

확장성을 갖는 웹 기반 분산 시뮬레이터의 구현 및 설계

김종은

현재 단국대학교 정보컴퓨터과학부 강의전임강사

semico@dankook.ac.kr

Design and Implementation for A Scalable Web-Based Distributed Simulator

JongEun Kim

요약

웹을 통하여 모델을 생성하고, 시뮬레이션을 수행하는 새로운 자바를 이용한 웹 기반 시뮬레이션 환경들이 제시되고 있다. 웹 기반 시뮬레이션은 인터넷에 연결된 많은 컴퓨터를 협동적으로 이용할 수 있다는 장점이 있지만, 웹 서버와 자바의 원격호출(RMI: Remote Method Invocation)이 이용되는 환경에서는 매우 많은 시뮬레이션 개체들간의 상호 동작과 정보 공유가 서버의 과다한 처리량과 통신 부하를 초래하여 시뮬레이션의 성능과 확장성을 저하시킬 수 있다.

본 논문에서는 확장성과 성능을 고려한 웹 기반 시뮬레이터를 설계하고, 객체 지향 언어인 자바를 사용하여 구현하였다. 또한, 제안된 시뮬레이터가 온라인게임에서 클라이언트와 서버 엔진의 모델을 생성하기 위해 활용될 수 있다. ATM 스위칭 모델의 시뮬레이션을 통해 구현된 시뮬레이션 환경의 활용을 제시한다.

Abstract

Many web-based simulation environments have been proposed to create models and execute simulation through the Web. Although web-based simulation has the advantage of utilizing many computers connected to Internet, overhead of the web server and heavy traffic caused by Java RMI degrade the performance and scalability of simulation.

In this paper, we design and implement web-based distributed simulator using Java for the scalability and performance. In addition, we can apply the proposed simulator to create model server and client engines in online game. We apply the proposed simulator to simulate ATM switch queueing model.

1. 서론

실존 시스템이나 이론적으로 존재하는 시스템을 수학적 방법으로 모델링하거나 분석하기가 어려울 때 컴퓨터 시뮬레이션은 유용한 방법으로 알려져 왔다. 시뮬레이션 모델이 대형화되고 시뮬레이션 결과를 실시간으로 연이어 하는 상황에서는 기존의 순차적 시뮬레이션 방법으로는 문제를 해결할 수 없게 되었다. 시뮬레이션 대상을 여러 프로세서들 또는 컴퓨터들에서 나누어 수행하는 병렬 시뮬레이션과 분산 시뮬레이션은 이러한 상황을 해결하기 위한 방

법으로 개발되어 연구가 이루어지고 있다[2]. 최근에 인터넷 활용 중에서 특히 웹은 지난 수년간 급속도로 성장하였다. 웹의 근본적인 특성은 분산 속성이며 웹 서버와 클라이언트(브라우저)의 상호작용으로 동작한다. 이러한 웹의 특성과 인터넷으로 연결된 컴퓨터들을 시뮬레이션 분야에 응용한 웹 기반 시뮬레이션의 구현에 대한 연구가 활발히 이루어지고 있다[1].

웹의 특성을 이용한 웹 기반 시뮬레이션의 발전과 더불어, 다른 종류의 다양한 시뮬레이터들을 통합하여 시뮬레이션을 확장하고자 하는 분산된 상호 동작 시뮬레이션(DIS :

Distributed Interactive Simulation) 분야의 연구가 활발히 진행되고 있다.

자바를 이용한 웹 기반 시뮬레이션은 웹을 통하여 시뮬레이션 모델을 생성하고, 시뮬레이션을 수행하는 새로운 환경들이 제시되고 있다.

또한, 시뮬레이션 구현기술을 게임 엔진에 응용하고자 하는 연구가 진행되었다. 게임 엔진은 그래픽과 사운드 등의 소프트웨어 기술을 결합해 게임에 적용하기 쉽도록 미리 만들어 놓은 핵심 프로그램이다. 온라인 게임은 클라이언트/서버 구조로 클라이언트와 서버는 각각의 엔진을 이용하여 게임진행을 위한 다양한 프로세싱을 하고 필요한 데이터를 분산하여 처리한다. 클라이언트/서버 기반의 시뮬레이터 기술은 온라인 게임의 클라이언트/서버 엔진을 구현하는데 이용될 수 있다.

