

분산 환경하에서의 웹기반 시뮬레이션에 관한 연구*

Web-Based Simulation under Distributed Environment

이영해**, 곽성근**, 김숙한**

Young Hae Lee, Seong Gun Kwak, Sook Han Kim

Abstract

This paper introduces the concept of web-based simulation and suggests the structure of web-based simulation which reduces the simulation run time and performs simulations in efficient way under distributed environments. Since its introducing in 1996, web-based simulation has been studied only with a tool of applet, but in this paper a method of server applications for client applets will be used. In server application, server transfers objects requested by clients such as simulation engines, reports, files. After each client connects to web-server, and then server allocates simulation modules to connected clients. These work magnify the transferring applets from server and simulation models which were made by clients. This paper also proposes a structure for managing efficiently web-based simulation under distributed environment, and steps in which clients connect, model, simulate with distributed structure, and programs of proposed structure.

* 본 연구는 1998년도 정보통신부 대학기초연구지원사업에 의하여 지원되었음.

** 한양대학교 산업공학과

1. 서론

현대 시스템은 점점 대형화, 복잡화 되고 있다. 이러한 시스템들에 대하여 기존 시스템의 효율적 개선을 위해 시스템을 평가하거나, 새로운 시스템을 설계할 때, 간단한 분석적인 방법이나 비현실적인 가정들을 가진 수리적인 방법으로 해결할 수 없는 경우가 많이 발생한다. 이러한 경우를 위하여 이산 사건 시뮬레이션 기법이 많이 활용되어 왔다. 빠른 시간 내에 시스템을 설계 및 개발하고자 할 때, 개발된 시스템을 시뮬레이션을 통하여 사전 분석, 평가한 후에 시스템에 적용시킴으로써 개발시간 단축, 비용 감소 및 개발의 위험을 줄일 수 있다[1].

또한 인터넷 환경을 이용하여 시뮬레이션을 수행한다면 시뮬레이션 엔진을 전송받아 실행할 수 있기 때문에 편리할 뿐 아니라 추가의 소프트웨어 구입없이 사용할 수 있어 비용의 절감 효과도 누릴 수 있다[3, 4].

본 연구는 이러한 인터넷과 시뮬레이션의 결합, 즉 웹기반 시뮬레이션(Web-Based Simulation)에 관한 정의 및 최근 연구 동향을 논하고, 본 연구에서 개발한 웹 상에서 구현 가능한 시뮬레이션 S/W의 구조와 개발 환경 및 단계별 수행 절차에 대해서 기술한다

2. 웹기반 시뮬레이션

웹기반 시뮬레이션은 컴퓨터 시뮬레이션과 World Wide Web (WWW)의 두 방법론의 장점을 접목한 분야이다. 즉 하나의 시뮬레이션 소프트웨어로 여러 명의 클라이언트들이 접속을 통해 웹에서 제공하는 분산 환경하에서의 시뮬레이션 수행을 할 수 있다. 그리고 시뮬레이션에 필요한 모델들을 분산하여 모델링하고 정보 및 데이터를 실시간으로, 원격으로 입력할 수 있다. 또한 이렇게 입력한 모델들을 인터넷을 이용하여 분산 수행하여 수행 시간을 줄일 수 있다는 장점이 있다.

웹기반 시뮬레이션의 일반적인 수행 절차를 다음과 같이 간략화할 수 있다.

- 단계 1 : 시뮬레이션을 원하는 클라이언트들을 웹 서버에 접속한다.
- 단계 2 : 서버는 초기화면과 시뮬레이션에 필요한 엔진을 전송한다.
- 단계 3 : 클라이언트들을 다운받은 초기화면으로 시뮬레이션 모델들을 설계하거나 기존에 모델들을 DB에서 찾아 재사용한다.
- 단계 4 : 각각의 클라이언트들은 시뮬레이션을 수행한 후 원하면 서버 DB에 모델 및 결과들을 저장한다.
- 단계 5 : 접속을 끊는다.

이러한 웹기반 시뮬레이션은 기존의 시뮬레이션들이 시뮬레이션을 수행하기 위해서는 수행하고자 하는 컴퓨터에 시뮬레이션 엔진이 내장되어 있어야 했지만 이러한 기술을 이용함으로써 웹 서버에 내장되어 있는 시뮬레이션 엔진을 각각의 필요한 클라이언트들에게 제공해 줌으로써 클라이언트는 적은 용량의 메모리가 필요하게 되고 간단히 접속하여 사용할 수 있다.

웹기반 시뮬레이션의 목적은 다음과 같이 몇 가지로 나열할 수 있다[4].

- 1) 모델에 대한 정보를 구체화시키고, 많이 사용하는 모델 및 정보를 형상화 시킬 수 있다.
- 2) 웹과 시뮬레이션을 결합시킨다.
- 3) 웹의 DB에 저장되어 있는 수많은 모델을 쉽게 접속하여 탐색할 수 있다.
- 4) 시뮬레이션을 분산 수행하여 수행 시간을 줄일 수 있다.

이것은 모델링하는 데 있어서 많은 시간을 소비해야 하는 것이 아니라 네트워크를 통해서 기존에 다른 클라이언트가 만들어 놓은 모델과 설계하고자 하는 모델과 일치하는 것을 찾아내어 그 모델을 재사용해서 개발에 필요한 시간을 단축시킬 수도 있다는 것을 의미한다.

또한 웹기반 시뮬레이션 S/W를 사용함으로써 얻을 수 있는 장점은 다음과 같이 간단히 요약할 수 있다[7, 10].

