

# 웹-기반 분산 시뮬레이션 환경을 위한 분산 시뮬레이션 플랫폼

설도환, 조상영

한국의국어대학교 컴퓨터공학과

## A Distributed Simulation Platform for Web-based Distributed Simulation Environment

Do-Hwan Seol, Sang-Young Cho

Dept. of Computer Science Engineering, Hankuk Univ. of Foreign Studies

### 요 약

최근 웹의 폭발적 확산과 웹 기술의 급속한 발달에 따라 웹 환경과 이산사건 시스템 시뮬레이션 환경을 결합 시키려는 웹-기반 시뮬레이션에 대한 연구가 활발히 진행되고 있다. 웹 문서 내에서 시뮬레이션 모델을 수행시키는 웹-기반 시뮬레이션은 교육, 광고, 제품 소개 및 훈련, 동적 문서의 제작 등 다양한 응용 분야가 있다. 기존의 웹-기반 시뮬레이션 환경은 전체 시뮬레이션이 한 시스템에서 수행되는 순차적 시뮬레이션에 기반을 두고 있다. 본 논문에서는 분산된 환경 하에서 동시 공학적인 시스템 개발에 이용될 수 있는 분산-모델 분산-시뮬레이션 형태의 웹-기반 시뮬레이션을 제시하며 이의 구현을 위한 전 단계로서 인터넷 하에서 구축된 분산 시뮬레이션 환경에 대해 설명한다. 본 환경에서는 네트워킹 시스템 및 하드웨어 모델링을 용이하게 하기 위하여 확장된 DEVS 형식론을 도입하였고 개발된 모델이 이종의 시스템들 간에 자유로이 전송되어 수행될 수 있도록 Java를 사용하여 시뮬레이션 환경을 구축하였다.

### 1. 개요

인터넷을 기반을 둔 웹 기술의 발달에 따라 웹 문서는 단순한 텍스트에서 화려한 멀티미디어 문서로 바뀌고 있다. 또한 플랫폼에 독립적인 언어인 Java의 등장에 따라 웹 문서는 애플릿을 이용하여 상호작용이 가능한 동적 문서로 확장되었다. 이러한 웹 기술의 발전은 단순한 정보의 획득뿐만 아니라 기존의 사람과 사람 사이의 정보소통 체계를 혁신하고 있다.

현재 이러한 웹 환경과 이산사건 시스템 시뮬레이션의 결합을 시도하는 웹-기반 시뮬레이션 (Web-based Simulation) 연구가 진행되고 있고 교육, 광고, 제품 소개 및 훈련, 동적 문서의 제작 등 다양한 응용분야를 창출하고 있다[1].

본 논문에서는 지역적으로 분산된 환경 하에서 동시 공학적인 시스템 개발에 이용될 수 있는 분산-모델 분산-시뮬레이션 형태의 웹-기반 시뮬레이션을 제시하며 이의 구현을 위한 전 단계로서 인터넷 상에서 구축된 분산 시뮬레이션 환경에 대해 설명한다. 2장에서는 기존의 웹-기반 시뮬레이션 형태와 구축된 분산-모델 분산-시뮬레이션 환경에서 채용한 DEVS에 대해 기술한다. 3장에서는 확장된 DEVS에 대해 설명하고 이를 적용하여 구축된 시스템에 대해 설명하고 4장에서 결론은 맺는다.

### 2. 관련 연구

기존의 웹-기반 시뮬레이션 환경은 다음 세 가지 형태로 분류될 수 있다[2].

**애플릿-수행형태:** 시뮬레이션 모델을 애플릿으로 만들어 클라이언트의 웹 브라우저에서 이를 내려 받아 시뮬레이션이 수행되는 형태이다. 기존의 C++로 구축된 이산사건 시뮬레이션 환경을 Java로 재 구축한 형태가 대부분이다.

