

# 실행시간 바인딩을 이용한 동적 웹 기반 시뮬레이션 방법론

정문규, 한일형, 백희정, 서현곤, 김기형  
영남대학교 컴퓨터공학과  
E-mail : hgseo@cse.yeungnam.ac.kr

## Dynamic Web Based Simulation Methodology Using Execution time Binding

MoonGyu Jung, IlHyung Han, HeeJung Beak, HyunGon Seo and KiHyung Kim  
Network Computing Laboratory  
Dept. of Computer Engineering, YeungNam University

### 요 약

웹기반 시뮬레이션이란 인터넷을 기반으로 플랫폼에 관계없이 시뮬레이션을 실행할 수 있는 것으로 기존에 Jsim, SimJava, JavaSim, DEVS-Java등과 같은 다양한 도구들이 개발되었다. SimJava는 자바 프로그래밍 언어를 이용하여 시뮬레이션 라이브러리를 개발할 수 있는 이산 사건 시뮬레이션 패키지로 Simjava 패키지, SimAnim 패키지, SimDial 패키지로 구성되어 있다. SimJava는 문자기반의 시뮬레이션뿐만 아니라, 시뮬레이션 실행과정과 결과를 시각적으로 볼 수 있는 기능을 제공하기 때문에, 웹 기반 시뮬레이션에서 많이 사용되고 있다. 본 논문에서는 SimJava를 이용하여 웹에서 모델을 설계하고 시뮬레이션 및 애니메이션이 가능한 SimDraw를 제안한다. 특히 SimDraw는 동적 바인딩을 이용함으로써 모델을 설계하고 번역할 필요 없이 바로 웹에서 실행 된다.

### 1. 서론

기존의 시뮬레이션 환경은 시뮬레이션 언어 기반의 접근과 라이브러리 기반 접근에 기초를 두고 있다. 따라서 특정한 시뮬레이션 언어 또는 시뮬레이션이 실행되는 플랫폼에 종속 될 수밖에 없다. 즉, 시뮬레이션을 위해 개발된 모델은 다른 시뮬레이션 환경에 사용할 수 없는 상호작용이 결여되어 있고 또한 특정 시뮬레이션 언어로 구축되었기

때문에 다른 시뮬레이션 들, 다른 운영 체제에서 사용할 수 없는 호환성문제가 발생하게 된다. 뿐만 아니라 인터넷을 이용한 웹 기반 시뮬레이션을 지원하지 못하는 단점을 가지고 있다. 즉, 기존의 시뮬레이션 라이브러리는 기계종속(machine dependent)인 실행 코드를 생성하기 때문에 다양한 컴퓨터 기종과 플랫폼으로 구성된 인터넷에서 실행하는데는 문제점이 있다. 인터넷에서 실행되기

위한 웹 기반 시뮬레이션은 JAVA 등을 이용함으로써 문제를 해결할 수 있기 때문에 자바를 이용한 다양한 시뮬레이션 라이브러리가 소개되고 있다.

SimJava는 자바를 이용한 대표적인 시뮬레이션 라이브러리이다. Sim\_entity 객체에 의해 생성된 각 엔티티(entity)는 멀티 쓰레드로 동작되며, 이 객체들에 의하여 시뮬레이션이 이루어진다. 또한 각 엔티티 사이에는 Sim\_port 객체에 의해 생성된 포트를 통해 이벤트(event)가 전달되어 정보를 교환하게 된다. 현재 SimJava를 이용하여 구현된 라이브러리들이 많이 있지만 이것들은 개발 프로그램에서만 사용될 수 있고, 또 프로그래머에 의해 결정된 상황에서만 시뮬레이션이 가능하기 때문에 동적인 웹 기반 시뮬레이션을 제공하지 못하고 있다.

## 2. 시뮬레이션의 분류

시뮬레이션 환경은 다음 여러 경우에 따라 분류할 수 있다. 첫 번째는 시뮬레이션 모델의 디자인 방법에 따른 분류, 두 번째는 정적인 것과 동적인 실행에 따른 분류, 마지막으로 시뮬레이션 모델의 파라메타 값의 바인딩에 따른 분류로 나눌 수 있다.