- 1) 웹을 통한 분산 모델링 및 시뮬레이션이 가능하

여 복잡한 시뮬레이션 모델에 대하여 수행 시간을 줄일 수 있다.

- 2) 시뮬레이션에 필요한 정보 및 데이터를 원격으로, 실시간으로 입력하고, 시뮬레이션 결과도 동시에 원격 거리에 있는 다수에게 즉시 제공할 수 있다.
- 3) 하나의 S/W로 원격 거리에 있는 다수의 사용자가 시간 및 공간의 구애를 받지 않고 동시에 사용할 수 있으며, 이는 특히 교육적 목적에 크게 효과적이다.
- 4) 서버 DB에 저장되어 있는 다수의 모델을 재사용함으로써 개발에 필요한 시간을 단축시킬 수 있다.

3. 웹 분산 환경을 위한 기술

3.1. 자바

자바(Java)는 Sun Micro Systems사에서 개발한 객체지향 프로그래밍 언어로써 인터넷상에서 산재된 자원들을 효율적으로 이용할 수 있는 방법들을 제공한다. 또한 자바의 실행 파일이 구조 중립적(Architecture-neutral)인 바이트 형태의 코드로 되어 있기 때문에 자바로 개발된 프로그램은 운영체제에 관계없이 동일하게 작동한다.

지금 현재 자바 개발 환경 JDK(Java Development Kit)는 정식버전으로 1.2까지 나와 있으며 엔터프라이즈 환경에 쓰일 수 있도록 자바 데이터베이스 연동 패키지인 JDBC(Java Database Connectivity), 자바 RPC(Remote Procedure Call)인 RMI(Remote Method Invocation), 그리고 자바의 CORBA(Common Object Request Broken Architecture) 지원이 가능하게 되었다.

또한 자바는 네트워크 상에 산재에 있는 정보를 효율적으로 사용할 수 있고 분산 환경을 쉽게 이용할 수 있게 해 준다. 또한 자바는 객체 지향적인 프로그램을 작성하도록 도와주기 때문에 클라이언트/서버간의 정보를 객체에 포함시켜서 주고받을 수 있다. 또한 자바는 애플릿(Applet)과 애플리케이션(Application)으로 나누어지는데 애플릿은 애플리케이션에 비해 프로그램의 크기가 작다는 장점이 있지

만 많은 제약 조건 - 서버에 파일 저장을 할 수 없고 보안상 허술하다 - 이 있다. 반면에 애플리케이션은 서버로의 역할을 하는 프로그램, 예를 들면 메일 서버 등과 같은 여러 클라이언트를 관리할 수 있는 역할이나 각각의 클라이언트들을 연결시켜주는 등의 복잡한 역할을 맡아 수행할 수 있다는 장점이 있다.

본 연구에서는 클라이언트쪽에는 애플릿을 서버 쪽에는 애플리케이션을 사용하여 클라이언트는 프로그램을 빠르게 다운 받을 수 있도록 하였고 서버는 복잡한 시뮬레이션을 처리해야 하기 때문에 애플리케이션을 이용하였다.

3.2. 자바의 장점

분산 환경하에서의 웹기반 시뮬레이션을 자바로 구현한 이유는 컴퓨터의 운영체제와 상관없이 구동 가능하기 때문이다. 또한 웹상에서 시뮬레이션을 구현하는 데 자바를 사용하는 다른 주요 원인을 나열하면 다음과 같다[2].

- 1) 시뮬레이션 모델의 효과적인 재사용이 가능하다.
- 2) 기존의 툴보다 -시뮬레이션을 하기 위하여 서버에 접속하는 것이 용이하다.
- 3) 사용자 인터페이스가 용이하다.
- 4) 자바를 구동하는 웹 브라우저로 어느 곳에서나 사용할 수 있다.
- 5) 자바는 자체가 네트워킹을 위한 IP(Internet Protocol)를 지원한다.

4. 웹기반 시뮬레이션의 기존 연구

본 장에서는 지금까지 개발된 웹기반 시뮬레이션 프로그램을 간략한 구현 원리와 함께 몇 가지만 소개하고자 한다.

4.1. DIGSIM

DIGSIM은 디지털 회로에 대한 시뮬레이터로 자주 사용되는 모델을 자체 프로그램상에서 지원울 해 주고 편집 기능까지 제공해 준다. 또한 복잡한 전자 회로를 시뮬레이션 함으로써 짧은 시간에 최적의 해

를 제시하고 회로 구조를 이해하는 데 도움을 준다 [12].

4.2. Simkit

Simkit은 이산 사건 시물레이션을 모델링하기 위한 것이며, 크게 3가지의 자바 패키지로 되어 있다. 모듈 collections의 경우는 데이터 구조를 지원해 주고 모듈 g2d는 차트등의 그래픽과 관계된 것을 사용자가 쉽게 이용하도록 한다. 또한 Simkit의 하위 구조에 있는 util, Javsim, awt는 각각의 특징적인 클래스를 지원함으로써 웹 상에서 시물레이션이 가능하도록 한다[3].

