

웹을 이용한 분산 시물레이션 환경

정진립, 우영제, 정창성
고려대학교 전자공학과

Distributed Simulation Environment Using WWW

Department of Electronic Engineering Korea University

요 약

웹을 이용한 소프트웨어의 개발이 빠른 속도로 발전하고 있다. 이에 따라 시물레이션도 웹을 이용한 새로운 개발 분야로 발전하고 있다. 자바는 웹의 객체지향 언어로 새로운 분산 객체모델을 지원한다. 본 연구는 자바의 분산 객체모델인 RMI를 웹 기술과 통합하고, 분산 시물레이션의 기존 알고리즘을 사용하여 대규모의 개발 및 유지비를 요구하는 시물레이션의 개발에 웹을 이용하는 새로운 분산 시물레이션 환경을 제시하고자 한다.

1. 개 요

Web(World Wide Web)의 기술은 짧은 역사에도 불구하고 오늘날 급속도로 발전하고 있다. 이는 컴퓨터의 용량에 제한을 받지 않고 문서, 그림, 비디오, 음성과 같은 다양한 형태의 문서를 간단한 웹 브라우저를 통하여 쉽게 이용할수 있기 때문이다. 이러한 Web의 기술은 최근에 시물레이션 분야에도 강한 영향을 주고 있다.

Web-based simulation이란 시물레이션 분야와 웹의 통합을 의미하며, 시물레이션에 웹 기술을 이용하는 개념적인 표현이다. 이에 따라 시물레이션의 새로운 분야로 자리잡기 위한 개념 연구가 진행된 바 있다.[1]

분산 시물레이션 분야의 연구는 Chandy-Misra 이후 지금까지 매년 활동적인 연구가 이루어져 왔고 주로

순차적 시물레이션을 분산 및 병렬로 실행시켜 속도를 증가시키는 방향으로 연구가 이루어 졌다. 군사적 이용으로서 분산 시물레이션은 지난 10년간 미국방산업의 초점이 되어왔는데, 여기서의 관심은 속도문제가 아니라 상호 운용성에 있었다. 이러한 국방 관련 노력은 SIMNET으로 시작되어 DIS(Distributed Interactive), ALSP(Aggregate Level Simulation Protocol)로 발전하게 되었고, 미국방성에 의해 새로 개발된 HLA(High Level Architecture)가 시물레이션과 모델링의 표준으로 자리잡고 있다.[2]

본 연구는 시물레이션의 개발비, 운용비, 유지비가 많고, 대규모의 인적, 물적 자원의 이동을 요구하는 현재의 전술 시물레이션에 대해, 개발비용, 운용 및

유지 보수면에서 좀더 효율적인 새로운 시뮬레이션 환경을 제시하고자, 객체지향 언어인 자바를 이용한 Web-based simulation과 자바의 분산 객체모델을 지원하는 Remote Method Invocation를 이용하여 새로운 분산 시뮬레이션 환경을 제시하고자 한다.

2. 웹에 기반한 시뮬레이션

시뮬레이션에 대한 현재의 소프트웨어 개발 환경은 단일 Processor에서 순차적으로 실행되거나 병렬 컴퓨터 혹은 근거리망으로 연결된 워크스테이션들에 의해 분산 처리 하는 것이다. 이러한 환경은 많은 사용자를 요구하는 큰 규모의 시뮬레이션 수행을 위해 인적, 물적 자원의 대규모 이동과 시뮬레이션 수행을 위한 별도의 클라이언트 컴퓨터가 준비되어야 하므로 시뮬레이션 운영 및 유지, 보수면에서 많은 비용을 필요로 한다. 따라서 시뮬레이션에 웹 기술의 적용은 이러한 단점을 해결할 수 있는 새로운 방법이 될 수 있다.

웹 기술의 시뮬레이션 적용에 관련하여 연구한 Fishwick[1]은 시뮬레이션 분야중 다음과 같은 3가지 분야에 대하여, 웹 기술의 잠재적인 효과에 대한 연구를 실시한 바 있다.