본 논문에서는 웹 기반 시뮬레이터를 확장성과 성능을 고려해 설계하고, 실 예를 통해 구현된 시뮬레이션 환경의 활용을 제시하고자 한다. 또한, 제안된 시뮬레이터의 모델이 확장성과 재사용성을 고려한 게임엔진을 구현하기 위해 활용될 수 있다. 이를 위하여 2장은 시뮬레이션의 기법, 게임엔진 기술에 관하여 고찰하고, 3장에서 본 논문이 제안하는 시뮬레이터의 설계 및 게임엔진의 설계를 설명한다. 4장에서는 실제 구현된 시뮬레이터에서 통신 부하에 대한 모델을 적용하여 시뮬레이션 활용을 제시하고, 5장에서 결론으로 마무리한다.

2. 관련 연구

2.1 병렬 및 분산 시뮬레이션

시뮬레이션 환경은 시뮬레이션이 거대해지고 시뮬레이션 결과를 실시간으로 얻어내야 하는 경우가 있다. 이와 같은 상황을 만족시키기 위해 분산 및 병렬 시뮬레이션에 대한 연구가 활발하다. 이러한, 병렬 및 분산 시뮬레이션에서의 초점은 감소된 시뮬레이션 시간, 지리적 분산, 시뮬레이터의 통합, 오류의 처리에 있다[5].

병렬 시뮬레이션은 분산 시뮬레이션의 다른 형태중의 하나이며, 병렬성을 갖춘 기계에서 시뮬레이션 프로그램을 동시에 수행한다.

분산 시뮬레이션은 시뮬레이션 프로그램을 서로 다른 컴퓨터들로 분산시키고 동시에 수행한다. 병렬 및 분산 시뮬

레이션에서 시뮬레이션 프로그램을 수행하는 논리적 개체를 논리 프로세스라고 한다. 시뮬레이션 시간(Simulated Time)의 진행과 표현에 따라 동기 방식과 비동기 방식으로 구분되는데, 동기 방식에서 논리 프로세스의 시뮬레이션 시간은 전체 시뮬레이션 시간(Global Time)이 되고, 비동기 방식에서 논리 프로세스들의 시뮬레이션 시간은 서로 다르다. 동기 방식에서 논리 프로세스들의 시뮬레이션 시간들은 동일한 시간을 유지해야 하고, 이것은 비동기 방식보다 비효율적일 수 있다.

그러나, 비동기 방식은 인과오류(casuality error)가 발생할 수 있다는 문제를 갖는다. 이러한, 인과 오류에 대한 해결 방법으로 낙관적인 방법과 보수적인 방법이 연구되어왔다. 보수적 방법은 인과 오류를 허용하지 않는 방식이며, 논리 프로세스들이 안전한 메시지만을 처리하도록 보장되고, 시간 순서대로 처리되도록 보장되어 한다[4][6]. 낙관적 방법은 논리 프로세스들이 서로 다른 시뮬레이션 시간을 가지는 것을 허용한다. 즉, 논리 프로세스들 사이에서 전달되는 메시지의 발생시간(TimeStamp)순서는 어긋날 수 있다. 이러한 동기화 문제를 해결하기 위해 보수적 기법은 Null 메시지, 낙관적 기법은 Time-Warp, Time bucket 등 많은 기법들이 제안되었다.

위에서 제시된 논리 프로세스 구조는 게임엔진의 데이터 처리와 동기화 방법으로 활용될 수 있다.

2.2 DIS(Distributed Interactive Simulation)

DIS는 다른 종류의 시뮬레이터들을 상호 동작하도록 하는 구조이고, 시뮬레이터의 가상 세계는 발생된 사건에 의해 상호 동작하는 개체들로 이루어진다. 하나 이상의 개체들로 이루어진 시뮬레이터는 독립적으로 시뮬레이션을 수행하면서 다른 시뮬레이터와 연관된 동작과 특성을 보고한다. 분산된 상호 동작 시뮬레이션의 장점은 네트워크를 통해 독립적인 시뮬레이터를 연결해서 거대한 시뮬레이션을 만들 수 있는 구조를 제공한다.