4.3. Simjava

Simjava는 복잡한 시스템의 모델을 설계하는 데 사용할 수 있는 프로그램으로 작은 애플릿 단위의 클래스로 구성되어 있다. 각각의 것들을 실행시켜 데이터를 입력받기도 하고 drag and drop 방식으로 모델링을 하기도 하고 또한 실행하여 보여주는 역할을 각각 수행하게 된다. [8, 11]

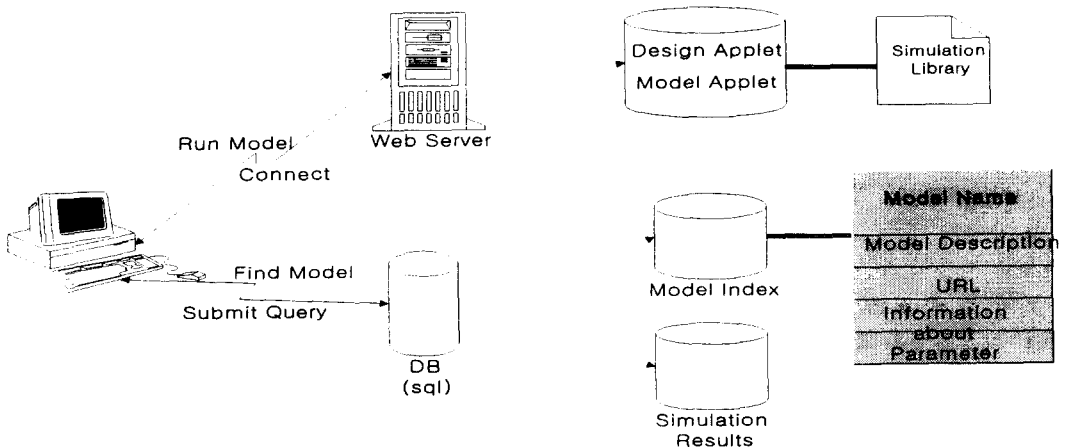
4.4. Jsim

Jsim은 은행시스템 형태의 문제를 시물레이션 하기 위한 모델을 쉽게 만드는 데 도움을 주는 프로그램이다. 그리고 데이터베이스에 저장된 내용들을 질의어를 통하여 쉽게 알고자 하는 경우에 데이터나 모델에 접근이 용이하도록 도와준다[9].

위의 Jsim 등과 함께 소개된 JavaBeans로 프로그램한 Silk라는 graphical modeling을 지원하는 프로그램도 있고, Symantec Visual Café, Microsoft J++, Borland JBuilder 등 다양한 툴로 웹기반 시물레이션 S/W를 개발하고 있다[5, 6].

4.5. 기존 웹기반 시물레이션 S/W의 구조 및 단점

기존 웹기반 시물레이션 S/W의 구조는 그림 1로 표현할 수 있다. 즉 기존 연구의 구조는 클라이언트가 시물레이션을 필요로 하여 서버에 접속하게 되면 서버는 요청한 클라이언트에게 시물레이션에 필요한 시물레이션 엔진을 전송하게 된다. 그 후 클라이언트는 전송 받은 시물레이션 엔진을 가지고 자기가 원하는 모델을 설계한 후 실행을 하게 되면 결과치를 클라이언트에게 보여 주게 된다. 이러한 구조



<그림 1> 기존 Web-based simulation의 구조

에서 서버가 하는 역할은 단지 클라이언트의 요구가 있을 때 필요한 시뮬레이션 엔진을 전송하는 역할만을 하면 된다.

또한 만일 다른 클라이언트와의 메시지 교환을 원한다고 해도 메시지를 교환할 수 있는 방법이 없다는 단점은 가지고 있다. 이러한 단점을 나열하면 다음과 같다.

- 1) 클라이언트-클라이언트 간의 메시지 교환이 극히 제한적이다.
- 2) 클라이언트들은 서버로부터 필요한 애플릿을 다운받아서 실행하기 때문에 다른 클라이언트들과의 동시성(분산 시뮬레이션 환경)을 유지하기 어렵다.
- 3) 애플릿의 단점인 보안상의 문제가 있다.

5. 웹기반 시뮬레이션 S/W의 설계 및 구현

5.1. 구현 환경

본 연구에서 개발한 웹기반 시뮬레이션 S/W의 구현 환경은 다음과 같다.

- Web 서버 : Window NT 4.0, Pentium 166
- System : Window 98, Pentium 133
- RAM : 32MB
- S/W tool : Symantec Visual Cafe dbDE 2.5

개발한 S/W를 자바로 구현한 이유는 컴퓨터의 운영체제와 상관없이 구동 가능하기 때문이다.

5.2. 서버 애플리케이션

분산 환경하에서의 웹기반 시뮬레이션은 기존 stand-alone 환경에서의 단점을 보완하고 시뮬레이션을 수행할 때 걸리는 수행 시간의 단축을 위해 분산 개념을 도입하였다. 또한 서버의 효율과 multithread를 사용하여 여러 명의 클라이언트가 접속, 수행에 용이하도록 애플리케이션을 사용하였다.

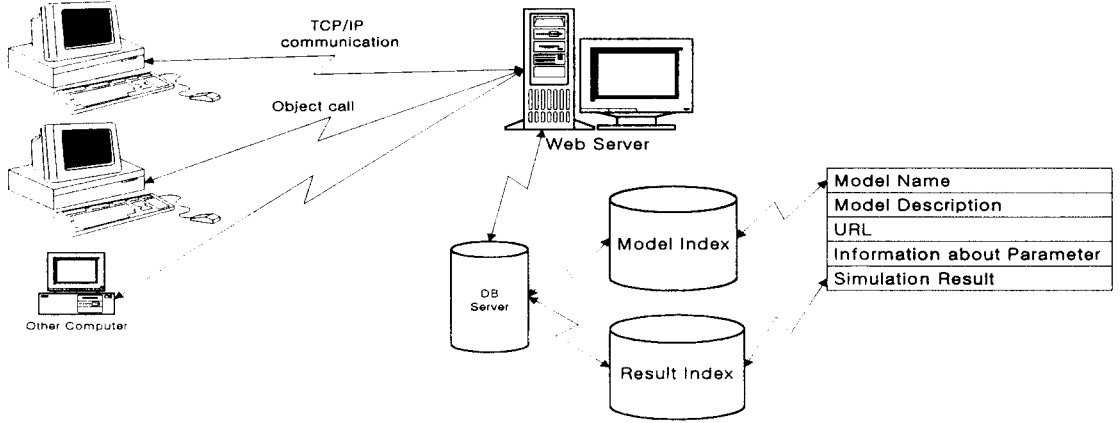
자바 프로그램을 작성할 때 크게 두 가지 방법이 있다. 첫 번째는 애플릿을 이용하는 방법이고, 두 번째는 애플리케이션을 이용하는 방법이다. 웹기반