첫째, 교육과 훈련면에서, 웹은 디스켓이나 CD-ROM의 능력을 초과하는 음성이나 비디오와 같은 자원집중적인 매개물에 대하여 거의 무제한적인 용량을 제공함으로써 학습 환경을 매우 상호작용적으로 할 수 있다. 둘째, 출판에 대해서, 웹은 출판물의 연구 결과를 제출, 심사, 보급시키는데 새롭고 편리한 메카니즘을 제공한다. 셋째, 본 연구에서 다루고자 하는 시뮬레이션 자체에 대한 것이다. 웹 문서는 전통적인 텍스트 문서 뿐만 아니라 비디오, 그림, 음성, 상호작용 시뮬레이션을 포함할 수 있다. 그렇다면 어떻게 시뮬레이션을 웹문서에 포함시킬 수 있겠는가?

자바는 이식성, 융통성, 보안성이 좋고 간단하면서도 동적인 능력을 가진 객체지향적이며, 하드웨어에 독

립적인 프로그래밍 언어이다.[3] 자바의 애플릿은 웹 브라우저에서 실행 가능하도록 HTML문서에 포함될 수 있는 간단한 자바 프로그램으로, 애플릿의 소스코드가 바이트코드 형태로 컴파일되어 브라우저로 이동한다. 따라서 간단한 웹 브라우저만 있으면 한번 작성된 애플릿 코드는 어디에서든 실행이 가능하므로 대 규모의 클라이언트-서버 형태의 시뮬레이션에 매우 효율적인 프로그래밍 언어라 할 수 있다.

3. 분산 시뮬레이션

분산 시뮬레이션은 다중 프로세스에 시뮬레이션을 분산하여 전체 실행 시간을 줄이는데 목적이 있다. 이는 분산된 프로세스가 비동기적으로 실행되는 이산사건 시뮬레이션에서 본질적으로 병렬성을 이용하므로써 달성된다. 따라서 분산된 프로세스간의 동기화는 분산 컴퓨팅 알고리즘에서 주요한 문제이다.

분산 시뮬레이션의 동기화 전략에는 크게 두가지가 있다. 첫째, Chandy-Misra의 보수적 알고리즘[4]은 전체 시뮬레이션을 통하여 사건들의 인과성 조건이 엄격히 지켜진다. 그러므로 시뮬레이션 실행동안 어떤 시간에도 프로세스에서 처리되는 사건들의 순서는 정확하게 유지된다. 둘째, 낙관적 알고리즘[5]은 프로세스들이 낙관적으로 사건들을 처리한 후 인과성 오류가 발생시 롤백을 통하여 잘못 처리된 사건을 무효화 시킨 후 사건의 시간 순서대로 다시 처리한다. 본 연구에서는 보수적 알고리즘을 이용한다.

분산 시뮬레이션은 하나의 모델이 몇 개의 부분모델로 분할되고, 부분 모델은 logical processes(LPs)에 의해 동시에 실행하게 되므로, *logical process simulation*(LP simulation)이라 볼 수 있다[6]. LP simulation은 하나의 컴퓨터 혹은 다른 컴퓨터에서 독자적으로 실행을 하며, 어떤 logical process가 다른 logical process에 영향을 미칠 수 있다. 따라서 logical process 사이에는 상호작용이 필요하며, 이는 각각이 채널로 연결되어 메시지의 교환을 통하여 상호작용을 하게된다. 이 때 메시지는 이전에 보낸 시

각보다 작지않은 TimeStamp를 가진 메시지를 보냄으로써 인과성 오류를 방지할 수 있다. 그러나 보수적 알고리즘에는 두가지 문제가 발생하는데, 메모리 소모로 인한 버퍼의 오버플로우와 교착상태의 발생이다. 지금까지 교착상태를 회피하거나 복구하기위한 많은 기술이 개발되어 왔다. 대표적인 알고리즘으로 Misra의 Null 메시지 방법이다.[7] Null 메시지는 시스템의 상태에 어떠한 영향을 주지 않고 단순히 시물레이션의 시간을 진행시키는 것으로, 어떤 프로세스에 대해서 실제 메시지가 없다는 것을 알려 그 프로세스가 교착상태에 빠지는 것을 방지한다.

