

웹기반 시뮬레이션을 위한 클라이언트 환경

사 공 봉 한 일 형 성 증 혁 남 영 환 서 현 곤 김 기 형

요 약

SimBox는 웹을 기반으로 한 원격 교육 시뮬레이션 도구로서 시뮬레이션 수행 절차에 익숙하지 않은 피교육자들에게 시각적 모델 개발 환경을 제공하여 쉽게 시뮬레이션 모델을 제작할 수 있도록 하였다. 모델 개발을 위해 클라이언트는 자바의 드래그 앤 드롭을 이용하여 시뮬레이션 모델들의 추가/삭제를 용이하게 하였고, 시뮬레이션 코드 작성을 서버가 대신함으로써 피교육자의 시뮬레이션 코드 작성에 대한 부담을 줄여 주었다. 또한 컴포넌트 기반의 모델을 개발해 재사용이 가능하며, 자바를 이용해 웹 환경에서 시뮬레이션의 진행상황을 애니메이션을 통해 효과적으로 시각화 할 수 있다.

A Client Environment for the Web-based Simulation

Bong Sagong Il-Hyung Han Jong-Hyuk Seong Yeunghwan Nam Hyungon Seo Kihyung Kim

ABSTRACT

SimBox is a client/sever-based web-based simulation tool designed for distance education. Simbox provides the students who are poor in the simulation process to develop easily simulation models with a visual model development environment. For developing a model, a modeler just uses simple drag-and-drop operations from model library toolbox in the client environment. After the development of models, the models are sent to a server which generates Java codes, compiles the codes into byte-codes, and return the codes to the client for the animation. By this partitioning of simulation job into client/server parts, this scheme can make a thin client environment, which can be easily installed into various client platforms. Also the use of the Java component technology, models can be reusable and simulation progress can be effectively visualized by the Java animation in the web.

1. 서론

교육용 시뮬레이션 개발 도구를 작성하는 경우에는 직관적이며 사용하기 쉬운 인터페이스를 제공해야 한다. 이는 교육자나 피교육자가 시뮬레이션이나 컴퓨터, 혹은 네트워크에 익숙하지 못 할

수 있기 때문이다. 또한 원격지에 있는 학습자를 위한 시뮬레이션 환경의 구축은 다양한 매체에 대한 연동의 어려움이 크다. 그러나 다양한 매체를 연동하는 것은 학습자에게 질의 처리, 통신과 협력 작업을 제공해 시뮬레이션의 이해와 사용에 큰 도움을 준다. 이러한 시뮬레이션 수행을 위한 시간을

절약하기 위해서 재사용성이 있는 모델의 구현을 제공해야 한다. 실제 학습자는 자신의 요구 사항에 부합하도록 모델의 구조를 검사하고 환경을 변화시킬 수 있다.

시물레이션 수행절차에서 요구되는 웹기반 시물레이션 환경은 크게 모델개발과 시물레이션실행 환경으로 구분될 수 있다. 먼저 모델 개발 환경을 보면 기존의 시물레이션 도구들은 다음과 같은 두 가지 형태의 개발 방법을 제공한다.

첫째, 개발자의 코딩에 의한 시물레이션 모델의 개발 환경이 있다. 이때 개발자는 자바 언어와 사용되는 시물레이션 라이브러리에 능숙해야 한다. 따라서 모델 개발에 소요되는 시간이 길어지며 이는 시물레이션 비용이 상승하는 결과를 초래한다. 대표적인 예로는 Simjava[1], JavaSim[2] 등이 있다.

둘째, 일반적인 드로잉 환경과 유사한 시각적 모델 개발 환경이 있다. 이는 시물레이션 모델을 구성하는 요소들을 미리 정의하고 개발자들이 이들을 적절하게 위치시킨 후 사건 정보 등의 처리 등을 조절하도록 한다. 이때 모델 구성요소의 정의와 재사용을 위해 컴포넌트화 한다. 개발자의 모델에 대한 기술이 완료되면 시물레이션 실행을 위한 코드가 생성되며 이는 개발자의 코딩에 대한 부담을 덜어준다. JSIM[3]이 대표적인 예이며, 미리 정의된 몇 가지 시물레이션 요소를 이용해 모델을 개발할 수 있는 환경을 제공한다.

SimBox는 웹을 기반으로 한 원격 교육 시물레이션 도구로서 클라이언트/서버 구조의 시물레이션 환경을 제공한다. 모델 개발과 시물레이션 실행은 클라이언트가 담당하며, 컴포넌트 관리와 코드 생성 그리고 컴파일 등의 기능을 서버가 담당하는 클라이언트/서버의 시물레이션 환경을 제공한다

[4].