첫째 시뮬레이션 모델의 디자인 방법에 따른 분류로 프로그램 코딩 방법과 GUI 방법으로 구분할 수 있다. 프로그램 코딩에 의한 시뮬레이션의 디자인의 경우 개발자는 개발언어에 대한 상당한 수준을 요구하게 되며 또한 소요시간을 많이 요구할 뿐만 아니라 개발된 모델에 대한 유지보수 비용도 증가하는 문제점을 가지게 된다. 대표적인 예로 SimJava, JavaSim, DEVS-JAVA 등이 있다. 다음으로 GUI 방법을 이용한 시각적인 모델 개발이다. 이는 시뮬레이션을 구성하는 요소들이 미리 개발되어 있어 사용자가 이들을 이용하여 시뮬레이션을 디자인 하는 것으로 이용자는 시뮬레이션 프로그램에 대한 지식 없이 목적하는 시뮬레이션을 실행할 수 있는 장점을 가지고 있다. 하지만 이 경우

미리 설정된 시뮬레이션만 수행하기 때문에 기능이 제약적이다. 또한 플랫폼에 종속되기 때문에 웹 기반 시뮬레이션을 지원하지 못하는 단점을 가지고 있다. 대표적인 것으로 JSIM, COMNET III 등이 있다. 특히, COMNET III는 컴퓨터 네트워크 시뮬레이션을 위한 제품으로 개발자가 다양한 네트워크 상황을 설정하여 시뮬레이션 할 수 있는 기능을 제공한다.

두 번째로 웹 기반 시뮬레이션은 시뮬레이션이 실행되는 상태와 상황에 따라 정적인 웹 기반 시뮬레이션과 동적인 웹 기반 시뮬레이션으로 구분할 수 있다. 정적인 웹 기반 시뮬레이션은 개발자에 의하여 시뮬레이션 모델과 시나리오가 프로그램 되어 실행된다. 또한 시뮬레이션 할 수 있는 상황/상태 역시 프로그램 개발자에 의하여 결정되어 개발자가 지원하는 기능에 대해서만 웹 기반 시뮬레이션을 할 수 있다. 동적인 웹 기반 시뮬레이션은 시뮬레이션 모델은 개발자에 의하여 개발되지만 시나리오와 상황/상태 등은 시뮬레이션을 실행하는 사용자에게 의하여 결정된다. 즉 정적인 웹 기반 시뮬레이션은 시뮬레이션에 필요한 모든 정보가 시뮬레이션 프로그램이 실행되기 전에 결정이 되며, 동적인 웹 기반 시뮬레이션은 시뮬레이션 프로그램과 관계없이 사용자의 요구와 필요에 따라 시뮬레이션 실행할 수 있는 것을 의미한다.

마지막으로 모델의 파라메타 값의 바인딩에 따라 컴파일 시간에 바인딩 되는 것과 실행시간에 바인딩 되는 것이 있다. 컴파일 시간에 값이 결정이 되는 것은 정적인 시뮬레이션이라 할 수 있고, 실행 시간에 결정이 되는 것을 동적이라 할 수 있다. 이것도 역시 프로그래머에 의해 값이 결정되는가 아니면 시뮬레이션 사용자에게 의해 결정되는가로 구분되어 진다.

본 논문에서는 GUI를 지원하고 동적인 웹 기반 시뮬레이션을 지원하며, 실행시간에 시뮬레이션 모델의 파라메타가 결정되는 동적인 웹 기반 시뮬레이션 방법론인 SimDraw을 제안한다.

### 3. SimDraw (웹 기반 시물레이션 환경)

#### 3.1 SimDraw 구조

SimDraw Ver2.0은 기존의 SimDraw와 같이 서버/클라이언트 구조로 이루어져 있다. 전체 시스템의 구성은 그림 1과 같다. 서버에는 시물레이션을 수행하기 위해 필요한 모델 컴포넌트, 엔티티 데이터베이스와 사용자 인증 데이터베이스 등으로 구성되어 있다. SimDraw서버 대문은 클라이언트에서 접속을 대기하다가, 접속이 요청되면 사용자 인증을 처리한 후 애플릿 관리자가 클라이언트로 SimDarw 클라이언트를 다운로드 시켜준다. 서버의 엔티티 관리자는 현재 시물레이션 가능한 모델 엔티티를 관리하는 것으로서, 클라이언트에서 필요한 모델을 요청하면 해당 모델 엔티티를 데이터베이스에서 읽어 Simdraw클라이언트로 전송해주는 역할을 수행한다.