4. 자바

4.1 Java / WWW

웹을 기반한 시물레이션의 개념은 자바가 등장함으로써 현실화 되고 있다. 자바는 웹을 위한 객체지향 중심 언어로서 웹에서 실행 가능한 코드를 가지는데, 이는 자바 이전의 정적이고 문자 중심의 환경에서 완전히 동적인 환경을 제공한다.

자바는 웹 기술에 중요한 발전 요소이며, 시물레이션에서는 특히 중요하다. 자바 이전에는 웹에서의 동적인 작용은 CGI에 주로 의존하였는데, CGI가 웹 서버에서 실행되므로 원격지의 웹 브라우저에서 동적인 환경을 표시하는 데는 제한이 되었다.

자바의 애플릿은 원격지 웹 서버로부터 전송되어, 자바의 바이트 코드를 인터프리터할 수 있는 웹 브라우저에서 실행되므로 약간의 속도 제한은 있으나 "이식성"이 매우 좋다. 시물레이션을 자바로 구현 시이점에 대하여 알아본다.[8]

첫째, 자바 애플릿으로 구현된 시물레이션은 광범위하게 사용될 수 있다. 즉, 웹 브라우저는 원격지의 시물레이션 모델을 실행시킬수 있다.

둘째, 웹서버로부터 애플릿을 다운로드 후 실행하므로 이식성이 좋다.

셋째, 브라우저에서 실행되는 자바 애플릿은 매우 높은동적인 환경을 제공한다.

넷째, 자바의 쓰레드는 상호작용 시물레이션의 구현을 쉽게한다.

다섯번째, 정교한 애니메이션을 쉽게 만들 수 있다.

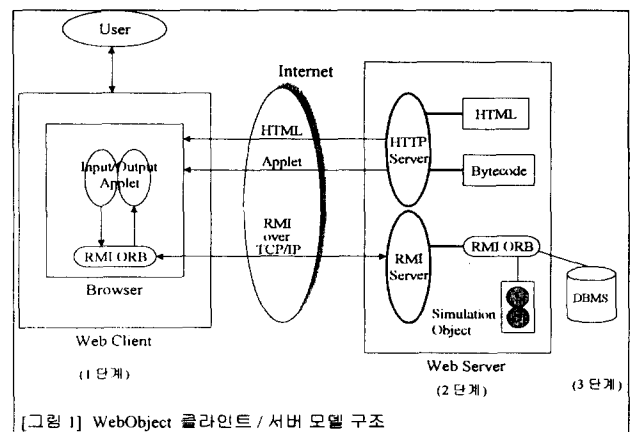
여섯째, C++ 보다 배우기 쉽고 안전하며, 규모가 적다.

4.2 Java / Client-Server Model

인터넷에서 클라이언트-서버 환경은 1993년 그래픽 웹 브라우저로 시작되어, 1995년 HTML/CGI로 발전하였고 1996년 분산 객체와 웹이 결합된 ObjectWeb 개념이 생기게 되었다.

[그림1]은 [9]에서 제시된 내용을 참조한 ObjectWeb 모델의 제3 단계 클라이언트-서버구조이다. 1 단계 클라이언트는 브라우저, 자바 애플리케이션, 애플릿이 여기에 속한다. 2 단계는 HTTP/RMI에 의한 클라이언트 요구를 서비스하는 서버가 되며, 3 단계는 전통적인 서버가 여기에 속한다.

RMI 객체는 2단계 서버로 RMI ORB를 통하여 인터넷상의 클라이언트와 상호작용을 하게된다. 이것은 자바 객체간의 통신에 대해서 RMI가 HTTP/CGI를 대신할수 있음을 의미한다. 따라서 HTTP는 애플릿,



이미지등을 다운로드하는데 사용하며, RMI는 자바 클라이언트-자바 서버간의 통신을 위해 사용된다.

