

# 계층적 DEVS 모델의 비 계층적 분산 시뮬레이션

<sup>†</sup>강원석, <sup>†</sup>김기형

<sup>†</sup>영남대학교 컴퓨터공학과

## A Non-hierarchical Distributed Simulation of Hierarchical DEVS Models

<sup>†</sup>Wonseok Kang, <sup>†</sup>Kihyung Kim

<sup>†</sup>Dept. of Computer Engineering, Yeungnam Univ.

### Abstract

DEVS(Discrete Event System Specification) 형식론은 계층적이고 모듈화된 형태로 이산사건 시스템을 기술한다. 본 논문에서는 DEVS 형식론에 기반한 모델들을 시뮬레이션하기 위한 분산 시뮬레이션 방법을 제시한다. 본 논문에서 제시한 시뮬레이션 방법은 계층적 DEVS 모델들을 비 계층적 모델로 구성하여 시뮬레이션 한다. 제시한 시뮬레이션 방법은 전통적인 계층적인 시뮬레이션 시 발생하는 overhead를 제거한다. 또한 시뮬레이션 놓기화를 쉽게 구현할 수 있고 더불어 시뮬레이션 엔진의 안정성을 높일 수 있다. 제시한 시뮬레이션 방법의 효용성을 보이기 위해 Windows 시스템에 실행 가능한 시뮬레이션 엔진을 구현하여 대규모 물류 시스템으로 시스템 성능을 측정하였다.

있어 분산 시뮬레이션에 많이 이용된다.

DEVS 형식론에 기반한 분산 시뮬레이션은 전통

### 1. 서론

일반적으로 이산 사건 시뮬레이션은 시스템의 성능 분석 및 측정에 많이 사용된다. 그러나 대규모 시스템 시뮬레이션에 있어서 시뮬레이션 수행 시간 증가의 문제점이 남아있다. 분산 시뮬레이션은 이를 시뮬레이션 수행 시간을 줄이는 방법들을 제시하고 있다. 또한 분산 시뮬레이션은 대규모 시스템을 다루기 때문에 모델 검증, 모델의 타당성, 모델의 재사용성과 사용자에 대한 투명성을 제공하여야 한다. Zeigler에 의해 개발된 DEVS 형식론은 이산사건을 기술할 수 있는 정형화된 구조를 가진다[1]. DEVS 모델링과 시뮬레이션은 프로그램 언어에 관계없이 구현할 수

적인 프로세스 기반 분산 시뮬레이션과 다음과 같은 차이점을 가진다[2]. 첫 번째로 DEVS 형식론은 시뮬레이션 모델에 대하여 외부/내부 사건에 대해 구별한다. 두 번째로 계층적인 모델을 구성한다. 이러한 차이점으로 병렬성을 제공하는 DEVS 형식론으로 인해 분산 시뮬레이션은 이를 적용하고 있다[3-5].

본 논문에서는 계층적이고 모듈화된 DEVS 모델에 대하여 효율적인 분산 시뮬레이션 방법을

제시한다(DEVSCluster). DEVSCluster는 본 연구에서 제안한 비계층적 구조를 가지고 있다. 이는 계층적 시뮬레이션에서 발생하는 시뮬레이션 overhead를 제거하고 시뮬레이션 엔진의 안정성을 높인다. DEVSCluster의 효율성을 보이기 위해 대규모 물류 시스템에 대한 시뮬레이션 성능을 측정하였다. 측정결과 기존의 접근법보다 성능이 우수하였다.

본 논문에서는 2장에서 DEVS형식론을 기술하고 3장에서 본 연구에서 제안한 시뮬레이션 방법을 기술하고 4장에서는 시뮬레이션 성능 측정 결과에 대해 기술한다. 마지막으로 5장에서 결론을 맺는다.

