeler, Job Manager, Data Manager, User Manager, Solver Manager, Solver로 구성된다. 사용자 인터페이스인 O-DYN/Modeler는 웹서버에서 사용자 웹브라우저로 다운로드되어 수행되며, 기타 시스템 모듈은 웹서버에 위치하여 사용자의 작업요청을 처리하며 모든 데이터는 데이터베이스에 저장된다. 사용자는 O-DYN/Modeler를 사용하여 해석모형을 생성, 저장하며, 사용자의 해석요청에 따라 Job Manager가 저장된 해석모형을 Solver Manager에 전달하여 해석을 수행한다. User Manager는 사용자 인증을 담당하며, Data Manager는 시스템의 모든 데이터를 데이터베이스에 저장하고 저장된 해석모형, 해석결과를 해석기의 입출력 파일형식으로 데이터를 변환시킨다.

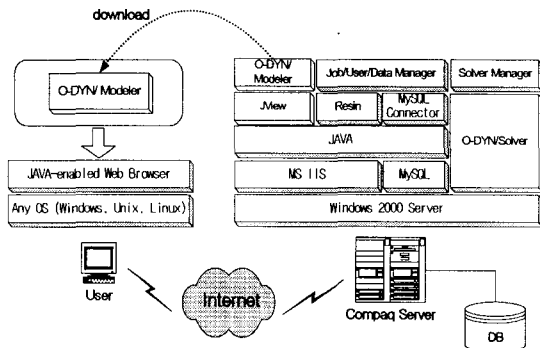


Fig. 1 System architecture

## 2.2 시스템 운영 및 개발환경

앞 절에서 기술한대로 O-DYN/Modeler 는 웹 브라우저를 통하여 실행하며, Data/Job/User Manager, Solver Manager 와 Solver 는 데이터베이스와 함께 웹 서버에서 운영된다. Table 1 은 시스템의 개발 및 운영에 이용되는 하드웨어, 소프트웨어를 보여주고 있다. 사용자의 작업이 처리되는 서버는 하드웨어로 Compaq ML-570 Server, 운영체제로는 Windows 2000 Server 를 사용하였다. O-DYN/Modeler 를 위해 서버에서 운영되는 소프트웨어로는 MySQL 데이터베이스 서버, Resin Servlet/ JSP(JAVA Server Page)<sup>5</sup> 엔진, 자체 개발한 동력전달계 성능 해석기(Solver), MS-IIS 웹서버 등이 있다. 프로그래밍 개발툴은 JDK 1.3 을 사용하였으며, GUI 라이브러리 JViews<sup>6</sup> 는 블록다이어그램 방식의 사용자 인터페이스를 개발하는데 사용되었다. 데이터를 관리하기 위해 사용된 MySQL 은 관계형(Relational) 데이터베이스 관리 시스템으로 별도의 비용 없이 사용할 수 있고 동시에 다수의 사용자가 데이터를 공유할 수 있으며 데이터의 백업이나 관리가 용이한 장점이 있다.

Table 1 System development environment

OS	Windows 2000 Server
H/W	Compaq Server, 1 CPU
Web Server	MS IIS 5.0
Servlet/JSP Engine	Resin 2.1
Programming Tools	JDK 1.3
DBMS	MySQL 3.23
Library	JViews 5.5

웹기반 시스템인 O-DYN/Modeler 는 다수의 사용자가 동시에 시스템을 이용하기 때문에 동시 사용자 수의 제한 및 로드 밸런싱(load balancing) 문제가 발생할 수 있다. 로드 밸런싱 및 시스템 동시 사용자 수의 제한은 서버의 하드웨어 성능, 네트워크