DIS의 시뮬레이션 모델은 네트워크를 통해 수많은 컴퓨터가 연결된다. DIS의 설계 원칙은 시뮬레이션 노드의 독립성(시뮬레이션을 수행하는 노드는 동일한 알고리즘으로 동작해야 된다.)을 보장하여야 한다. 또한, 사실 정보 전송(시뮬레이션 노드의 실행은 사실을 전달한다.)이 있어야 하며, 상태 변화 정보 전송(전송된 정보는 반드시 시뮬레이션

상태를 변경하도록 하는 정보이다.)이 필요하며, 시뮬레이션 시간 제약 등이 있다[7].

DIS 특성은 확장성과 재사용이 가능한 게임엔진의 구현 기법에 적용될 수 있다. 또한, 개발된 게임엔진은 다양한 게임에 적용이 가능하게 되므로 게임 개발자는 게임 컨텍스트 개발에 집중할 수 있다.

2.3 웹기반 시뮬레이션

WWW(World Wide Web)과 자바(Java)의 등장으로 확장성 있는 분산 시뮬레이션의 구현이 쉽게 되었다. 어떠한 사용자들도 웹을 통해 분산된 시뮬레이션 모델을 자바가 수행이 가능한 브라우저로 수행 할 수 있다[3]. 웹기반 시뮬레이션은 1996년 이후에 본격적으로 개발되었으며 특징은 다음과 같다.

- (1) 시뮬레이션 모델은 웹 문서, 멀티미디어로 연결되어(linking)도달 할 수 있다.
- (2) 시뮬레이션은 웹의 다양한 사이트로부터 모델의 요소들이 모여 개발되어진다, 심지어 서로 다른 지역이라도 원격으로 개발될 수 있다.
- (3) 시뮬레이션 모델은 다양한 웹에서 실행될 수가 있다.

이런 3가지 요소는 마침내 웹기반 시뮬레이션 모델의 개발을 가능하게 한다. 웹 기반 시뮬레이션의 플랫폼에 독립적인 특징은 모바일 게임을 개발하는데 응용될 수 있다.

2.4 온라인 게임의 엔진기술

게임 개발하기 위한 핵심기술로 네트워크 처리기술을 포함하여 서버 엔진기술과 클라이언트 엔진 기술로 구분된다. 서버의 엔진은 클라이언트들의 데이터 처리, 메시지 전달, 동기화 등의 주요 프로세싱을 담당한다[8].

- ① 객체지향 엔진 개발 기술(엔진의 확장성 및 재사용성의 극대화)
- ② 데이터 처리 기술(클라이언트 메시지의 처리 및 대용량 DBMS의 이용)
- ③ 분산처리 기술(서버의 부하 분산과 서버기능 극대화)

3. 시뮬레이터의 설계 및 구현

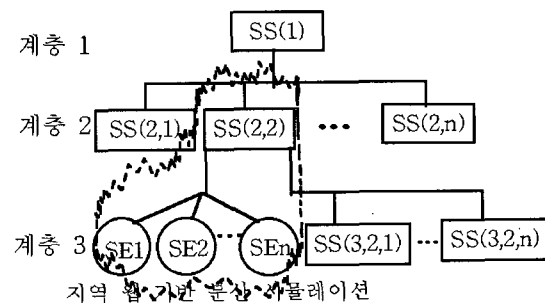
3.1 시뮬레이터의 확장적 구조

시뮬레이터는 구조적 확장성, 사용하기 편리한 인터페이스와 범용적 특성을 가져야 한다. 또한, 시뮬레이션 모델은 재사용이 가능하도록 모델화 될 수 있어야 한다. 본 논문에서 제안하는 모델은 TimeWarp 기법의 논리 프로세스를 사용한 것으로 가정한다.

그림 1의 계층적 모델은 인터넷과 웹을 통해 연결된 많은 컴퓨터들을 분산 시뮬레이션에 이용하기에 적합하다. 계층적 모델에서 시뮬레이션 개체들과 시뮬레이션 서버들 간의 계층은 지리적인 위치에 따른 결정이 아니라 연결 설정에 따른 논리적인 계층이다. 이러한 논리적인 계층은 인터넷과 웹을 통해 이루어진다.