## 2. DEVS 형식론

DEVS는 이산사건 시스템을 위한 집합이론에 기반한 형식론이다[1]. DEVS 모델은 실시스템의 동작을 기술하는 원소(atomic) 모델과 연결(coupled) 모델로 구분되며, 이 두가지 모델을 사용하여 계층적이고 모듈화된 형태로 모델구성이 가능하다.

$$AM = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$$

단,

$X$ : 외부 입력사건의 집합

$Y$ : 출력 사건의 집합

$S$ : 순차 상태의 집합

$\delta_{ext}: Q \times X \rightarrow S$  : 외부 상태 천이 함수

$$Q = \{(s, e) | s \in S \text{ and } 0 \leq e \leq ta(s)\}$$

$\delta_{int}: S \rightarrow S$  : 내부 상태 천이 함수

$\lambda: S \rightarrow Y$  : 출력 함수

$ta: S \rightarrow R^+ [0, \infty]$  : 시간 진행 함수

원소 모델 AM은 외부 입력 사건  $x$ 에 의해 상태

천이( $s' = \delta_{ext}(s)$ )를 일으키며 이에 대한 반응으로 시간진행함수  $ta(s')$ 시간 후에 출력 사건  $y$ 를 일으키게 된다. 이때 모델의 상태는 집합  $S$ 의 원소  $s$ 로 표현된다. 출력 사건  $Y$ 를 일으킨 후에는 내부 상태 천이를 하게 되는데 이 함수가  $\delta_{int}$  함수이다. 그러나 만일  $ta(s')$ 이전에 다른 외부 입력 사건  $x'$ 을 받게 되면 새로운 외부 상태천이를 하게 되며 새로운 사이클이 진행되게 된다.

연결 모델은 모델의 결합 및 연결 상태를 기술하게 된다.

$$CM = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, SELECT \rangle$$

단,

$D$ : 컴포넌트 이름 집합

For each  $i$  in  $D$

$M_i$  : 컴포넌트 모델

$I_i$ :  $i$ 의 출력에 연결된 컴포넌트 집합

For each  $j$  in  $I_i$

$Z_{i,j} Y_i \rightarrow X_j$ :  $i$  to  $j$  출력 변환

$SELECT: 2^M - \phi \rightarrow M$ : 같은 시간에 내부 상태 천이를 하려는 모델중 우선순위를 정해주는 함수

연결 모델 CM은 컴포넌트(원소모델이나 또 다른 연결모델이 될 수 있다) 구성되며,  $\{I_i\}$ 와  $\{Z_{i,j}\}$ 는 컴포넌트들간의 연결을 나타낸다.

연결 모델과 원소모델을 이용하여 사용자는 계층적인 모델 구성을 할 수 있고 수학적 형식론에 기반함으로써 모델 재사용, 논리적 분석등의 기법을 활용할 수 있는 장점을 가진다.

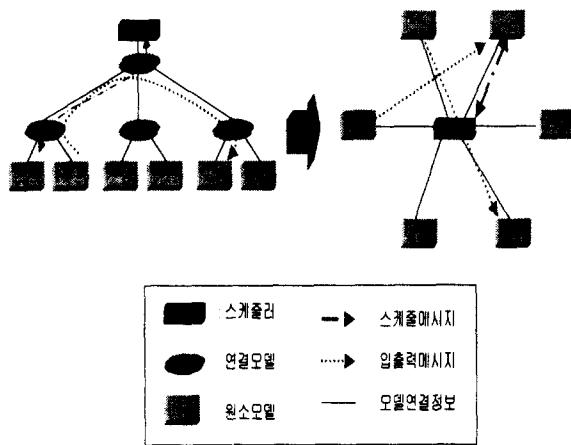
## 3. 비계층적 분산 시뮬레이션 구조

## (DEVSCluster)

본 연구에서는 새로운 DEVSCluster라는 분산 시뮬레이션 구조를 제시하였다. 기존의 계층적 시뮬레이션 구조와는 달리 DEVSCluster는 비계층적 분산/단일 시뮬레이션을 수행한다. 3.1 절에서는 DEVSCluster의 단일 노드에서의 시뮬레이션 수행 구조에 대해 설명하고 3.2절에서는 분산 노드에서의 시뮬레이션 수행 구조에 대해 설명한다.