SimBox 클라이언트는 GUI형태의 모델 개발 환경을 제공해 주기 때문에 사용자들이 쉽게 모델을 개발 할 수 있게 하고, 또한 컴포넌트 기반의 모델을 개발해 재사용이 가능하며, 자바를 이용해 웹 환경에서 시물레이션의 진행상황을 자바 애니메이션을 통해 효과적으로 시각화 할 수 있다.

2. 클라이언트/서버 기반의 웹기반 시물레이션

SimBox는 클라이언트/서버 구조의 시물레이션 환경으로서 클라이언트는 피교육자에게 시물레이션 하고자 하는 모델을 개발하기 위한 환경들을 제공해주고, 서버는 피교육자가 작성한 모델들에 대해 컴파일, 코드 생성과 피교육자가 개발한 모델에 대해 검증할 수 있도록 자바 애플릿을 통해 시각화 할 수 있는 환경을 제공해 준다[4].

클라이언트는 시물레이션에 익숙하지 못한 사람들에게 좀더 쉽게 시물레이션을 할 수 있도록 드래그 앤 드롭을 이용하여 만들고자 하는 컴포넌트를 쉽게 추가/삭제 할 수 있도록 하였고, 각 컴포넌트에 대한 시물레이션 파라메타도 직접 설정해 줄 수 있도록 하여서 시물레이션에 능숙하지 못한 사람들에게도 쉽게 시물레이션 모델을 개발할 수 있도록 하였다.

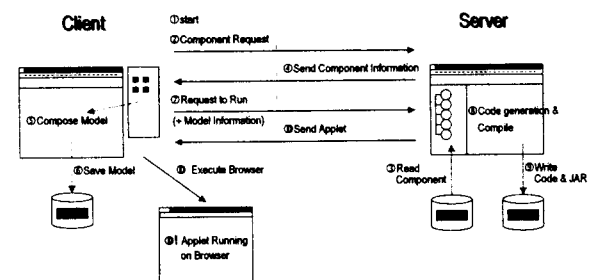


그림 1. SimBox의 전체 구조

(그림1)은 SimBox의 전체 구조를 나타낸다. SimBox를 이용한 시물레이션 수행과정은 다음과

같다.

- ① 데몬(daemon)형태의 서버를 동작시킨다.
- ② 클라이언트의 모델개발 환경을 동작시킨다.
- ③ 모델 개발에 필요한 컴포넌트 리스트를 서버에 요청한다.
- ④ 서버는 클라이언트가 요청한 컴포넌트 리스트를 전송한다.
- ⑤ 전송 받은 리스트 중 필요로 하는 컴포넌트를 지정한다.
- ⑥ 서버로부터 필요한 컴포넌트를 전송 받는다.
- ⑦ 000000 사용자의 모델을 개발한다.
- ⑧ 시물레이션 실행을 지시한다.
- ⑨ 서버로 사용자의 모델에 대한 정보를 전송한다.
- ⑩ 서버는 사용자의 모델정보를 이용해 시물레이션 코드를 생성하고 컴파일한다.
- ⑪ 서버에서 컴파일된 시물레이션 애플릿이 클라이언트에 전송된다.
- ⑫ 전송 받은 애플릿을 애플릿 브라우저를 통해 실행한다.

클라이언트는 개발하고자 하는 모델을 위해 필요한 컴포넌트를 서버로부터 전송 받아 사용하게 되는데, 이는 클라이언트의 전체적인 크기를 줄일 수 있게 도와준다. 따라서 클라이언트의 시물레이션 성능을 개선할 수 있다. 또한 공유되는 컴포넌트를 중복해서 관리하는 컴포넌트 관리 부담을 줄일 수 있다.

3. 클라이언트의 환경

3.1 클라이언트의 구조

SimBox 클라이언트는 피교육자의 모델 저작을 위한 모델 저작도와 피교육자가 작성한 모델을 서버에게 전달하고, 시물레이션을 실행하기 위한 커뮤니케이션 관리자로서 구성된다.