속도, 사용된 소프트웨어의 사용자 라이선스 수에 제한을 받는다. 동시 사용자 수는 시스템에 연결하여 각 모듈을 동시에 사용하는 사용자 수이며 O-DYN/Modeler 는 사용자 웹브라우저로 전송된 후 수행되므로 시스템에 영향을 주지 않는다. 또한 인터넷으로 연결되어 시스템을 사용하기 때문에 전송량이 중요하며 O-DYN/Modeler 는 GUI 라이브러리 JViews 를 포함한 기본 모듈이 대략 2MB 정도이다. 기본모듈은 사용자 웹브라우저의 캐쉬에 저장되어 매 사용시마다 전송되지 않으므로 사용자의 해석모델이 전송량을 결정한다. 웹기반 시스템은 인터넷의 특성상 불특정 다수가 시스템에 접속할 수 있으므로 허가된 사용자들만이 시스템을 이용할 수 있는 방법이 제시되어야 한다. O-DYN/Modeler 에서는 네트워크 보안성(방화벽이나 기타 네트워크 레벨의 보안성)이 확보되었다는 가정 하에 사용자에게 id 와 password 를 부여하고 사용자의 네트워크 주소(IP 주소)를 서버에 등록하는 2 단계의 인증을 수행한다. 1 단계에서는 비등록된 네트워크 주소로부터의 시스템 연결은 거부되며 2 단계에서는 등록된 네트워크 주소라도 사용자 id 와 password 를 검사하여 일치하지 않을 경우엔 거부된다. 또한, 인증을 거쳐 시스템의 사용이 허가되었다라도 시스템의 모든 사용자가 작업데이터를 공유하는 것은 바람직하지 않으며, O-DYN/Modeler 에서는 사용자별로 작업데이터를 관리하고 사용자가 자신과 작업데이터를 공유할 사람을 지정하는 방법을 사용한다. 시스템의 시험적 운영결과 작업데이터의 공유방법, 버전닝(versioning), 작업그룹 개념의 도입 등 협업 작업의 개념을 확대시켜야 될 것으로 사료된다. 이러한 운영환경 하에서 O-DYN/Modeler 의 웹기반 공학 시스템으로의 장점은 다음과 같다. 첫째, 사용자 컴퓨터에 별도의 S/W 를 설치할 필요가 없으므로 S/W 구입비용과 설치비용을 절감할 수 있다. 둘째, S/W 의 업그레이드가 서버측에서만 이루어지므로 업그레이드 비용과 시간을 절감할 수 있다. 셋째, 자체 개발한 해석기를 이용하기 때문에 고가인 해석기 사용권 문제에서부터 자유롭다. 넷째, 각 사용자 컴퓨터가 고성능일 필요가 없으며 고성능의 서버만을 유지함으로써 비용절감 및 효율적인 사용이 가능하다. 다섯째, 작업결과가 서버에 저장되고 관리되기 때문에 중요 작업데이터에 대한 관리 및 보안성이 강화되며 작업데이터에 대한 공유가 가능하다. 여섯째, 정보를 동기 또는 비동기적으로 다수의 사용자가 공유할 수 있어 협업 및 동시공학 지원이 가능하다.