시뮬레이션 개체(SE: Simulation Entity)는 독립적인 시뮬레이션을 수행하는 개체이며, 시뮬레이션 개체들간의 상호 동작에 대한 관리는 시뮬레이션 서버(SS: Simulation Server)가 수행한다. 즉, 시뮬레이션 서버는 크게 시뮬레이션 개체들간의 상호 동작을 관리하고, 상위 또는 하위 계층의 시뮬레이션 서버들간의 상호 동작을 관리한다.

시뮬레이션 개체들간의 통신은 반드시 자신을 관리하는 해당 시뮬레이션 서버를 통해서 이루어진다. 그러므로, 시뮬레이션 서버가 상호 동작이 빈번하고 외부사건의 발생 간격이 비슷한 시뮬레이션 객체들을 관리하도록 구성하고, 과도한 통신 부하를 유발하지 않도록 구성되어야 한다.



SS : 시뮬레이션 서버 SE : 시뮬레이션 개체

그림 1. 시뮬레이터의 계층적 구조

그림 2의 수평적 모델에서의 시뮬레이션 개체들간의 통신은 계층적 모델보다 적은 수의 시뮬레이션 서버를 통과한다. 그러나, 시뮬레이션 서버는 시뮬레이션 모델의 모든 시뮬레이션 개체들의 위치 정보를 알고 있어야 한다.

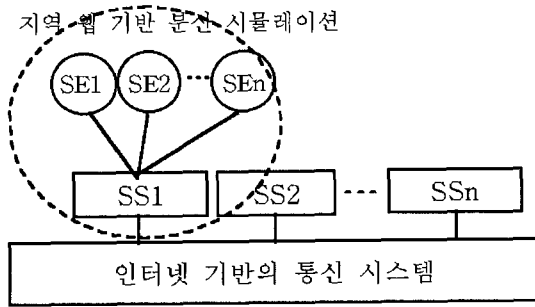


그림 2. 시뮬레이터 수평적 구조

지역 웹 기반 분산 시뮬레이션은 하나의 시뮬레이션 서버와 직접 연결된 시뮬레이션 개체들로 분산 시뮬레이션 수행의 기본 단위이다. 논리적인 모델에서 시뮬레이션 모델의 기본적인 단위는 시뮬레이션 서버, 웹 기반 통신 시스템, 시뮬레이션 개체들로 구성된 지역 웹 기반 시뮬레이션이다. 시뮬레이션 서버와 시뮬레이션 개체들은 웹 기반의 통신 시스템으로 연결되어 있고, 시뮬레이션 서버는 인터넷의 웹을 통해 지리적으로 분산된다.

위와 같은 모델들은 인터넷 환경을 이용해서 거대한 시뮬레이션 대상을 분산 수행할 수 있고, 논리적인 인터넷 연결에 대한 가상 모델이다. 본 논문에서 구현하는 웹 기반 분산 시뮬레이터는 시뮬레이션 대상과 환경에 따라 계층적 모델과 수평적 모델로 분산 수행하도록 설계하고자 한다. 또한, 이러한 분산 모델은 온라인 게임에서의 서버엔진의 부하를 감소시키고 기능을 극대화에 이용될 수 있다.

3.2 논리 프로세스의 구조 설계

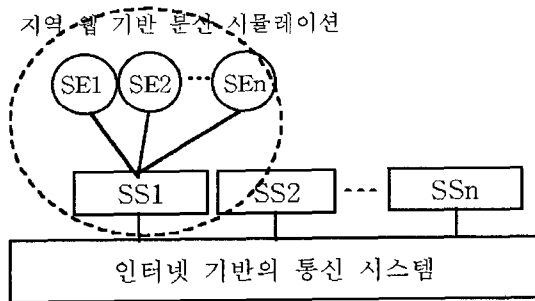


그림 3. 논리 프로세스 구조

시뮬레이션 개체의 모델은 비동기 방식의 Time-Warp 기법을 변형하여 확장성을 고려한 웹 기반 분산 시뮬레이션 모델에 적용한다. 그림 3에서 보이는 변형된 논리 프로세스 구조는 통신 인터페이스, 시뮬레이션 엔진, 자원으로 구성된다. 이것은 2장에서 설명된 논리 프로세스 구조와 동일하지만, 통신 인터페이스는 출력버퍼가 없고, 외부 사건의 송수신을 위해 GVCI와 LVCI라는 인터페이스가 존재한다.