시뮬레이션을 구현하기 위해서는 위의 두가지 방법이 모두 유효하다. 하지만 클라이언트와 서버간의 통신을 함에 있어 데이터와 모듈을 상호 교환을 해야하는 데 이때 만일 애플릿인 경우에는 보안상의 이유로 구현이 불가능한 경우가 생긴다. 또한 애플리케이션의 경우 분산 환경을 위한 RMI의 사용을 가능하게 해 주기 때문에 분산 시뮬레이션을 위한 프로그램을 작성하기 위해서는 애플리케이션을 반드시 사용해야 한다.

본 연구에서 사용한 분산 환경하에서의 웹기반 시뮬레이션의 특징을 정리하면 다음과 같이 설명할 수 있다.

- 1) 애플리케이션 사용으로 분산 환경을 위한 웹 서버의 역할 확대
 - 시뮬레이션 Thread를 실행함으로써 다중 클라이언트와의 시뮬레이션 수행을 가능하게 함
 - 각각의 클라이언트들이 다른 모델을 수행함을 허용함
- 2) 분산환경에서의 메시지 전송은 객체 단위로 전송
- 3) 클라이언트를 위한 애플릿, 서버를 위한 애플리케이션을 사용
- 4) 클라이언트-클라이언트간의 메시지 교환 및 객체 호출을 위한 모듈 (IP 주소, JDBC, RMI) 사용

그림 2는 본 연구에서 사용한 분산 환경하에서의 웹기반 시뮬레이션의 구조이다. 기존의 분산 시뮬레이션 구조와 비교하면 기존 방법의 경우 각각의 컴퓨터에 시뮬레이션 엔진을 가지고 시뮬레이션을 수행하는 반면에 그림 2에서의 구조는 각각의 클라이언트 컴퓨터에는 시뮬레이션을 수행하기 위한 엔진이 없지만 필요할 때 웹 서버에 있는 시뮬레이션 엔진을 호출하여 사용한다는 차이점이 있다. 즉 클라이언트가 시뮬레이션을 수행하고자 할 때 웹 서버에 접속한 후 입력할 모수들을 입력하고자 할 때는 서버는 입력에 관한 객체를 클라이언트에게 전송하고, 시뮬레이션을 수행하고자 할 때는 클라이언트 쪽에서 시뮬레이션 객체를 호출하면 서버는 그에 맞는 객체를 전송해 주게 된다. 이러한 절차로 수행하는 점이 기존의 분산 시뮬레이션과는 차이가 있다고 할 수 있다.



<그림 2> 분산 환경하에서 Web-based simulation의 구조

5.3. 개발된 웹기반 시물레이션 S/W의 구조

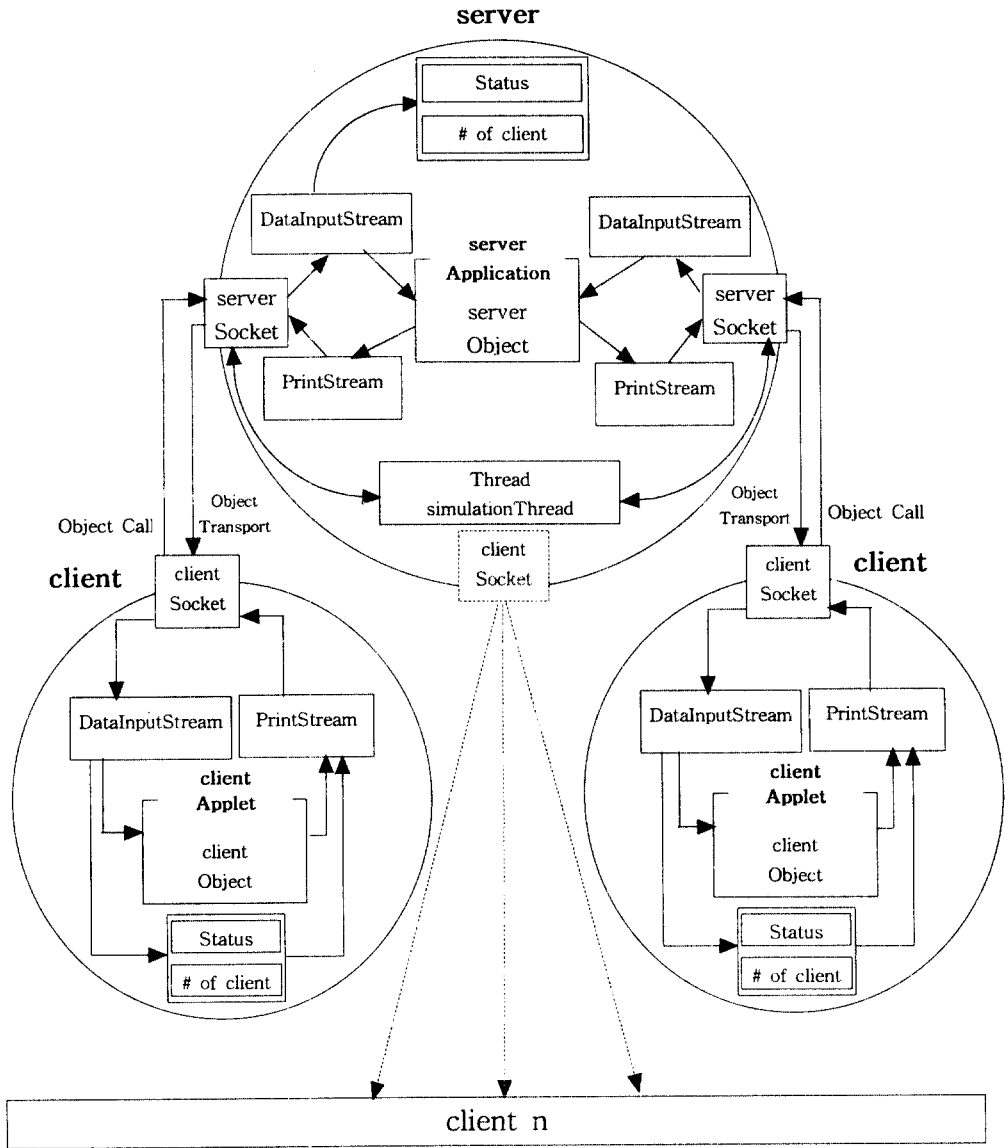
그림 3은 본 연구에서 개발한 S/W의 개략적인 전체 구조를 나타낸 것이다. 분산 환경하에서 웹기반 시물레이션을 구현하기 위한 것으로 클라이언트와 서버간의 메시지 전송은 객체 단위로 이루어진다. 그림 3의 내용을 절차화하면 다음과 같다.