**서버-수행형태:** 시뮬레이션 모델은 서버에서 수행되고 클라이언트에서는 시뮬레이션 모델에 대한 인터페이스만 내려 받아 이를 통해 클라이언트에서 시뮬레이션의 조직이나 시뮬레이션 수행 결과를 보는 형태로 CGI (Common Gate Interface) 또는 Java의 RMI

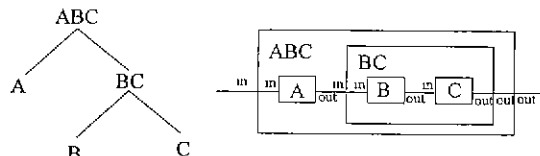
(Remote Method Invocation)를 이용하여 구현한다.

**분산-모델 클라이언트-수행 형태:** 시뮬레이션 모델이 웹 상의 여러 시스템이 분산되어 있고, 이 모델들을 임의의 클라이언트에서 내려 받아 전체 시뮬레이션 모델을 만들어 시뮬레이션을 수행하는 형태이다[4].

위의 세 가지 형태의 웹-기반 시뮬레이션은 전체 시뮬레이션 모델이 하나의 시스템에서만 수행되는 순차적 시뮬레이션을 수행하기 때문에 기존의 시뮬레이션 기술과 웹 기술로 쉽게 구현이 된다. 그러나 지역적으로 분산되어 있는 상황에서의 동시 공학적 시스템 개발과 같은 응용 분야에 적용하기가 어렵다.

본 논문에서는 이산사건 시스템을 계층적 구조를 가진 모듈화된 모델로 모델링하는 DEVS 형식론[3]에 기반하여 시뮬레이션 환경을 구축한다. Zeigler가 제안한 DEVS 형식론은 집합이론에 기반을 두고 있으며, 이산사건 시스템을 계층적으로 분해하여 나타내기 위하여 두 가지의 모델 클래스 즉, Atomic 모델 및 Coupled 모델로 나누어서 표현한다. 여기서 Atomic 모델은 더 이상 분해할 수 없는 시스템 컴포넌트를 표현한다. Coupled 모델은 여러 개의 컴포넌트 모델 (Atomic 혹은 Coupled 모델)들로 구성된 복합적인 모델을 표현한다[5].

ABC, BC - Couple model  
A, B, C - Atomic model

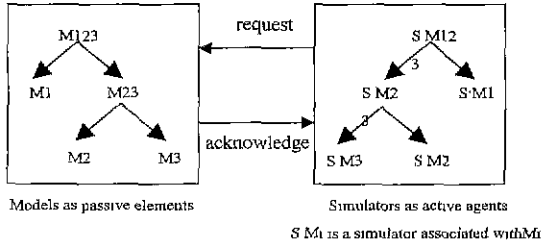


a) DEVS 모델의 계층구조

b) 시뮬레이션 모델 구조

<그림 1> DEVS 모델의 계층적 구조

Abstract simulator[3][4]는 DEVS 형식론에 의해 명세된 모델을 해석하여 시뮬레이션 하는 알고리즘이다. Atomic model을 위한 Simulator, Coupled model을 위한 Coordinator의 두 가지 유형으로 나누어지며, 한 모델 당 한 개의 Abstract simulator가 할당되어 시뮬레이션이 이루어진다.



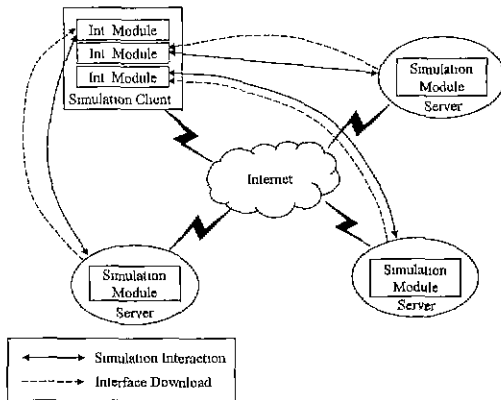
<그림 2> 수동적 모델과 능동적 시뮬레이터

Simulator는 모델의 상태전이와 출력사건 발생을 수행한다. 상태전이는 입력사건이 없이 정해진 시간이 경과하면 내부적으로 발생하는 내부 상태전이와 외부 입력사건이 도착한 시점에 발생하는 외부 상태전으로 나누어지며, 출력사건은 내부 상태전이가 발생하는 시점에 생성된다. Coordinator는 하위 모델간에 사건을 전달하고 하위 모델의 내부 상태전이에 대한 스케줄링을 담당한다.