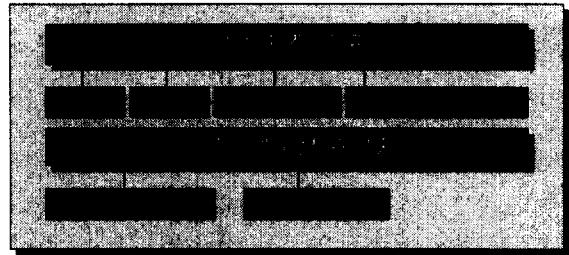


그림 2 SimBox 클라이언트의 구조

모델 저작 도구는 피교육자의 모델 저작을 돕기 위해 메뉴/툴바, 모델 개발을 위한 모델 박스, 모델을 작성하기 위한 모델 드로잉 패널, 원격지 모델을 액세스 하기 위한 원격지 모델 리스트 프레임으로 구성되며, 커뮤니케이션 관리자는 피교육자가 서버에게 요청하는 명령을 수행하는 역할을 한다.

3.2 클라이언트와 서버간의 통신

클라이언트와 서버간의 통신은 메시지 교환과 데이터 교환으로 구분된다. 메시지 교환은 클라이언트가 모델 작성을 하면서 필요한 데이터등을 요청할 때 사용되며, 클라이언트의 요청에 대해 서버는 필요한 데이터를 클라이언트로 전달한다. 이러한 메시지는 UDP 소켓과 객체 직렬화(Object Serialization)를 이용해서 전달하고, 데이터는 TCP/IP 소켓을 이용해서 전달된다. (그림3)은 클라이언트 서버간의 통신 구조를 나타낸다.

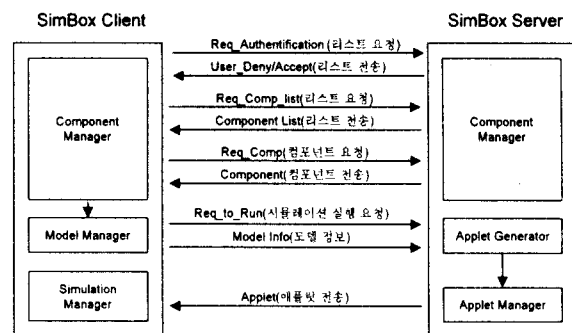


그림 3 클라이언트/서버간의 통신구조

3.3 전달 메시지 및 데이터 구조

클라이언트가 서버에게 요청하는 메시지는 사용자 인증, 컴포넌트 리스트 요청, 컴포넌트 요청, 시물레이션 실행 요청 4가지로 구성되어진다. 각 요청 메시지의 구조는 (그림4)와 같다.

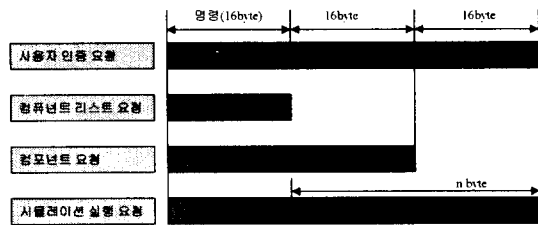


그림 4 메시지 구조

메시지의 처음 16byte는 명령 필드이고 각 명령에 따라 필요한 데이터들을 추가하여서 서버에게 전달된다. 시물레이션 실행 명령을 제외한 나머지 명령들은 자바의 String type의 데이터들이 추가된다. 시물레이션 실행 명령의 Model Information table은 피교육자가 저작한 모델들에 대한 모든 정보를 전달하기 위해서 Vector type으로 되어 있다. Vector는 배열과 같은 자료 구조의 기능을 제공하면서 동적으로 자신의 크기를 변경시킬 수 있고, Object들을 저장할 수 있도록 설계되어 있기 때문에 피교육자가 작성한 모든 정보들을 저장할 수 있다.

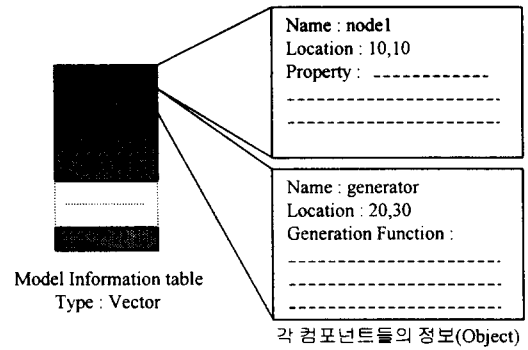


그림 5 Model Information Table

(그림5)는 Model Information table에 피교육자가 저작한 모델에 대한 정보들이 저장된 모습을 보여 준다. 각각의 컴포넌트들의 정보는 Object들로 구성되어 있으며 컴포넌트의 이름, 위치, 기타 데이터들이 들어 있다. 이러한 데이터들은 서버가 코드 생성과 컴파일, 애플릿 생성에 필요한 자료가 된다.