SimDraw 클라이언트는 자바 애플릿으로 시물레이션을 하기 위해 필요한 시물레이션 엔진, Model 작성기, 뷰어, 엔티티 로더 등으로 구성되어 있다.

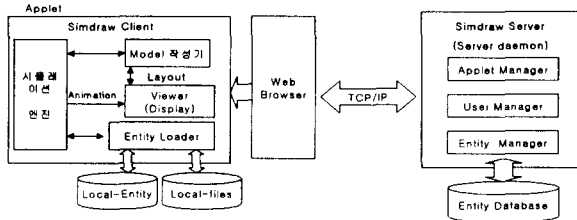


그림 18 SimDraw Ver2.0 구조

시물레이션 엔진은 시물레이션 제어를 담당하는 것으로서 시물레이션 엔티티들의 관리와 드로잉된 모델의 시물레이션 및 애니메이션에 필요한 모든 처리를 담당하는 부분이다. 또한 모델작성기에서 사용자가 드로잉한 정보를 이용하여 시물레이션 실행시 필요한 객체 쓰레드를 생성하여 뷰어로 애니메이션을 출력하는 역할을 담당한다.

모델 작성기는 Swing으로 만든 GUI로서 사용자들이 설계할 시물레이션 모델을 드로잉하는 영역이다. 사용자는 모델작성기의 툴바에서 필요한 객

체를 선택하여 드로잉 패널에 추가하고 Copy, Paste, Move, Select 등과 같은 명령과 Ctrl, Shift, Alt 키 등을 이용하여 각 객체의 상태와 파라미터를 설정함으로써 시물레이션 모델을 드로잉 한다.

엔티티 로더는 시물레이션 모델 실세계에 필요한 모델 엔티티를 선택하거나, 기존에 작업한 모델이 파일로 로컬에 저장되어 있는 경우, 로컬파일의 모델을 로딩하는 일을 담당한다. 모델 엔티티는 로컬 엔티티 데이터베이스와 SimDraw 서버측의 엔티티 데이터베이스에 있는 것을 사용하는데, 필요에 따라 선택하여 사용할 수 있다. 또한 자신의 로컬 시스템에 작업한 모델을 저장하고 불러올 수 있는 기능을 담당하기 때문에 필요에 따라 전에 제작했던 모델을 다시 불러와서 시물레이션 할 수 있다.

뷰어는 모델작성기에 포함된 드로잉 패널이라 할 수 있다. 사용자는 뷰어를 통해 시물레이션 되는 과정을 볼 수 있다. 즉 사용자의 입장에서 볼 때는 모델 작성기의 패널이 결국 뷰어가 된다고 할 수 있다.

#### 3.2 SimDraw Ver2.0의 특징

SimDraw Ver2.0은 기존의 SimDraw의 기능을 모두 수용하면서 새롭게 추가된 몇 가지 기능과 특징을 가지고 있다.

첫째, SimDraw클라이언트는 자바 애플릿으로 애플릿 자체에 시물레이션 엔진, 모델 작성기, 뷰어 등을 포함하고 있다. 애플릿 자체에 시물레이션 엔진을 가지고 있다는 것은 SimDraw클라이언트가 서버의 도움 없이 자체적인 시물레이션이 가능하다는 것을 의미한다. 즉, 기존의 SimDraw는 시물레이션 엔진이 서버에 존재했기 때문에 사용자가 설계한 모델의 시물레이션 실행은, 먼저 시물레이션 코드를 생성하여 서버가 번역하고 클라이언트로 번역된 바이트 코드가 전송되어 실행되었다. 이러한 방법은 많은 시간을 요구하게 되고, 번역도중 오류, 통신 잡음 등에 의하여 잘못된 결과를 생성

할 수 있는 단점을 가지고 있었다.