외부 사건의 송신은 발생 즉시 GVCI를 통해 서버에게 전송한다. 외부 사건의 수신은 LVCI를 통해 서버에게 요청하고, 서버는 응답으로 요청한 논리 프로세스에 대한 외부 사건들을 반환한다. 시뮬레이션 엔진은 사건을 선택하고 처리하는 역할을 담당한다.

이와 같은 논리 프로세스 구조는 시뮬레이션 개체에서 독립적으로 시뮬레이션 서버와 메시지를 송수신하고 처리한다.

3.3 온라인 게임엔진에 대한 적용

3.1 절의 시뮬레이터의 확장적 구조는 온라인 게임에서 서버엔진의 부하 감소와 기능의 극대화를 위한 분산구조로 적용될 수 있고, 3.2 절의 논리프로세스 구조는 온라인 게임에서 클라이언트 엔진에 적용될 수 있다.

그림 4는 논리 프로세스 구조를 온라인 게임에서 클라이언트 엔진에 적용하기 위한 구조이다.

네트워크 인터페이스는 TCP/IP의 소켓으로 구성되고 게임의 진행과 게임 진행시간의 대화에 따른 사건의 송/수신을 담당한다. 이것은 논리 프로세스 구조의 네트워크 인터페이스의 기능과 같다. 엔진 관리자는 동기 관리자와 사건 관리자로 구성된다. 동기 관리자는 클라이언트들간의 게임 동작을 처리하는 방식에 따라 동기적 기법 또는 비동기적

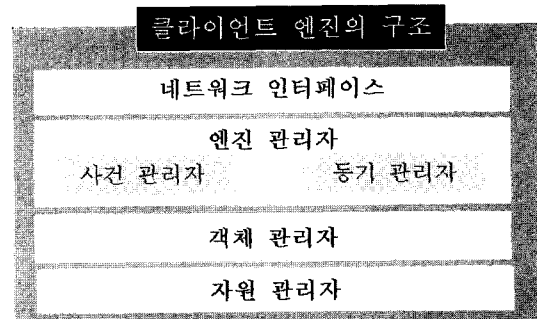


그림 4. 클라이언트 엔진의 구조

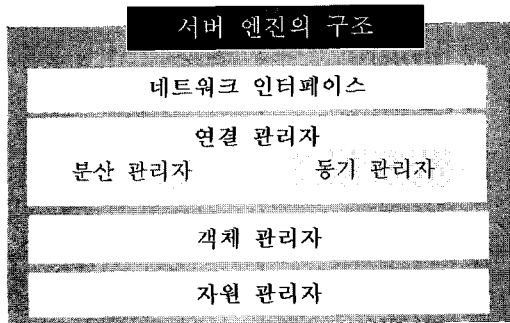


그림 5. 서버 엔진의 구조

기법(NULL 메시지, TimeWarp, time bucket 등)을 적용한다. 사건 관리자는 사건을 서버에 송수신을 위한 외부 메시지와 엔진에서 처리해야하는 내부 메시지로 구분한다. 또한, 객체관리자를 통하여 메시지의 처리를 위해 자원 관리자에게 전달한다.

자원 관리자는 게임의 진행 동작에 따라 그래픽 처리, 사운드 처리와 인공지능 처리 등을 담당한다. 그림 5는 서버의 엔진 구조로 연결 관리자는 분산 관리자와 동기 관리자로 구성된다. 서버 엔진은 클라이언트들간의 메시지에 대한 중재를 하고, 대용량 데이터 베이스를 이용하여 게임 진행에 따른 데이터를 처리한다.