본 연구의 분산 시물레이션과 관련시키면, 클라이언트

가 HTTP를 통하여 웹 서버로부터 HTML문서와 클라이언트용 입/출력 애플릿을 바이트코드 형태로 받

는다. 다음으로 브라우저의 입력 애플릿으로부터 사용자의 명령이 RMI를 통하여 RMI서버의 입력버퍼에 메시지가 전달되면 메시지가 처리되는 시물레이션 객체(서버)로 보내진다. 시물레이션 객체에서 메시지를 처리 후 그 결과는 데이터베이스의 자료를 수정하고, 다시 RMI서버 객체의 출력버퍼에 보내진다. 이 때 브라우저의 출력 애플릿은 RMI서버로부터 처리된 메시지를 받아 브라우저를 통하여 사용자에게 전달된다. 물론 시물레이션 서버들도 RMI서버가 되며, 서버간은 같은 컴퓨터 혹은 다른 컴퓨터에 있는 시물레이션 서버간에도 서로 통신을 하게된다.

4.3 Java의 분산 컴퓨팅 환경

자바 기술이 분산시스템 개발자들 사이에 많은 주목을 받아왔다. 이는 자바의 가상 머신, 이식 가능한 바이트 코드, 웹과의 통합이 가능하여 인터넷을 기반으로 하는 개발에 새로운 모델을 제시하기 때문이다. 오늘날 인터넷에서 3단계의 클라이언트-서버 모델은 아직 CGI/HTTP에 주로 의존하고 있는데, HTTP는 주로 어플리케이션 프로그램간 보다는 사용자와 서버간의 상호작용을 지원하고, CGI는 속도가 매우 늦고, 객체지향의 자바 클라이언트에는 적당하지 않다.[9]

이에따라 OMG(Object Management Group)에서는 새로운 분산 객체모델을 제시하였는데, 하나는 동일 네트워크 환경을 지원하는 자바 RMI(Remote Method Invocation)와 다른 하나는 이기종 네트워크 환경을 지원하는 자바 IDL(CORBA Interface Definition Language)이다. 두 방법은 각각의 장점을 가지고 있다.[10] 연구에서는 비교적 사용하기 쉬운 RMI를 이용한 분산 객체 모델을 적용하였다.

5. 분산 시물레이션 환경

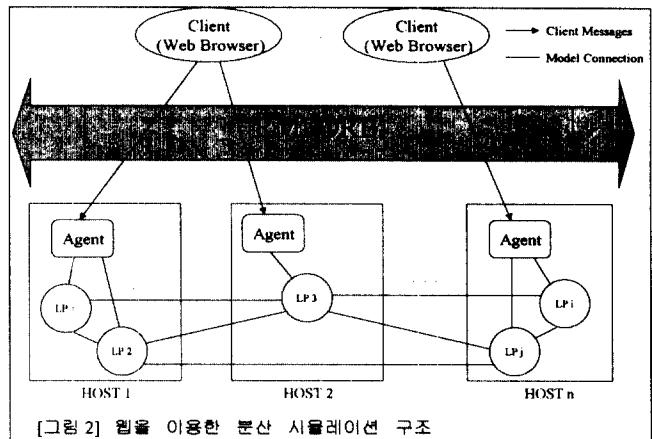
분석용이 아닌 대부분의 훈련용 전술 시물레이션은 사용자가 많으며, 지리적으로 분산되고, 규모가 크고, 시물레이션 수행을 위해 많은 인적, 물적 자원이 이

동하므로써 시물레이션 개발, 운영, 유지보수 단계의 전체 소프트웨어 생명단계(Life cycle)에 걸쳐 많은 비용을 필요로 한다.

본 연구에서는 이러한 단점을 극복하기 위하여 웹기술을 이용한 완전히 새로운 방법의 시물레이션 환경을 제시하고자 한다. [그림2]는 웹을 이용한 분산 시물레이션 환경 구조를 보여준다.