두 번째로 기존 방식에서는 드로잉된 모델은 항상 서버에서 번역되었기 때문에 서버에 과부하가 발생하며 네트워크의 통신비용을 증가시키는 문제점을 가지고 있었다. 그러나 SimDraw Ver2.0에서는 시물레이션 엔진을 클라이언트에 내장되어 있기 때문에, 사용자가 드로잉한 모델을 번역 없이 바로 실행할 수 있어 아주 효과적으로 시물레이션을 수행할 수 있다. 즉, 처음 사용자가 웹 브라우저를 이용하여 SimDraw 서버에 접속하면 SimDraw 클라이언트가 다운로드 되고, 시물레이션에 필요한 모델 엔티티를 클라이언트가 요청하면 서버가 모델 엔티티를 다운로드 시켜줌으로써, 더 이상 서버/클라이언트간의 통신 없이 시물레이션을 실행할 수 있다.

세 번째로 모델을 드로잉하는 드로잉 패널과 뷰어가 동일한 애플릿이다. 기존 방법은 사용자가 모델을 드로잉하는 패널과 뷰어가 각각 다른 애플릿으로 동작되기 때문에 혼동을 주었지만, SimDraw Ver2.0에서는 드로잉 패널과 뷰어를 같은 애플릿 내어 둬으로써 보다 효과적이다.

네 번째로 기존방법에서는 드로잉된 모델의 객체의 위치가 변경되거나 새로운 객체의 추가 및 삭제할 경우, 다시 자바 코드를 생성하여 서버에서 번역해야 했지만 SimDraw는 파라메타 값만 변경해 줌으로써 문제를 해결할 수 있다.

마지막으로 시물레이션의 각 엔티티는 시물레이션 로직과 객체의 속성 자료(argument)로 구성되어 있다. 시물레이션 로직은 실제적인 자바 클래스이고, 속성 값은 자바 클래스에서 필요한 값(argument)인데 기존 방법은 이 두 가지를 컴파일할 때 같이 번역하여 시물레이션 객체를 생성하지만, SimDraw Ver2.0에서는 시물레이션 로직을 포함하고 있는 자바 컴포넌트가 이미 독립적인 모듈로써 컴파일 되어 Simdraw 클라이언트에 내장되어 있고, 시물레이션 로직에서 필요한 값(argument)은 실행시간에 바인딩해 줌으로써, 실행

시간에 객체 쓰레드를 생성한다. 엔티티(Entity)는 독립적으로 존재하는 실세계의 사물, 객체를 나타내는 것으로 여기서는 Simjava시물레이션 로직을 가진 객체를 말한다. 본 논문에서는 시물레이션에 사용되는 모든 엔티티 클래스를 그림 2와 같이 두 개의 독립된 자바 클래스를 이용하여 생성한다.

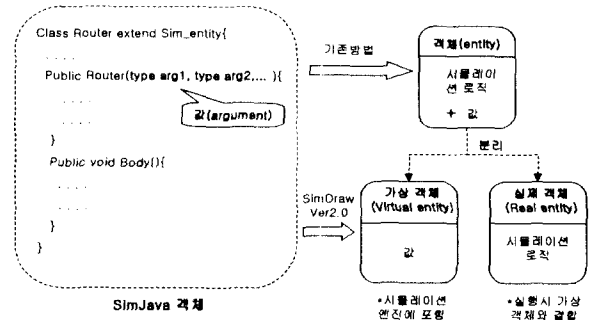


그림 19 SimDraw객체의 구성

하나는 가상 엔티티 클래스(Virtual Entity Class)이고 다른 하나는 실제 엔티티 클래스(Real Entity Class)이다. 실제 엔티티 클래스는 시물레이션 로직을 가지고 있는 것으로, SimJava의 Sim\_entity 클래스를 상속받아 생성된 클래스이다. 가상 엔티티 클래스는 실제 엔티티의 속성을 부여하기 위한 속성 값을 가지는 엔티티이다. 모델 작성기에서 사용자의 모델 드로잉으로 생겨난 객체의 좌표위치, 이름, 포트 등과 같은 값이 속성 값에 속한다.