### 3. 인터넷-기반 분산 시뮬레이션 플랫폼

#### 3.1. 분산-모델 분산-시뮬레이션 형태

2장에서 언급한 바와 같이 기존의 대부분의 웹-기반 시뮬레이션 환경은 분산 시뮬레이션을 지원하지 않는다. 본 논문에서는 인터넷 상에서 분산 시뮬레이션을 지원할 수 있는 분산-모델 분산-시뮬레이션 형태의 웹-기반 시뮬레이션의 플랫폼을 제안하고 구현된 환경을 설명한다.



<그림 3> 분산-모델 분산-시뮬레이션 형태

본 논문의 분산-모델 분산-시뮬레이션 형태에서는 전체 시뮬레이션 모듈이 인터넷상에 분산 개발되고 있는 것을 가정한다. 임의의 클라이언트에서 전체 모듈을 통합하여 시뮬레이션할 경우에는, 전체 모듈을 연결하는 Coupled 모델을 만든 후 분산되어진 시뮬레이션 모듈에 대한 인터페이스들을 클라이언트에 내려 받고, 각 시뮬레이션 모듈들은 개발되어진 사이트에서 순차적으로 수행된다. 전체 시뮬레이션은 클라이언트에 내려 받은 인터페이스들을 통해 시뮬레이션 모듈간의 메시지 교환을 통해 분산적으로 수행된다. 이와 같은 분산-모델 분산-시뮬레이션 형태는 기존의 세 가지 형태의 웹-기

반 시뮬레이션을 포함하며 추가하여 다음의 두 가지 형태의 웹-기반 시뮬레이션이 가능하다.

완전 분산 시뮬레이션: 분산된 모듈을 갖고 있는 사이트들이 모두 시뮬레이션에 참가하는 분산 시뮬레이션

부분 분산 시뮬레이션: 일부의 모듈들은 한 사이트에 내려 받아져서 시뮬레이션에 참가하는 분산 시뮬레이션

#### 3.2. DEVS 형식론의 확장

하드웨어 또는 네트워크 시스템은 각 모듈간의 신호 전송 시에 신호 지연이 발생한다. Zeigler의 DEVS 형식론에서는 시뮬레이션 모델들 간에 사건 전달 시간은 없는 것으로 정의된다. 따라서 모듈 사이의 사건 전달 시 전송 지연이 있는 시스템의 모델링은 실제 시스템과 모델링 된 시스템 사이에는 구조적 또는 의미론적 차이가 발생하게 되거나, 전송 라인을 모델링하게 되어 모델링이 복잡하여진다. 이를 극복하기 위하여 본 논문에서는 Coupled 모델을 확장하여, Coupled 모델 내의 컴포넌트 모델들 간의 연결상태를 나타내는 Coupling 정보에 모델들 간의 전송 시간 정보를 추가하도록 변경하였다. 또한 스케줄링 시간이 같은 컴포넌트 모델들의 순차적인 시뮬레이션을 위해 도입된 SELECT를 삭제하고 같은 시각에 스케줄링될 수 있도록 하였다. 이를 위하여 Abstract simulator의 알고리즘을 변경하였다.

확장된 Coupled 모델: CM

$$CM = \langle D, \{M_i\}, \{I_i\}, \{Z_{ij}\}, \{W_{ij}\} \rangle$$

D: 모든 컴포넌트 모델 이름의 집합

for each  $i$  in D

$M_i$ : 컴포넌트 모델

$I_i$ :  $M_i$ 의 influencees 집합

for each  $j$  in  $I_i$

$Z_{ij}$ :  $i$ -to- $j$  output translation

$W_{ij}$ :  $i$ -to- $j$  output translation 지연시간

Coupled Model의 수학적 정의 및 의미론의 변경을 반영한, 병렬 시뮬레이션 수행을 위한 Abstract Simulator의 주요 동작은 다음과 같다.