3.4 서버와 클라이언트간의 통신

서버와 클라이언트가 통신을 위해서 각각 쓰레드를 두어서 데이터와 메시지 전달을 용이하게 하였다. 클라이언트가 서버에게 메시지 전달을 위해 자바의 객체 직렬화(Object Serialization)를 사용하였다. 객체 직렬화란 자바가 제공하는 객체 저장기능이다. (그림6)에서 Vector Type의 모델 정보 Object는 클래스 ObjectOutputStream의 write() 메소드에 의해 byte stream으로 변환된 후, OutputStream을 통해 서버로 전달되면 서버는 InputStream을 통해 받은 byte steam을 다시 Object(Vector type)으로 변환해서 클라이언트가 보낸 객체를 서버에서 다시 원래의 객체로 활성화하여 코드 생성에 필요한 정보를 제공한다. 또한 클라이언트의 컴포넌트 리스트 요청에 대해서도

이러한 객체 직렬화 방법을 사용하여 서버로부터 컴포넌트 리스트를 다운로드 받는다.

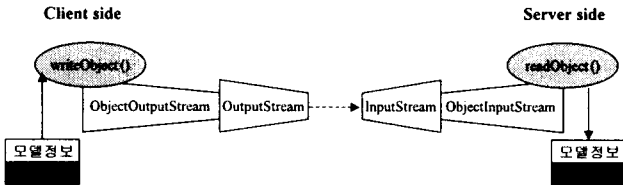


그림 6 객체 직렬화(Object Serialization)

3.5 시물레이션 엔진(Simjava 1.2)

클라이언트에 의해 전달된 모델 정보들은 서버에서 시물레이션이 가능한 Simjava 코드로 생성된다. Simjava는 프로세스 기반의 discrete event simulation 패키지로서 Simjava가 제공하는 시물레이션 기능에는 메시지 교환을 위한 이벤트의 전달과 시물레이션 시간 관리 기능, 일정한 분포에 따른 임의값(random value) 생성 기능이 있다. 다음은 Simjava가 지원하는 시물레이션 지원 기능들이다.

- sim_schedule() : 포트를 통해 다른 요소에게 사건을 전달함
- sim_hold() : 시물레이션 시간을 잠시 멈춤
- sim_wait() : 사건 객체가 도달할 때까지 대기
- sim_select() : 큐로부터 사건을 선택함
- sim_trace() : 시간정보가 포함된 메시지를 trace file에 기록

4. 구현

SimBox 클라이언트는 프로그램이 처음 로딩되면서 모델에 관한 리스트를 요청한다. 그러면 서버로부터 모델 개발을 위한 컴포넌트들에 관한 정보들을 네트워크를 통해서 전송한다. 이렇게 다운로드 된 정보들은 SimBox 클라이언트에서 트리 형태의 리스트

로 사용자에게 보여진다. 사용자는 이러한 모델을 이용해서 모델을 개발해 클라이언트에 저장한 후 서버에게 "Request to Run"을 요청하면 서버는 클라이언트의 웹브라우저에서 사용할 수 있는 컴파일된 애플릿 클래스와 html 파일을 클라이언트로 전송해 준다. 그러면 클라이언트는 서버로부터 다운로드한 html 파일을 웹브라우저나 자바 컴파일러에 포함되어 있는 Appletviewer를 이용해서 사용자가 개발한 모델들의 시물레이션을 보여준다.

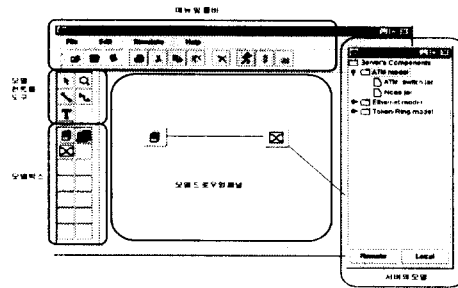


그림 7 SimBox 클라이언트의 GUI

SimBox클라이언트의 기본적인 GUI는 툴바와 모델을 위한 툴박스, 컴포넌트 개발을 위한 컴포넌트 드로잉 패널(drawing panel)과 모델을 컨트롤하기 위한 모델 컨트롤 도구, 서버로부터 다운로드 받은 컴포넌트들의 목록을 트리 형식으로 보여주는 프레임과 메뉴로 구성되어 있다.

4.1. 모델 컨트롤 도구

모델 컨트롤 도구는 모델의 기본적인 선택 기능, 이동과 확대 기능이 있으며, 컴포넌트 드로잉 패널에 피교육자가 선택한 컴포넌트간의 연결을 설정해주는 기능 등을 포함하고 있으며 또한 서버로부터 전송된 추가적인 컴포넌트를 등록하고 이를 이용한 모델 개발을 지원할 수 있는 기능을 한다.