### 3.3 쓰레드 생성

쓰레드 생성은 사용자가 모델 드로잉을 마치고 실행버튼을 클릭할 때 생성된다. 드로잉 과정에서 선택된 모델은 좌표, 이름, 상태 정보를 저장하는 노드를 생성하여 엔티티 리스트에 추가한다. 또한 각 객체 모델을 연결하는 링크 정보도 링크리스트에 저장된다. 그림 3에서 Router와 Node는 시물레이션 로직을 가지고 있는, 즉 컴파일 된 자바 코드이다. 또한 엔티티 리스트와 링크 리스트는 드로잉하는 과정에 드로잉 패널에 위치하는 객체의 정보(위치, 이름, 고유설정 값 등)와 객체를 연결하는

링크 정보를 저장하고 있는 리스트이다.

이선 수행 절차를 나타낸 것이다.

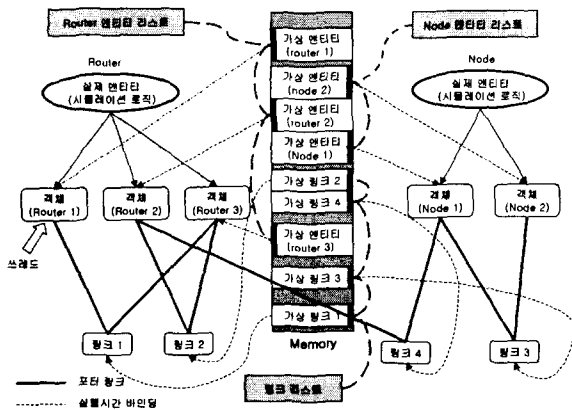


그림 20 실행시간 바인딩을 이용한 쓰레드의 생성

각 엔티티 리스트와 링크 리스트에 저장된 값은 실행시간에 실제 엔티티와 결합되어 시뮬레이션 쓰레드를 생성한다. 엔티티 리스트와 링크 리스트는 로컬 메모리에 저장되어 있다. 실행시간에 시뮬레이션 객체 쓰레드가 생성되기 때문에 SimDraw클라이언트는 서버의 도움 없이 시뮬레이션이 가능하다.

### 3.4 시뮬레이션 절차

Simdraw 클라이언트에 애플릿이 로드된 후 사용자는 메뉴에서 필요한 모델을 선택하면, 선택된 시뮬레이션 모델 엔티티가 로드된다. 그러면 그림4와 같이 화면 왼쪽에는 객체, 중앙에는 드로잉 패널이 위치한다. 툴바의 원하는 객체 아이콘을 선택하여 드로잉 패널에 배치하여 시뮬레이션 모델을 드로잉 한다. 드로잉 패널에 위치한 모든 객체 아이콘은 생성될 때 좌표와 속성 값을 엔티티 리스트에 저장한다. 각 객체 아이콘의 속성 값 설정은 대화상자를 이용하여 설정할 수 있다. 대화 상자에서는 객체의 이름, 다른 객체와의 링크 정보, 그 객체가 가지는 고유 속성 값 등을 지정할 수 있다. 또한 이벤트 신호를 전달하기 위한 링크 역시 링크 리스트를 생성하여 링크 정보를 저장한다. 그림 5는 SimDraw클라이언트 사이드에서 시뮬레