- 단계 1. 클라이언트 1이 서버에 접속한다.
- 단계 2. 서버는 연결되었음을 알려주고 초기화면 애플릿을 보내 준다.
- 단계 3. 클라이언트 1은 초기 화면에서 시물레이션에 필요한 모수들을 입력한다.
- 단계 4. 클라이언트 1이 시물레이션을 수행하기 위해 필요한 서버의 시물레이션 객체를 요청한다.
- 단계 5. 클라이언트 1의 요청을 받은 서버는 시물레이션 객체를 보내고 분산 시물레이션을 수행하기 위해 접속되어 있는 다른 클라이언트들에게 시물레이션 객체와 클라이언트 1에서 입력한 모수들을 데이터 전송 객체에 포함시켜 전송한다.
- 단계 6. 서버는 각각의 클라이언트들에게 시물레이션이 끝났음을 알려 준다.
- 단계 7. 각각의 클라이언트들과 서버가 수행한 결과를 서버에서 모아서 결과 분석 과정을 통해

얻은 결과 값을 결과 창에 보여 주고 필요시 결과를 서버에 저장한다.

그림 3은 기존 연구에 비해 서버의 역할이 커졌고 클라이언트의 경우 다른 클라이언트와의 메시지의 전송도 서버에 의해 가능하게 되었다. 서버 애플리케이션은 클라이언트가 서버에 접속할 때마다 thread에 의해 서버 소켓을 만들고 접속한 클라이언트의 IP 주소를 기억한다. 그런 후 나중에 시물레이션 수행할 때 기억하고 있는 주소에 할당량을 배분하여 실행하도록 한다. 만일 작업이 10개가 있는 데 클라이언트수가 10명 이상일 경우에는 FIFO(First In First Out)에 의해 먼저 접속한 클라이언트에게 작업을 할당하여 준다. 또한 만일 10명보다 적은 7명일 경우에는 작업이 10개이므로 먼저 각각에게 1개의 작업을 할당한 후 3개의 작업을 FIFO에 의해 먼저 접속한 클라이언트에게 할당해 주는 방식을 채택하였다.

그림 3의 구조는 그림 4, 그림 5와 같은 초기화면에서 설계한 시물레이션 모델은 객체(Object), 그래프, Workload 등의 모수(parameter)들로 구성된다. 이러한 모수들은 발생기(Generator)를 통해 시물레이션 라이브러리와 연결, 그림 6과 같은 형식을 출력하게 된다. 초기 화면 및 결과 화면은 그림 4, 5, 6과 같이 구성되어 있다.