#### Coordinator 동작

컴포넌트 모델 스케줄링:

$$t := \min(\text{컴포넌트 모델들의 스케줄 시간});$$

for 스케줄 시간이  $t$  와 같은 모든 컴포넌트 모델( $M_i$ ) ( $M_i$ 에 상태전이 시작을 명령;

컴포넌트 모델에서 발생한 출력사건의 처리

$$t := \text{사건 발생 시간};$$

for 사건( $E$ )을 받는 모든 컴포넌트 모델( $M_i$ ) (

$$E.time := t + \text{전송지연시간}(W_{ij}) + \Delta;$$

$M_i$ 의 사건 리스트에 사건( $E$ )을 보낸다;

#### Simulator 동작

스케줄링에 의한 상태전이 처리:

if 내부상태전이를 기다리고 있을 경우

출력사건 발생 및 내부상태전이,

else

외부상태전이;

다음 스케줄링 시간 := min(다음 내부상태전이 시간, 대기 사건들의 사건도착시간중 최솟값),

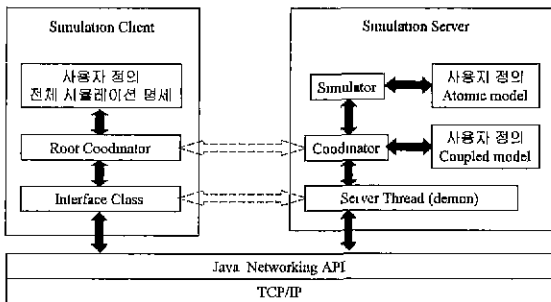
사건 리스트에 도달한 사건(E) 처리 :

```
Event 큐에 사건(E) 삽입;
if 사건 도착시간(E.time) <= 다음 스케줄 시간(tN)
    다음 스케줄링 시간 := E.time;
```

하나의 모델로 가는 사건이 동일 시기에 2개 이상 발생할 수 있기 때문에 Simulator에 사건 리스트를 두어 사건이 대기하도록 하고, 이 때 전송 지연 시간을 반영한다. 큐에서 대기중인 사건에 따른 외부 상태 진이는 사건 도착 시간에 의해 스케줄한다. 또한 전송 지연시간이 0일 경우를 대비하여 최소 전송지연시간( $\Delta$ )을 도입하고, 모든 사건의 전송이 이에 영향을 받도록 하였다.

3.3 시뮬레이션 환경 구현

분산-모델 분산-시뮬레이션 환경에서는 각각의 시뮬레이션 모듈들은 임의의 시스템에 다운로드 되어 수행될 수 있어야 한다. 따라서 기계-독립적이고 네트워크 친화적인 언어인 Java로 분산 시뮬레이션 플랫폼이 구축되었다 (<그림 4>).



<그림 4> 인터넷 기반 분산 시뮬레이션 환경 구조

Simulator, Coordinator' 확장된 Abstract Simulator의 알고리즘을 구현한 Java 클래스이며 사용자가 이를 상속받아 Atomic model과 Coupled model의 명세를 정의하는데 필요한 자료구조 및 함수를 선언한다.

Server Thread: 시뮬레이션 클라이언트의 Interface Class에서 전송된 사건들과 시뮬레이션 명령을 시뮬레이션 서버상의 최상위 Coordinator로 전송하고, 반대로 Coordinator에서 발생시키는 시뮬레이션 정보를 Interface Class로 전송한다. 서버상의 최상위 Coordinator의 관점에서 봤을 때 마치 상위 Coordinator인 것처럼 동작한다.

Interface Class: 시뮬레이션 수행 전에 Java가 제공하는 ClassLoader 클래스에 의해서 시뮬레이션 서버로부터 내려받아서 Root Coordinator의 하위 모델인 것처럼 동작하면서 Root Coordinator와 Server Thread간에 시뮬레이션 정보를 주고받을 수 있게 해준다.

Root Coordinator: 사용자가 정의한 시뮬레이션 명세를 바탕으로 인터넷 상에 분산되어 있는 시뮬레이션 모델들을 Interface Class를 통해 컨트롤하여 전체 시뮬레이션을 수행한다.

위의 분산 시뮬레이션 플랫폼 상에서 전체 시뮬레이션은 다음과 같은 과정을 통해 수행된다.