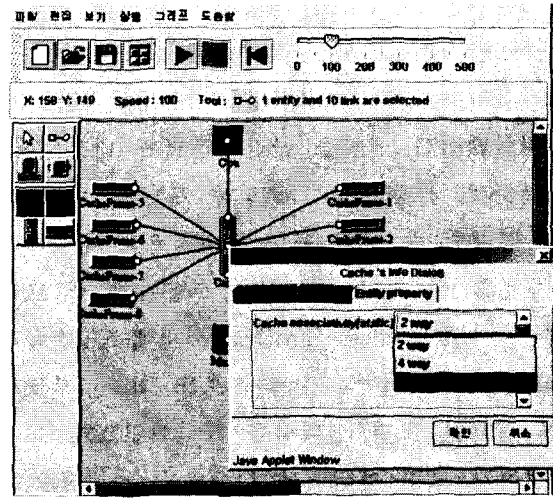


그림 21 Simdraw 드로잉 예제

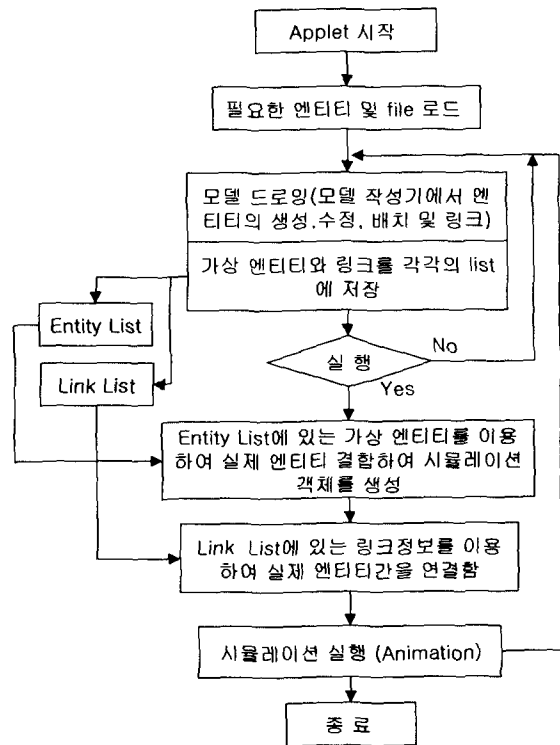


그림 22 클라이언트에서의 시뮬레이션 절차

## 4. 결론

본 논문에서는 프로세스기반의 이산 사건 시뮬레이션인 HASE++의 자바용 버전 Simjava를 이용하여 원격지에서 동적인 웹 기반 시뮬레이션이 가능한 방법론을 제시하였다. 제시된 방법은 드로잉

된 모델을 서버에서 번역하여 번역된 바이트 코드를 클라이언트로 다운 로드하여 실행하던 종전 방법과 달리 시뮬레이션 모델이 드로잉된 클라이언트 사이드에서 바로 시뮬레이션을 실행할 수 있는 획기적인 방법으로 통신비용의 감소, 서버 부하 감소, 실행시간 감소 등과 같은 특징을 가지고 있다. 또한 제시된 방법론을 이용하여 Simdraw Ver2.0을 구현하였다. 앞으로의 과제로는 Simdraw에서 제공하는 보다 다양한 시뮬레이션 모델을 개발하는 것이다.

#### 참고문헌

- [1] Ernest H. Page, Arnold buss, Paul A. Fishwick, Kevin J. Healy, Richard E. Nance, Ray J. Paul, "Web-Based Simulation: Revolution or Evolution?", ACM Transaction on Modeling and Computer Simulation, Vol. 10, No. 1, page 3-17 January 2000
- [2] Ernest H. page, Robert L. Moose, Jr. Sean P. Griffin, "Web-based Simulation in Simjava Using Remote Method Invocation", Proc. of the 1997 Winter Simulation Conference, 1998, page 468-473
- [3] Howell, F.W, The simjava homepage, <http://www.dcs.ed.ac.uk/home/hase/simjava>, April 1999
- [4] Healy, K.J and R.A. Kilgore, "Introduction to Silk and Java-based Simulation", Proc. of the 1998 Winter Simulation Conference, 1998
- [5] Austin, W.j., J. Liddle, R.C. Thomas, P. McAndrew, Networked Wductional Simulations in Java, <http://www.ltc.hw.ac.uk/mverse>, Oct. 1998
- [6] University of Newcastle upon Tyne, JavaSim home page, <http://javasim.ncl.ac.uk>, April 1999
- [7] Zeigler, B.P., "DEVS-JAVA modeling and simulation environment" in <http://www-ais.ece.arizona.edu/software/software.html>, 1997
- [8] Yeonghwan Nam, Kihyung Kim, "VMAC: A GUI-based Web-based Simulation Environment", IEEE TENCON, pp.1498-1501, 1999
- [9] Hyungon Seo, SaGong Bong, Kihyung Kim, "Web-based Modeling, Simulation and Animation of Routing Protocols", Proc. of the IASTED international Conference, November 2000, page 356-360