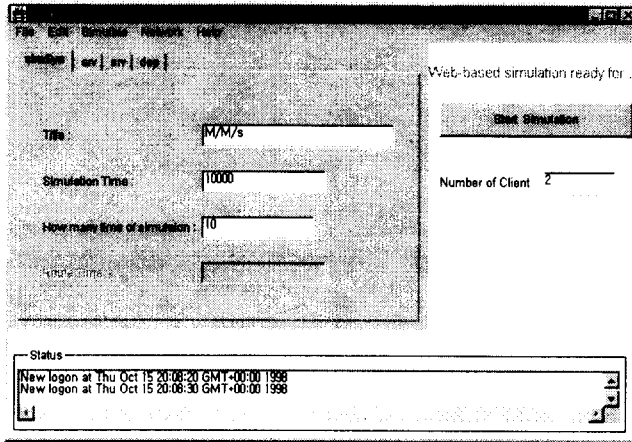


<그림 3> 분산 환경에서의 클라이언트와 서버 구조

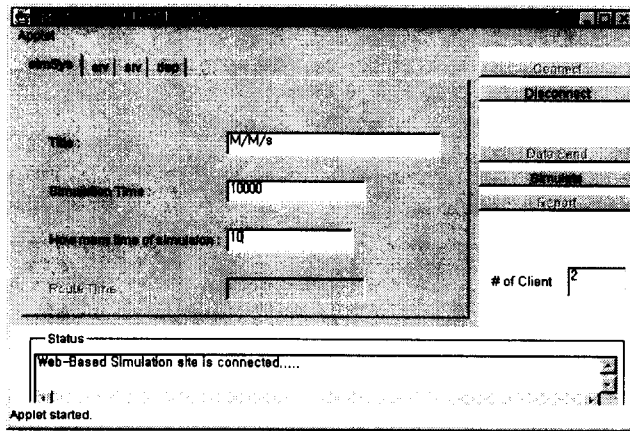
그림 7은 서버관련 구조로 크게 5개의 객체로 구성되어 있다. 5개의 객체는 각각의 역할이 있어서 클라이언트가 필요할 때 객체를 전송해 준다. 구체적인 설명은 다음과 같다.

- 시뮬레이션 객체 : 시뮬레이션을 수행하는 데 있

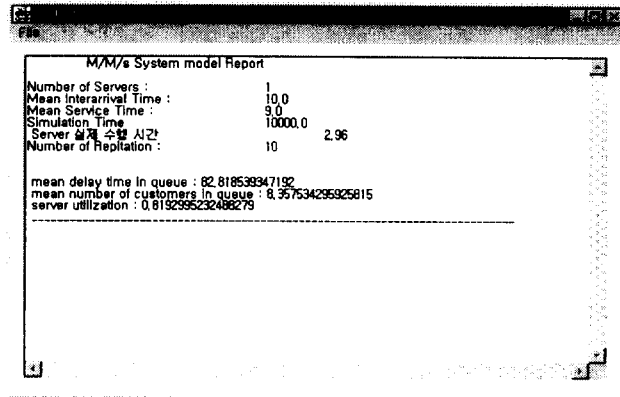
어서 필요한 기본적인 것들을 포함하고 있다. 또한 상세한 구조는 오른쪽 그림에 나와 있는데 웹상에서 모수로 입력되어 질 수 있는 3 가지 - 직접 입력, GUI상으로 디자인 하듯이 입력, 코드상에서의 입력 등 - 가 있는데, 이러한 모수들은 발생기를 통해 시뮬레이션에서 요구하는 코드로 변



<그림 4> 서버를 위한 초기화면



<그림 5> 클라이언트를 위한 초기 화면

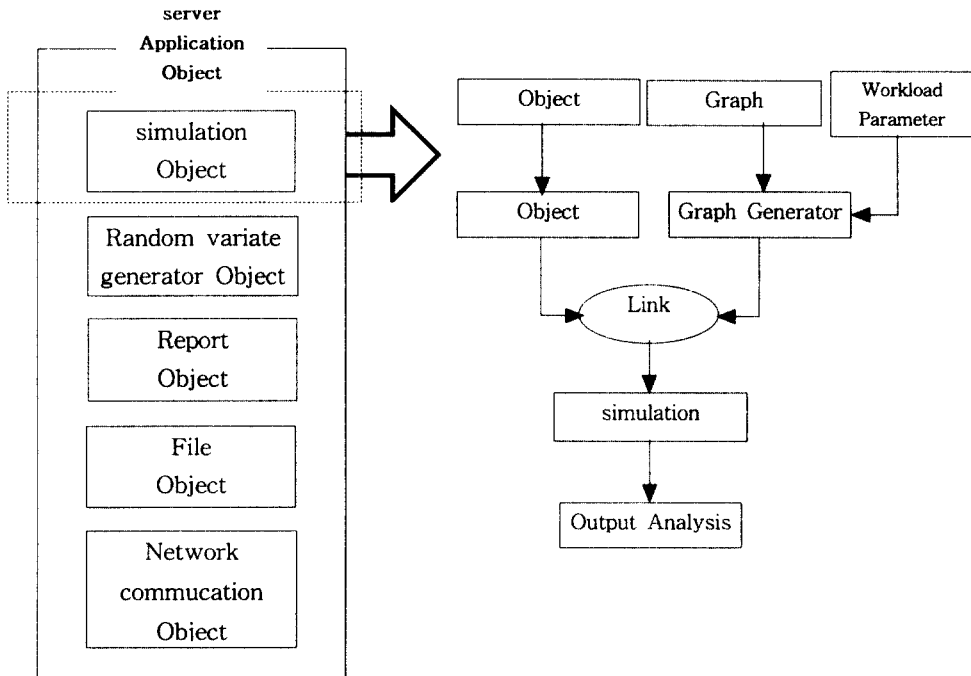


<그림 6> 시물레이션 결과를 위한 화면

형이 되고, 연결되어 시뮬레이션 라이브러리에 있는 여러 엔진들을 이용해서 결과치를 산출하게 된다.

- 확률 변수 발생기 : 지수 분포를 비롯한 이항 분포 등 필요한 확률 변수를 발생하게 된다.
- 레포트 객체 : 클라이언트나 서버가 시뮬레이션 결과값을 저장, 열기 등을 할 때 사용되는 객체로 클라이언트가 호출하게 되면 서버는 결과값을 객

그 이유는 많은 객체들을 가지고 있으면 서버로부터 객체들을 다운 받아야 하는 데 이 때 시간이 많이 걸리기 때문에 비효율적이다. 그래서 클라이언트는 가장 필요한 object request object - 클라이언트에서 필요한 객체가 있으면 서버에 호출하기 위한 객체 - 와 서버와의 정보 교환을 위한 네트워크 커뮤니케이션 객체만을 가지고 있다가 필요할 때 필요한 객체만을 서버에서 호출하여 사용하는 방식을 채택하였다.