시물레이션 객체는 모두 자바로 작성되고, 분산환경은 이기종 동일환경 즉, 자바환경을 지원하는 RMI 분산 객체모델을 사용하였다. 클라이언트의 인터페이스는 웹

브라우저에서 제공한다. 브라우저는 자바의 애플릿을 가져온 컴퓨터에만 접근이 가능하고 다른 컴퓨터



에는 접근이 불가하므로 사용자는 자기와 관련되는 모델이 존재하는 컴퓨터에서 클라이언트 애플릿을 가져와서 시물레이션 서버에 메시지를 전달한다. 따라서 이러한 특성은 전술 시물레이션과 같은 대규모의 시물레이션에 사용자를 모델의 종류에 따라 쉽게 구분할 수 있게한다.

Agent는 클라이언트와 LP(Logical Process)사이에 위치하여 클라이언트로부터 메시지를 받아서 메시지가 처리되어야할 LP로 보내고, LP로 부터는 클라이언트에게 보낼 메시지와 시물레이션 시간을 유지하여 클라이언트에게 제공한다.

LP는 전체 시물레이션 모델을 부분모델로 분할시켜 동시에 실행하므로써 본질적으로 병렬성을 달성할 수 있으며 서로 연결된 LP는 보수적 알고리즘에 의해

실행된다.

6. 결론

자바와 웹 브라우저의 통합으로 인터넷을 기반으로 하는 어플리케이션 개발이 크게 증가하고있는 바, 시뮬레이션 분야도 웹을 기반으로 하여 새로운 방법론이 제기되고 있다. 본 연구에서는 웹의 객체지향언어로 이용되는 자바와 자바 분산 객체모델을 지원하는 RMI를 이용하여 클라이언트와 서버의 상호작용을 포함하는 분산 시뮬레이션 환경을 제시 하였다. 특히 전문 시뮬레이션 구축에 웹을 이용할 경우 시뮬레이션 소프트웨어 개발 및 운영, 유지보수의 비용을 줄일수 있으며, 시뮬레이션에 음성, 이미지, 비디오 등의 웹 문서를 이용할 경우, 새로운 시뮬레이션 개발환경이 될것으로 생각된다.

7. 참고문헌

[1] Paul A. Fishwick. 1996. "Web-based simulation : some personal observations", Proceedings of the 1996 winter simulation, pp.772-779, Dec., 1996

[2] Ernest H.Page., Robert L.Moose,Jr., Sean P.Griffin. 1997. "Web-based Simulation in SimJava using Remote Method Invocation", In: Proceedings of the 1997 Winter Simulation Conference, S.Andrad tir, K.J.Healy,D.H. Withers, and B.L. Nelson, eds., pp.468-474, Atlanta, GA, 7-10 December.

[3] Sun 96. "Sun Microsystems The Java Language Environment A White paper, 1996. (http://java.sun.com:80/doc/language_environment)

[4] Chandy, K.M.,and J.Misra. 1979. Distributed Simulation A Case Study in Design and Verification of Distributed Programs. IEEE Transactions on Software Engineering SE-5(5): pp.440-452, 1979

[5] Richard M. Fujimoto. 1990. "Optimistic approaches to parallel discrete event simulation",

Trans. of The Society for Computer Simulation, vol. 7, no.2, pp.153-191,1990.

[6] Alois Ferscha. 1995. "Parallel and Distributed Simulation of Discrete Event Systems", Handbooi of Parallel and Distributed Computing, McGraw, 1995.

[7] Jayadev Misra. 1986. "Distributed discrete-event simulation", ACM Computing Surveys, vol.18, no.1, pp. 39-65, Mar., 1986.

[8] Rajesh S. Nair, John A. Miller and Zhiwei Zhang. 1996. "Java-based Query Driven Simulation Environment", Proceeding of the 1996 winter simulation, pp. 786-793, 1996.

[9] Robert Orfali and Dan Harkey. 1997. "Client/Server Programming Java and CORBA", John Wiley & Sons, Inc.

[10] Sun 97, "RMI and JAVA Distributed Computing", (<http://java.sun.com/features/1997/nov/rmi.html>)