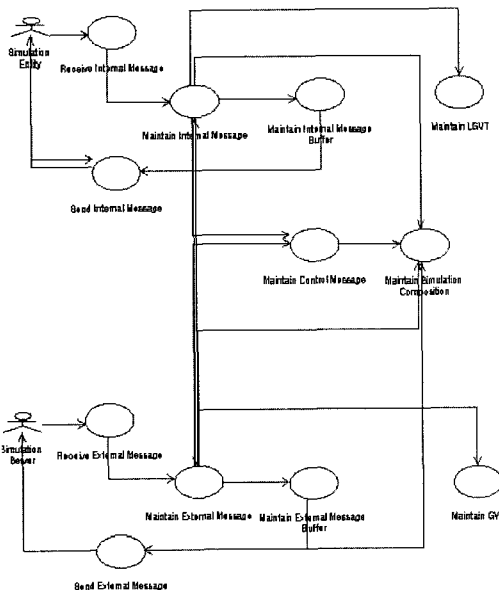


그림 6. 시뮬레이션 서버의 다이어그램

이와 같이 시뮬레이터의 엔진 기술을 게임 엔진의 구현에 응용함으로써 확장성과 재사용성을 향상시킬 수 있다.

3.5 구현된 클래스의 구조

시뮬레이션 서버의 기능에 대한 유스 케이스 다이어그램은 그림 6과 같다. 시뮬레이션 개체는 내부 메시지를 시뮬레이션 서버에게 전송하고, 시뮬레이션 서버는 수신한 내부 메시지를 내부 메시지 관리자에게 전달한다. 내부 메시지 관리자는 시뮬레이션 모델의 환경 설정을 제어 메시지 또는 상호 동작을 위한 외부 사건이다.

외부 사건의 도착 시뮬레이션 개체가 동일한 시뮬레이션 서버의 관리에 있는 시뮬레이션 개체이거나 다른 시뮬레이션 서버의 시뮬레이션 개체인 외부 메시지이다. 외부 사건은 해당 시뮬레이션 개체의 버퍼에 저장하고, 외부 메시지는 구성 관리자로부터 도착지의 시뮬레이션 서버의 패킷에 대한 정보를 얻어 전송한다. 제어 메시지는 제어 관리자에게 전송되고 처리된다.

시뮬레이션 개체는 시뮬레이션 서버에게 외부 사건을 요청하고, 시뮬레이션 서버의 내부 메시지 전송 관리자는 내부 메시지 버퍼로부터 내부 메시지를 시뮬레이션 개체에 반환한다.

4. 제안 시뮬레이터의 적용

본 장에서는 그림 7의 ATM 스위치에 대한 큐잉 모델을 본 연구에서 구현된 웹 기반 분산 시뮬레이터에서 적용하여 그 활용성을 제시한다. 스위치1, 스위치2 및 스위치3은 별도의 클라이언트에서 브라우저를 통하여 분산 수행되는 시뮬레이션 개체로 구현하였다.

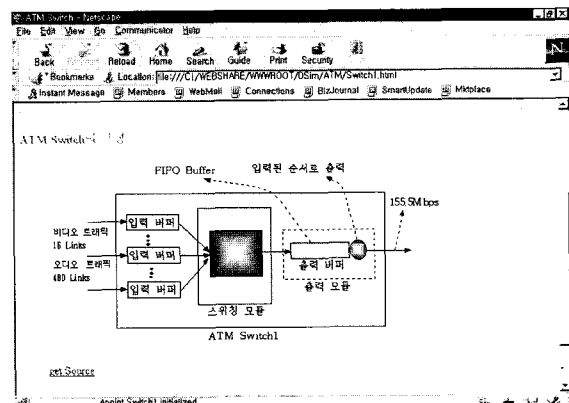


그림 7. ATM 스위치의 시뮬레이션 모델

본 연구에서 구현한 웹 기반 시뮬레이션을 적용하여 3대의 PC에서 시뮬레이션 개체를 수행하였다. 그림 8은 시뮬

레이션 개체가 수행하며 출력한 화면을 보인 것이다. 각 시뮬레이션 개체를 수행하는 PC에서는 해당 스위치의 시뮬레이션 수행 정보를 출력한다. 또한, ATM 스위치1에 대한 시뮬레이션 모델과 시뮬레이션 수행 결과로 얻은 ATM 스위치1의 셀 손실률과 셀 평균 지연을 10ms 단위로 출력한 것이다. 각 시뮬레이션의 개체의 수행 결과는 시뮬레이션 서버로 전송되어 DBMS로 저장된다.