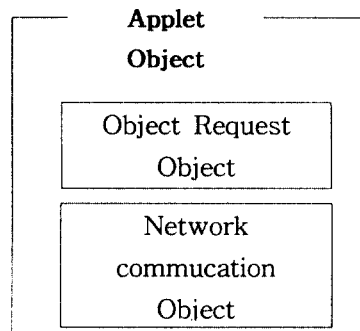


<그림 7> 서버 애플리케이션을 위한 구조

체에 포함하여 전송하게 된다.

- 화일 객체 : 과거에 모델링한 것은 JDBC에 저장되어 있으며, 화일 객체를 통해서 재사용이 가능하다. 클라이언트는 화일을 사용하기 위해서는 서버에 있는 애플리케이션에 허가를 받아야 한다.
- 네트워크 커뮤니케이션 객체 : 클라이언트와 서버간에 정보를 주고받고자 할 때 사용되는 것으로 이곳에서는 RMI를 사용하여 객체를 전송한다.

서버 관련 구조에 비해 간단한 클라이언트 구조 (그림 8 참조)는 가장 필요한 객체만을 가지고 있다.



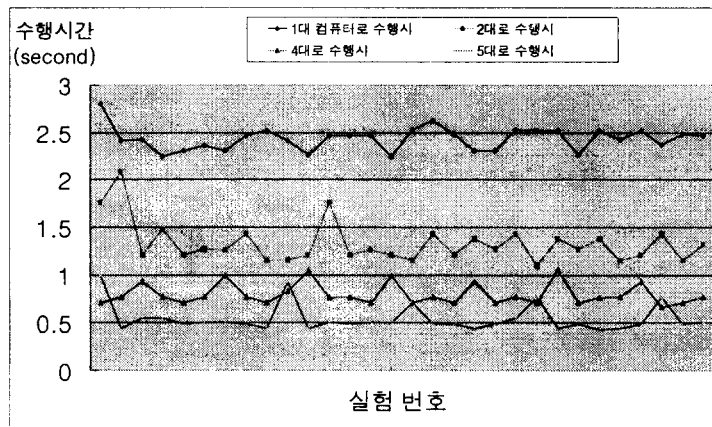
<그림 8> 클라이언트 애플릿을 위한 구조

5.4. 실험 및 평가

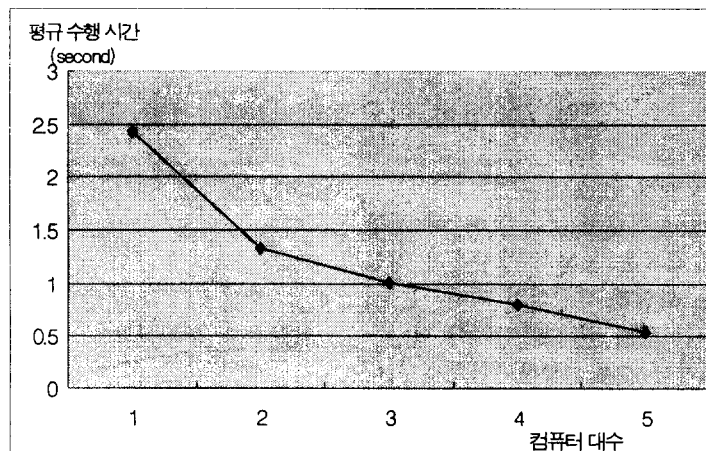
다수 서버를 가진 대기행렬시스템인 M/M/s 시스템을 대상으로 평균 도착 시간 간격을 10, 평균 서비스 시간을 9로 하여 실험을 하였다. 그림 6은 실험한 결과를 보여주는 그림이다. 또한 stand alone 컴퓨터와 분산 환경하에서 컴퓨터의 수를 늘려가면서 실험을 한 결과를 나타낸 것이 그림 9, 10이다.

그림 9와 10의 결과에서 보이는 것과 같이 분산 환경하에서 시뮬레이션을 수행할 때 컴퓨터의 수가 늘어나면서 수행시간이 단축되고 있는 것을 알 수 있

다. 즉 시뮬레이션을 수행할 때 많은 수의 클라이언트가 서버에 접속한다면 시뮬레이션을 수행하는 데 있어 걸리는 시간을 단축시킬 수 있다. 그러나, 클라이언트수가 많아지면서 서버에 걸리는 부하량도 커져서 시뮬레이션을 수행하는 시간이 늘어날 수 있다는 점도 유의를 해야 한다. 즉 시뮬레이션을 수행하는 데 있어 클라이언트에게 주어지는 모듈의 양과 서버에 걸리는 부하량을 비교해서 시뮬레이션 수행 시간을 줄일 수 있는 방법을 연구해야 할 것이다.



<그림 9> 수행 시간 결과치



<그림 10> 평균 수행 시간 결과치

6. 결론 및 향후 연구

본 논문에서는 기존에 개발된 웹기반 시뮬레이션들에 관한 내용을 기술하고 제안한 구조로 구현을 하고 간단한 실험을 해 보았다. 기존 연구에서 진보된, 그리고 기존의 분산 시뮬레이션의 개념과는 다른 분산 환경하에서의 웹기반 시뮬레이션은 모든 분야에 걸쳐서 무한한 가능성을 가지고 있는 연구 분야이다. 또한 많은 연구를 통해서 효과적인 정보 관리를 하고 모델링에 필요한 시간을 단축시키기 위해 JDBC를 이용한다면 보다 효율적으로 시뮬레이션 모델링이 가능하다.