4.2 모델 박스

모델 박스는 SimBox 클라이언트가 가지고 있는 기본적인 컴포넌트들과 서버로부터 다운로드된 컴포넌트들이 추가될 부분이다. 피교육자는 모델 박스의 컴포넌트를 이용하여 자신이 개발하고자 하는 모델을 위해 컴포넌트 드로잉 패널에 추가하고 난 후 모델 컨트롤 도구를 이용해서 모델간의 연결을 설정해 주기만 하면 된다. 또한 각 컴포넌트를 클릭하면 각각의 속성을 수정할 수 있는 다이얼로그 박스가 화면에 나타나게 된다. 사용자는 다이얼로그 박스를 이용해 자신이 개발하고자 모델의 구성요소에 대한 속성을 지정하며, 이는 시물레이션에 필요한 정보를 제공하게 된다.

4.3 모델 드로잉 패널

드로잉 패널의 기능은 사용자가 모델을 개발할 수 있는 시각적인 환경을 제공해 주는 것이다. 피교육자는 모델 박스의 컴포넌트와 모델 컨트롤 도구를 이용해서 모델을 디자인하고 컴포넌트간의 연결과 각 컴포넌트들의 Property를 설정할 수 있는 환경을 제공해 준다. 컴포넌트간의 연결은 두 컴포넌트를 클릭해서 지정할 수 있으며, 이들 모델을 구성하는 컴포넌트와 연결에 관한 정보는 서버에 전송되어 시물레이션 코드를 생성할 때 참조되는 정보이다.

4.4. 모델 리스트 프레임

모델 리스트 프레임은 실제 서버가 가지고 있는 컴포넌트들의 리스트만을 나열한 것이다. 사용자가 어떤 모델을 개발하기 위해서 필요한 컴포넌트들은 대부분 서버가 가지고 있기 때문에 서버의 모델을 선택적으로 가져오기 위한 프레임이 필요하다. 결국 자신이 필요로 하는 컴포넌트를 선택하면 선택된 컴포넌트들은 서버로부터 다운로드 되어서 모델 박스에 추가되며, 이후부터 사용자는 자신이 개발하고자

하는 모델을 구성하기 위해 이들 컴포넌트들을 사용할 수 있다.

4.5 컴포넌트의 시물레이션 파라메타 설정

피교육자가 작성한 모델들은 각 모델에 대해 대한 시물레이션 파라메타들을 쉽게 설정하기 위해 각 컴포넌트마다 파라메타 설정 다이얼로그를 추가 하였다. (그림8)은 파라메타 설정의 예이다.

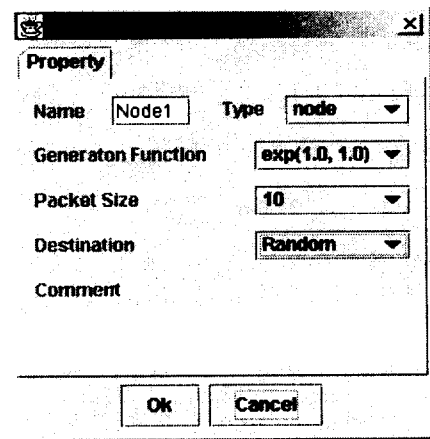


그림 8 파라메타 설정의 예

5. 결론

본 논문은 시물레이션에 익숙하지 못한 사람들에게 시각적이고 직관적인 인터페이스를 이용한 모델 개발 환경을 제공하며, 자바의 특징인 드래그 앤 드롭을 이용하여 쉽게 컴포넌트를 추가/삭제하고, 또한 코드 생성에 대한 부담을 서버가 대신 함으로 해서 사용자에게 좀더 쉽게 시물레이션을 수행할 수 있도록 하였다.

[참고문헌]

- [1] Howell, F.W, The simjava home page, <http://www.dcs.ed.ac.uk/home/hase/simjava>, April, 1999
- [2] University of Newcastle upon Tyne, JavaSim

home pge, <http://javasim.ncl.ac.uk/>, April, 1999

- [3] John A. Miller, Andrew F. Seila and Xuewei Xiang, "The JSIM Web-Based Simulation Environment," 1999
- [4] 사공봉, 원격교육을 위한 웹기반 시뮬레이션 환경, 멀티미디어 학회, 1999
- [5] Sun. Microsystems, The Java Tutorial, <http://java.sun.com/docs/books/tutorial/index.html>, April. 1999.
- [6] Merlin Hughes, Java Network Programming 2nd edition, MANNING, 1999.