## 3. 개발과 적용

시스템 개발에 사용된 프로그래밍 언어는 JAVA 이며, 웹 프로그래밍 기법으로는 웹 표준인 HTML 과 JAVA Applet, JSP, Servlet 을 사용하였다. 사용자 인터페이스인 O-DYN/Modeler 는 JAVA Applet 으로 개발하였고, 해석기 Solver 를 제외한 웹서버의 시스템 모듈은 JAVA Servlet 과 JSP 로 개발되었다. JAVA Applet 은 웹 브라우저에 부착되어 ActiveX 처럼 사용되며, 별도의 프로그램 설치가 필요 없이 필요한 모듈은 다운로드 받아서 사용할 수 있는 장점이 있다. 반면, JAVA Applet 은 보안상 사용자 컴퓨터의 파일 제어 및 자신이 위치한 웹서버 이외의 시스템과의 통신이 불가능하다. 따라서, 시스템의 모든 데이터는 웹서버의 DB, 파일 시스템에서 관리되어야 하며, 이를 통해 데이터의 체계적 관리와 보안성을 확보할 수 있다. 보안 제약상 사용자 컴퓨터로부터 서버로 파일을 업로드 할 경우(예: curve data) JSP 를 사용한 파일 업로더를 구현하여 사용하였다. 웹서버 쪽의 시스템 모듈 개발에 사용된 JAVA Servlet 은 JAVA Applet 이 처리할 수 없는 서버의 응용 프로그램 실행이나 파일처리를 수행할 수 있으며, JAVA Applet 과 통신을 통하여 데이터를 주고 받을 수 있으며 세션과 쿠키 제어를 통하여 사용자 인증을 제공할 수 있다. 현재 O-DYN/Modeler 는 단일 시스템에서 수행되지만, 향후 웹서버 시스템, 데이터베이스 관리 시스템, 동력전달계 성능 해석 시스템으로 분리되어 확장 운영될 수 있으며, 이를 위해서는 JAVA Applet 방식의 사용자 인터페이스가 앞에서 언급한 보안제약을 극복할 수 있어야 한다. 즉, 데이터베이스 서버, 동력전달계 성능 해석기가 웹서버와 다른 시스템에 위치해 있으면, 보안계약으로 O-DYN/Modeler 가 데이터베이스 서버, 해석기를 사용할 수 없는데 JAVA Servlet 을 통하여 이를 해결하였다. O-DYN/Modeler 는 웹브라우저 상에서 블록 다이어그램 형태의 사용자 인터페이스를 구현하기 위하여 GUI 라이브러리 JViews 를 사용하였으며, 사용자간의 해석모델 공유가 가능하다. 이때 동시에 같은 해석모델을 수정하는 것을 방지하기 위해 해석모델의 생성자만이 수정할 수 있으며, 공유가 허가된 타 사용자는 읽기가 허용된다.

Fig. 2 는 O-DYN/Modeler 의 주요 기능과 작업흐름 그리고 이와 관련된 데이터베이스를 나타내며 운영 시나리오는 다음과 같다. JAVA 를 이용할 수 있는 웹 브라우저를 이용하여 사용자는 시스템에 접속하고, 사용자 인증과정을 거친 후 웹서버에서 전송된 O-DYN/Modeler 를 사용한다. O-DYN/Modeler 는 모델링에 필요한 정보를 데이터 베이스에서 동적으로 요소 트리를 구성하므로 쉽게 확장될 수 있다.

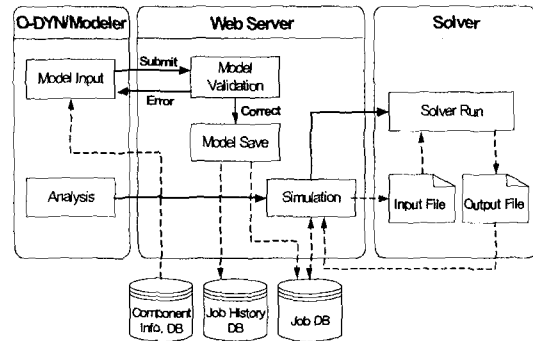


Fig. 2 Data / Control flowchart

Fig. 3 은 O-DYN/Modeler 의 초기화면으로, 왼쪽의 트리 구조가 데이터베이스로부터 다운로드 받은 요소 트리이고 오른쪽이 에디터로 사용된다. 사용자는 화면 왼쪽의 요소 트리에서 필요한 요소를 선택하여 블록 다이어그램을 모델링하고 해석모델을 저장한다. 각 요소의 파라미터 입력은 블록을 클릭하여 입력 윈도우를 띄워서 입력한다. 모델링 검토를 마친 사용자가 동적 해석을 위해서 'Analysis'를 선택하면 웹서버의 Job Manager 가 호출되어 데이터베이스에 저장된 해석모델을 해석기의 입력파일로 변환하고 Solver Manager 에 전달한다. Solver Manager 는 성능해석기 Solver 를 배치작업 형태로 실행하여 해석을 수행한다. 해석이 끝나면 Job Manager 가 생성된 해석결과 파일을 데이터베이스에 저장하고 사용자에게 해석이 끝났음을 알린다.