추후 연구되어야 할 과제는 본 연구에서 구현한 M/M/s 시스템에 한정된 모델링이 아니라 보다 일반적인 모델들을 모델링 할 수 있는 다양한 라이브러리의 개발이 필요할 것이다. 이러한 라이브러리들의 사용을 정의하고 분산 환경을 위한 구조를 토대로 확장해 간다면 큰 문제를 시뮬레이션하는 데 있어서 기존의 방법보다 시간과 비용을 상당히 줄일 수 있다. 그리고 웹기반 시뮬레이션의 툴인 자바가 비주얼화 되어 가고 있기 때문에 향후에는 더 GUI가 향상되어 클라이언트가 쉽게 프로그래밍 할 수 있도록 해야 할 것이다. 이러한 추세는 지금은 기초 연구 단계에 머물러 있는 웹기반 시뮬레이션의 활용을 넓히는 데 큰 도움을 줄 것이다.

또한 웹기반 시뮬레이션에서는 서버에 있는 DB를 이용함으로써 클라이언트 측의 저장 공간을 줄일 수가 있으며 또한 모델링 시간을 단축할 수가 있다. 하지만 기존의 연구들은 클라이언트/서버의 개념을 이용했지만 많은 수의 클라이언트가 동시에 접속할 때 서버의 속도나 분산 시뮬레이션이 가능한 환경 등은 고려하지 못하는 실정이다. 앞으로 웹기반 시뮬레이션의 발전을 위해서는 클라이언트에 보내는 속도뿐 아니라 JDBC를 이용한 모델링의 신속화, 편리화를 위한 환경 구축에도 많은 연구가 필요하다고 본다.

그리고 웹기반 시뮬레이션을 하는 데 있어 서버의 객체 뿐 아니라 클라이언트에 있는 객체들, 그리고 자바 이외의 언어로 구현된 프로그램도 웹기반 시뮬레이션에서 구동이 가능하도록 하기 위한

CORBA의 연구와 궁극적으로 기존 시뮬레이션 소프트웨어보다 경제적이고 짧은 시간에 프로그램이 가능한 리소스를 제공함으로써 다른 사용자와의 코드 및 모델의 공유와 재사용이 가능하게 할 수 있도록 연구가 이루어져야 할 것이다.

참고 문헌

- [1] 이영해, 백두권, 시스템시뮬레이션, 경문사, 1991
- [2] 이영해, 광성근, "Web-based Simulation의 최근 연구동향 및 향후 연구방향", 97 추계학술대회 논문집, 133-138, 한국시뮬레이션학회, 1997
- [3] Buss, Arnold H. and Kirk A. Stock, "Discrete event simulation on the world wide web using Java", Proceedings of the 1996 Winter Simulation Conference, 780-785, 1996
- [4] Cubert, M. and Paul A. Fishwick, "Toward a Web-Based Modeling Methodology and Repository", <http://www.isima.fr/scs/websim/wsim35/websim.html>, 1997
- [5] Fishwick, Paul A., "Senior Project for Deborah Lynch", <http://www.cise.ufl.edu/~fishwick/dig/dlesp.html>, 1997
- [6] Healy, Kevin J. and Richard A. Kilgore, "Silk: A Java-based process simulation language", <http://www.isima.fr/scs/websim/wsim96/healy.html>, 1997
- [7] Iazella, Giuseppe and Andrea D'Ambrogio, "A Web-Based Environment for the Reuse of Simulation Models", Proceedings of the 1998 Western Multiconference on Computer Simulation, 765-772, San Diego, 1998.
- [8] McNab, R. and F. W. Howell, "Using Java for Discrete Event Simulation", Proceedings of Twelvth UK Computer and Telecommunications Performance Engineering Workshop (UKPEW), 219-228, Univ. of Edinburgh, 1996
- [9] Nair, Rajesh S., John A. Miller, and Zhiwei

- Zhang, "Java-based query driven simulation environment", Proceedings of the 1996 Winter Simulation, 786-793, 1996
- [10] Fishwick, Paul A., "Web-based Simulation: some personal observations", Proceedings of the 1996 Winter Simulation Conference, 772-779, 1996
- [11] SimJava, <http://www.dcs.ed.ac.uk/home/hase/simjava/simjava-1.0>, 1998
- [12] DIGSIM, <http://www.lifl.fr/~bogaert/DigSim10b3/DigSim.html>, 1998

● 저자소개 ●



이영해

1977년

고려대학교 산업공학, 학사

1983년

Univ. of Illinois, 산업시스템공학, 석사

1986년

Univ. of Illinois, 산업공학 및 경영과학, 박사

경 력

한국시뮬레이션학회, 대한산업공학회, 한국경영과학회, 대한설비관리학회 이사, Purdue Univ., 오사카대학 방문교수, 대우중공업(주)

1986년 - 현재 한양대학교 산업공학과 교수

1995년 - 현재 한국시뮬레이션학회 부회장

관심 분야 Web-based Simulation, Simulation Output Analysis, Simulation Optimization



곽성근

1997년

한양대학교 산업공학 학사

1999년

한양대학교 산업공학 석사

관심 분야

Web-based Simulation



김숙한

1985년

육군사관학교 졸업

1991년

미 해군대학원(NPGS), 운영분석, 석사

현 재

한양대학교 산업공학과 박사과정

관심 분야

시뮬레이션, SCM, ILS