- 1) 각 시뮬레이션 서버 상의 Server Thread가 활성화된다.
- 2) 사용자가 정의한 전체 시뮬레이션 명세에 따라 Root Coordinator가 시뮬레이션 서버들에서 Interface Class를 다운로드 받는다.

- 3) 내려 받기된 Interface Class가 Root Coordinator의 하위 모델로 등록되고 Interface Class와 Server Thread간에 소켓이 연결된다.
- 4) 초기화 명령이 Interface Class와 Server Thread를 통해 Root Coordinator에서 Simulation Server로 전달된다.
- 5) 시뮬레이션 서버 상의 모델간의 계층구조를 통하여 초기화 명령이 전달되고 Simulator에서 사용자 정의 Atomic model의 명세를 바탕으로 자신의 스케줄시각을 설정한다.
- 6) 스케줄시간 정보가 4),5)의 역순으로 전달된다, 이 때 Coordinator는 컴포넌트 모델들의 스케줄 시간중 최소값을 자신의 스케줄시간으로 설정하고, Server Thread와 Interface Class의 스케줄 시간은 Simulation Server의 최상위 Coordinator의 스케줄 시간과 같은 시간이 설정된다.
- 7) Root Coordinator에서 Interface Class의 스케줄 시간을 기준으로 모델들의 상태전이를 스케줄링하기 시작한다
- 8) 모델의 상태전이 지시가 4),5)의 정로를 통해 Simulator로 전달되고, 모델들의 다음 스케줄 시간과 출력사건이 그 반대의 정로로 Root Coordinator에 전해진다.
- 9) Simulator의 다음 스케줄 시간이 무한대가 되면 스케줄 시간이 변경될 때까지 스케줄링 되지 않는다. Coordinator는 모든 컴포넌트 모델이 스케줄을 기다리지 않는 상태가 되면 다음 스케줄 시간을 무한대로 설정하고, 스케줄을 기다리지 않는다.
- 10) 9)의 과정을 통해서 마침내 더 이상 스케줄링을 기다리는 Interface Class가 존재하지 않게 되면 전체 시뮬레이션이 종료한다.

4. 결론

본 논문에서는 기존의 웹-기반 시뮬레이션의 확장된 형태로서 분산-모델 분산-시뮬레이션 형태를 제시하고 이의 구현을 위하여 인터넷 상에서의 분산 시뮬레이션 플랫폼을 구축하였다. 시뮬레이션 플랫폼은 확장한 DEVS형식론에 기반하여 기계-독립적이고 네트워크 친화적인 언어인 Java를 이용하여 구현하였다. 이러한 환경은 지역적으로 분산되어 시스템을 개발하고 있는 환경 하에서 서로 간의 인터페이스 정의를 분산 시뮬레이션을 통해 검증할 수 있는 플랫폼으로 사용될 수 있다.

현재는 간단한 동기적 분산 시뮬레이션 알고리즘을 사용하고 있으나 수행속도의 향상을 위해 비동기적인 보수적 또는 낙관적 분산 이산 사건 시뮬레이션 알고리즘을 사용하여 플랫폼을 구축하고자 한다. 또한 애플릿 및 서블릿 기술을 사용하여 사용자가 웹 브라우저를 통해 인터넷 상에 구축된 분산 모듈들을 통합하여 분산 시뮬레이션을 수 있는 웹-기반 분산 시뮬레이션 환경을 구축하고자 한다.

5. 참고문헌

- [1] P.A. Fishwick, "Web-Based Simulation; Some Personal Observations", Proc of the WSC, San Diego, CA, Dec., 1996.
- [2] Proc. of 1998 Int. Conf. on Web-Based Modelling & Simulation, San Diego, CA, Jan., 1998
- [3] B. P. Zeigler, Multifaceted Modeling and Discrete Event Simulation, Academic Press, Orland, 1984.
- [4] 조경훈, "Web을 기반으로 한 DEVS 모델의 시뮬레이션 환경", KAIST 석사논문, 1998.
- [5] 김탁관, "이산사건 시스템 모델링 시뮬레이션 기법", 전자공학회지 제 19권 제 1호, 1992.