### 3.1 단일 노드를 위한 시뮬레이션

DEVSClusters는 기존의 계층적 시뮬레이션 방법을 비계층적 방법으로 시뮬레이션하는데 목표를 두었다. (그림 1)은 단일 노드에서의 DEVSCluster의 시뮬레이션 구조를 나타내고 있다. (그림 1)에서 왼쪽 부분은 기존의 DDEVS[7-9]에서 적용하였던 방법을 나타내고



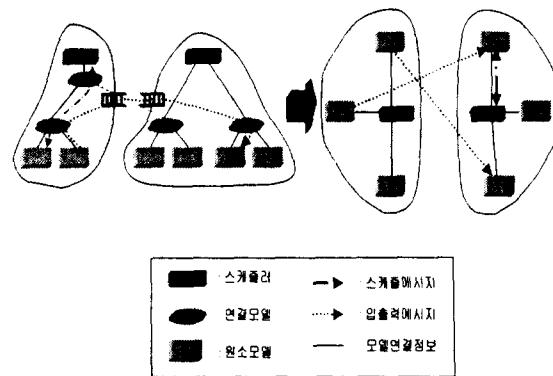
(그림 1) 단일 노드에서의 시뮬레이션 구조

오른쪽 부분이 DEVSCluster의 구조이다.

앞에서 언급한바와 같이 DEVS 모델은 2가지의 모델 형태(원소 모델, 연결 모델)를 취하고 있다. 연결 모델은 다른 모델들 사이의 연결을 기술한다. 연결 모델의 주된 기능은 두 가지로 들 수 있는데 첫 번째가 자신의 자식 모델들에 대한 사건 처리이고 두 번째가 외부 사건을 다른 모델

들에게 전달하는 것이다. 본 연구에서는 이를 연결 모델들을 제거하여 수평적 원소 모델들만이 존재하는 시스템으로 구현하였다. 원소 모델은 외부/내부 사건 모두를 처리하는 기능을 제공하게 구현되게 하여 다른 원소 모델들과 사건을 교환할 수 있게 하여야 한다.

기존 DEVS 시뮬레이션 시스템과 주된 변화는 내부 사건 처리에 중점을 두었다. 기존의 계층적 처리는 내부 사건 처리 시 각 연결 모델들이 자신의 자식 모델들에 대하여 내부 사건을 처리하였다. 그래서 사건 처리는 계층적으로 계승될 수 있었다. 반면 DEVSCluster는 사건의 스케줄링을 담당하는 중앙의 스케줄러를 가진다. 중앙 집중형 사건 처리는 기존의 계층적 스케줄링과 비교할 때 많은 사건을 처리해야 하는 overhead가 존재 할 수 있으나 이를 해결하기 위해 SplayTree를 사용하여 해결하였다.



(그림 2) 분산 노드에서의 시뮬레이션 구조

### 3.2 분산 노드를 위한 시뮬레이션

분산 DEVSCluster는 기존의 분산 시뮬레이션 방법보다 간단한 구조를 가지고 있다. (그림 2)는 분

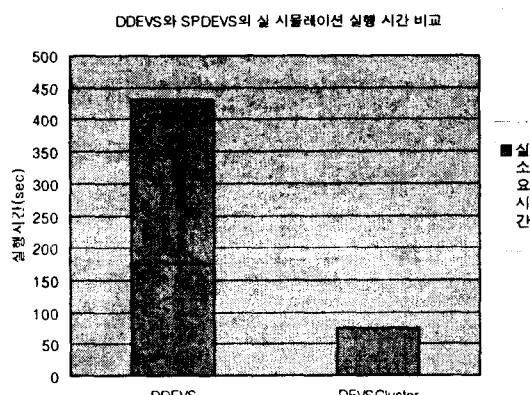
산 DEVSCluster를 잘 나타내고 있다. 기존의 Time Warp 동기화를 위해 명시적 메시지 큐를 사용한 분산 시뮬레이션에 반해 분산

DEVSCluster는 CORBA, DCOM등과 같은 분산 객체 호출 방법을 적용할 수 있다. 이는 분산 시뮬레이션 방법에 있어서 적합하다. DEVSCluster는 하나의 컴퓨터 머신에서는 C++로 구현된 시뮬레이터와 다른 컴퓨터 머신에서는 JAVA로 구현된 시뮬레이터와 시뮬레이션을 할 수 있다.

분산 DEVSCluster의 주된 개선점은 다른 컴퓨터 머신과 사건 교환에 두고 있다. 기존 분산 시뮬레이션의 외부사건은 명시적 메시지를 만들지만 분산 DEVSCluster는 객체 호출을 통하여 명시적 메시지 생성이 필요없이 직접 수행한다. 이는 사건 처리 동기화를 간편화할 수 있으며 신뢰성 높은 시뮬레이션이 가능하다. 기존의 분산 시뮬레이션에 대해서는 모델의 복잡도가 높아질 수록 비동기적 사건이 명시적 메시지 큐에 쌓이게 되어 복잡한 rollback, annihilation등이 발생하여 분산 시뮬레이션에 있어서 문제가 있다. DEVSCluster는 명시적 메시지 큐를 제거하고 직접적인 사건 교환법을 적용하였다.