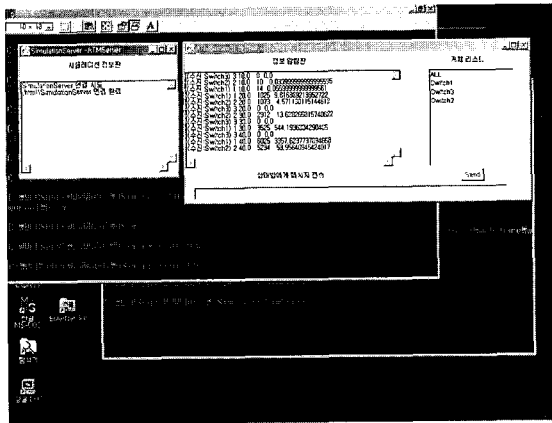


그림 8. 시뮬레이션 수행 화면

5. 결론

본 논문에서는 기존의 분산 시뮬레이션 또는 웹 기반 분산 시뮬레이션이 갖는 확장성의 문제점을 보완하여 확장적 특성을 갖는 웹 기반 시뮬레이터 모델을 제시하였고, 자바와 웹의 분산 특성을 이용하여 구현하였다. 시뮬레이션 기술은 게임엔진의 기반이 되는 기술이다. 제안된 웹 기반 시뮬레이터의 모델을 온라인 게임에 적용하여 확장성과 재사용성을 고려한 게임엔진을 설계할 수 있음을 설명하였다. 또한, 제안된 웹 기반 시뮬레이션 모델과 시뮬레이터 라이브러리들을 이용한 ATM 스위치에서의 큐잉 모델에 적용하였다.

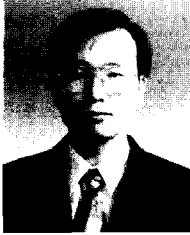
본 논문에서 제안하는 웹 기반 시뮬레이션 모델은 처리 시간의 개선을 목적으로 하는 멀티프로세서의 분산 시뮬레이션과는 달리 시뮬레이션 개체의 분산과 상호 운용에 목적을 두고 있고, 시뮬레이션 모델을 게임 엔진에 대한 적용에 있다.

향후, 제안된 모델은 처리 시간을 고려한 엔진 구현과 게임엔진의 컴포넌트를 위한 연구가 이루어져야한다. 또한, 제안된 게임 엔진을 이용한 온라인 게임의 구현하기 위한

연구가 한다.

참고문헌

- [1] John A Miller and Zhiwei Zhang, "Java-based query driven simulation environment", Procs. of the conference on Winter simulation, pp. 786-793, 1996.
- [2] Reuben Pasquini and Vernon Rego, "Optimistic Parallel Simulation Over a Network of Workstations", Procs. of the conference on Winter simulation, pp. 1610-1617, 1999.
- [3] Richard M. Fujimoto, "Parallel and Distribution Simulation", Procs. of the 1999 Winter Simulation Conference, pp. 122-131, 1999.
- [4] Richard M. Fujimoto, "Parallel and Distribution Discrete event Simulation", Procs. of the 1994 Winter Simulation Conference, pp. 106-114, 1994.
- [5] Sandra Cheung and Margaret Loper, "Synchronizing simulations in Distributed interactive simulation", Procs. of the 1994 Winter Simulation Conference, pp. 1316-1323, 1994.
- [6] 김종은, 조경산, "웹 기반 시뮬레이터의 구현," 단국대학교 제33집 논문집, 1998.
- [7] 이영해, 배강룡, "분산 시뮬레이션에서의 deadlock avoidance를 위한 효율적 알고리즘 개발에 관한 연구," 97 추계학술 논문집 한국시뮬레이션학회, pp. 1-6, 1997.
- [8] 최학현, 김정희, "온라인게임의 엔진기술 및 설계에 관한 연구", 한국게임학회 논문지 제 1권 1호, pp. 24-30, 2001.



김종은

1995년. 단국대학교 전자계산학과 이학사

1997년. 단국대학교 전산통계학과 이학 석사

2000년. 단국대학교 전산통계학과 이학박사 수료

2001년~현재 단국대학교 정보컴퓨터과학부 강의전임강사

관심분야: 분산시뮬레이션, 컴퓨터 구조, 네트워크, 분산처리, 온라인
게임엔진