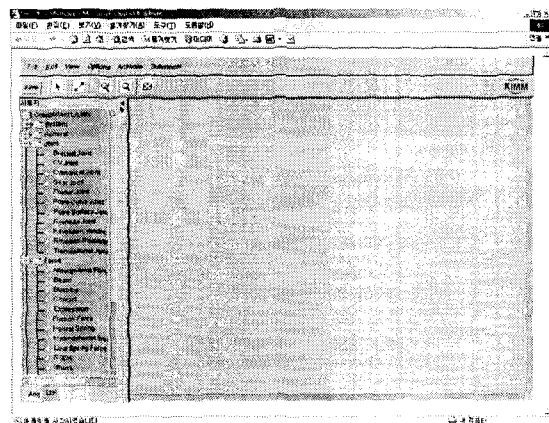


Fig. 3 O-DYN/Modeler

본 논문에서 소개한 블록 다이어그램식 GUI 를 이용하여 복동식 압축기의 설계를 수행하였다. Fig. 4 는 O-DYN/Modeler 를 이용하여 복동식 압축기를 모델링 한 예이다. 각 블록은 복동식 압축기의 요소를 표현한 것이고, 각 블록을 클릭하여 각 요소의 입력 파라미터 값을 설정한다. 각 요소들 간의

연결관계는 Connection 기능을 두어서 화살표 형태로 표현하도록 하였다. 각 모델링 요소의 데이터 입력과 연결관계 설정작업이 끝나면 해석모델을 저장하고, Analysis 메뉴를 통하여 웹서버의 동력전달계 성능 해석기를 실행하여 성능해석을 수행할 수 있다.

411~416, 2002.

5. <http://www.caucho.com/resin/>

6. <http://www.ilog.com/>

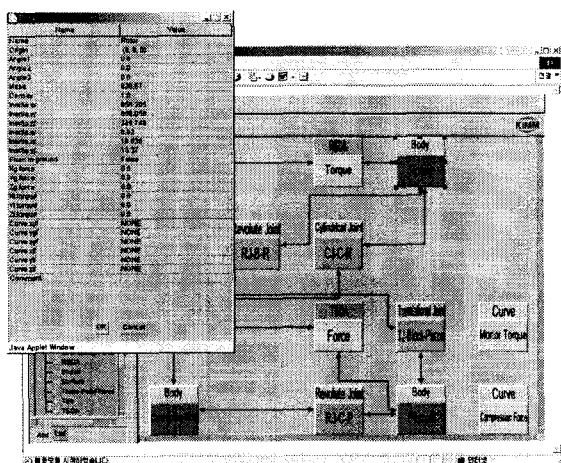


Fig. 4 Block-diagram type modeling in O-DYN/Modeler

#### 4. 결론

본 논문에서는 자바 애플릿을 이용한 블록 다이어그램식 GUI 개발을 소개하였다. 사용자 인터페이스는 웹 브라우저와 인터넷을 통하여 이루어지며 JAVA Applet 에 기반을 두고 개발되었다. 설계된 해석모델은 데이터베이스에 저장되어 해석모델의 공유가 가능하다. 결과적으로 사용자는 사용자 컴퓨터 환경과 위치에 상관없이 인터넷과 웹 브라우저를 통하여 블록 다이어그램식 모델링이 가능하다. 또한 동시적인 정보의 공유로 협업 및 동시공학 구현이 가능하고 정보의 데이터베이스를 이용한 체계적인 관리가 가능한 장점이 있다.

#### 참고문헌

1. <http://java.sun.com/>
2. Kim, H. K., JAVA Network Programming, Hanbit Media.Inc., 2001.
3. Kim, K. Y. and Nam, Y. H., "A Visual Modeling Environment for Web-based Simulation," Journal of the Korea Society for Simulation, Vol. 8, No. 1, pp. 101~111, 1999.
4. Jin, G. J., Kwak, M. K. and Heo, S., "Development of Web-Based Engineering Calculation Program Using JavaScript," Proc. of KSNVE Spring Conf., pp.