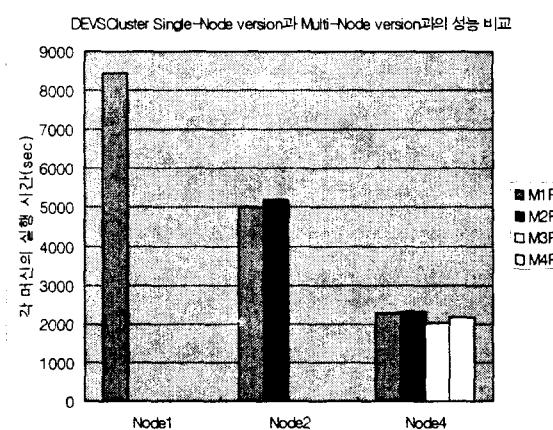
#### 4. DEVSCluster의 성능 측정

본 연구에서 DEVSCluster의 효율성을 측정하기 위해 물류 시스템 모델들을 사용하여 성능을 측정하



(그림 3) 단일 노드에서의 DDEVS와 DEVSCluster 성능 비교  
였다. (그림 3)은 단일 노드에서의 DDEVS와

DEVSCluster의 성능 비교를 나타내고 (그림 4)는 DEVSCluster를 네 개의 컴퓨터 머신까지 증가하면



(그림 4) 분산 노드에서의 DEVSCluster의 성능  
서 분산 처리의 결과이다.

#### 5. 결론 및 향후계획

본 논문에서는 DEVS형식론에 기술된 모델을 위하여 새로운 분산 시뮬레이션 방법을 제안하였다. 제안된 DEVSCluster는 기본적인 기능이 계층적 사건 처리를 비계층적 사건 처리로 구현함으로써 기존 분산 시뮬레이션의 overhead를 제거하였다. DEVSCluster의 성능을 보이기 위해 본 연구에서는 물류 시스템을 이용하여 기존 시뮬레이션 방법과 비교하였다.

#### 참고문헌

- [1] Zeigler, B.P. 1984. Multifaceted Modeling and Discrete-Event Simulation. New York: Academic Press.
- [2] Fujimoto, R.M. 1990. Optimistic approaches to parallel discrete event simulation, Transactions of the Society for Computer

Simulation International, vol. 7, no. 2, October,  
pp. 153-191.

[3] Christensen, E.R. and Zeigler, B.P. 1990.  
Distributed discrete event simulation:  
Combining DEVS and Time Warp.  
Proceedings of the SCS Eastern  
Multiconference on AI and Simulation Theory  
and Applications. Simulation Series.

[4] Zeigler, B.P. 1976. Theory of Modeling and  
Simulation. New York: John Wiley & Sons.

[5] Fujimoto, R.M. 1990. Parallel discrete  
event simulation. Commun. ACM vol. 33, no.  
10, pp.33-53.

[6] Jefferson, D.R. 1985. Virtual time. ACM  
Transactions on Programming Languages and  
Systems, vol. 7, no. 3, July, pp.404-425

[7] Kim, K.H., Seong, Y.R., Kim, T.G., and  
Park, K.H. 1995. Distributed optimistic  
simulation of hierarchical DEVS models.  
Proceedings of the 1995 SCS Summer  
Simulation Conference, Ottawa, Canada,  
pp.32-37.

[8] Kim, K.H., 1996. Distributed Simulation  
Methodology Based on System Theoretic  
Formalism: An Asynchronous Approach.  
Doctoral Dissertation, EE Department, KAIST.

[9] Kim, K.H., Seong, Y.R., Kim, T.G., and  
Park, K.H. 1996. Distributed Simulation of  
Hierarchical DEVS Models: Hierarchical  
Scheduling Locally and Time Warp Globally,  
Transactions of the Society for Computer  
Simulation International vol. 13, no. 3,  
pp.135-154.

[10] Kang, W.S., Hong, J.S. and Kim, K.H.,  
1999. An Integrated Visual Modeling  
Environment for Distributed Simulation.  
Proceedings of IEEE Region 10 Conference,  
pp.1502-1505.