

분산 시뮬레이션을 위한 HLA DEVS-Obj-C

환경 구축

최두진, 조대호

성균관대학교 정보통신공학부

경기도 수원시 장안구 천천동 300번지

Development HLA DEVS-Obj-C Environment for Distributed Simulation

Choi Doo Jin, Cho Tae Ho

Sungkyunkwan University

요약

Development of distributed simulation environment must be required in order to simulate the distributed models regionally and inter-operate with running simulations individually. Simulation based on DEVS formalism is difficult to simulate the distributed models. DEVS formalism is modeling methodology. To specify model, this formalism separates behavior and structure, therefore it is able to design complex model easily. HLA is standard framework of distribute simulation environment. It is defined to facilitate the interoperability and the reusability. RTI (Run Time Infrastructure) is software that provides common service to simulation systems and implementation of the HLA Interface Specification. Method of implementation is that modules cooperating with RTI are added to abstract simulator on DEVS simulation environment. On the DEVS simulation environment (DEVS-Obj-C) that already developed, Highest class of abstract simulator uses service that RTI provide, then This environment is able to change DEVS model into Federate and run distribute simulation that inter-operates with the RTI. Because this distributed simulation environment includes convenience of modeling that obtains through the DEVS formalism and accompanies HLA standard, this environment make it possible to simulate with complex systems and heterogeneous simulations

I. 서 론

컴퓨터 시스템과 네트워크의 구조 등이 복잡해짐에 따라, 시뮬레이션을 위한 모델들의 형태와 구

성 요소들도 복잡해지고 특성도 다양해지고 있다. 또한, 데이터베이스의 거대화와 지역적인 분산이 심화됨에 따라 시뮬레이션 모델도 분산되어지고 있다.

기존의 DEVS (Distribute Event

Specification)[1][2][3] 시뮬레이션은 독립적으로 수행되어지는 시뮬레이션과 복잡한 시스템의 디자인과 모델링에 편리함이 있지만, 분산되어진 모델들의 시뮬레이션과 여러 시뮬레이션을 동시에 수행하고 연동하기에는 적합하지 않다.

HLA (High Level Architecture)는 시간의 동기화, 시뮬레이션 객체들간의 메시지 통신, 이러한 객체들의 재사용성 (Reuseability)과 상호 운영성 (Interoperability)을 향상하기 위해 제안되어졌다 [4]. 이미 HLA는 분산 시뮬레이션에서 중요한 상호 운영성을 효과적으로 적용할 수 있는 장점으로, 국방 분야 등 여러 산업분야에서 적용하고자 많은 노력이 되어 왔으며, 사용되어지고 있다.[5][6]

DEVS 시뮬레이션에 HLA 시뮬레이션을 적용한다면, 복잡한 시스템의 구조를 간단히 모델링 할 수 있는 DEVS의 장점, HLA의 상호작용 용이성과 시간의 동기화, 분리된 객체간의 메시지 전송기능을 모두 갖춘 분산 시뮬레이션의 수행이 가능하다.[7][8]

이 논문에서는 독립적으로 시뮬레이션 수행이 가능하면서도 분산 되어진 환경에서 여러 시뮬레이션이 동시에 수행 가능한 분산 시뮬레이션 환경을 제안한다.

II. 배경이론

분산 시뮬레이션 환경의 가장 중요한 배경이 되는 것은, 이미 표준으로 자리잡은 HLA (High Level Architecture)와 분산 시뮬레이션의 하부구조가 되는 RTI (Run-Time Infrastructure)[14]이며, 또한 DEVS (Discrete Event Specification) 형식론[1][2][3]은 모델링 과정에서 사용되어지고 있으며, 이산사건 시뮬레이션을 위해 정립된 모델링 방법론이다.

2.1. HLA (High Level Architecture)

DMSO (Defense Modeling and Simulation Office)에서 개발되어진 HLA는 분산 시뮬레이션을 위한 표준이라고 할 수 있다.

HLA에서는 이 표준화를 위해서 HLA 규칙, Interface Specification, Object template (OMT)의 3가지 구성요소를 정의하고 있다. HLA 규칙[9]은 10개의 항목으로 되어 있으며, 분산 시뮬레이션에서 모델의 개념으로 이해할 수 있는 Federate과 이것들의 집합인 Federation에 관한 것으로 구분되어진다.

Interface Specification[10]은 RTI (Run-Time Infrastructure)에 대한 표준으로 정의되며, federate과 RTI의 상호 작용시 필요한 부분에 대하여 관리 체계를 정의한다. OMT[11]는 시뮬레이션시에 필요한 객체의 정보, 즉 Federate에 표현되어야 하는 정보들의 공통된 표현방법을 정의하고 있다.

2.2. RTI (Run - Time Infrastructure)

RTI란, HLA에서 정의하고 있는 Interface Specification의 구현 결과물로서, 시뮬레이션 시스템에게 공동의 서비스를 제공하기 위한 Library이다. RTI가 제공하는 서비스에는 시뮬레이션과 통신의 분리, Federate사이의 정의와 관리 체계를 제공한다.

관리 체계로는 Federation의 생성과 소멸등에 관한 Federation Management, Federate간의 통신을 위한 객체와 상호작용 (interaction)의 허용과 인증을 위한 Declaration Management, 시뮬레이션의 대상객체 (Object Instance)에 대한 Object Management, 객체의 속성에 관한 소유권을 관리하는 Ownership Management, 메세지나 정보의 효율적인 전송을 위한 Data Distribution Management, 그리고 시뮬레이션 타임과 동기화를 위한 Time Management의 6가지가 있다.

2.3. DEVS (Discrete Event Specification)

DEVS 형식론은 모델의 행동과 구조를 구분하여 명시함으로써 계층적이고 모듈화 되어진 방법으로 복잡한 모델을 쉽게 디자인하고 설계할 수 있는 모델링 방법론이다. DEVS에서 제안하는 2개의 모델의 구성은 다음과 같다.

Basic 모델

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

여기서,

X : 입력사건의 집합.

S : 순차적 상태의 집합.

Y : 출력 사건의 집합.

$\delta_{int} : S \rightarrow S$: 내부 전이 함수.

$\delta_{ext} : Q \times X \rightarrow S$: 외부 전이 함수.

$\lambda : S \rightarrow Y$: 출력 함수.

$ta : S \rightarrow R+0 \rightarrow \cdot$: 시간 진행 함수.

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

e : 최근의 상태 전이(state transition) 이후로 흐른 시간.

다음은 *coupled model*의 구조이다.

$$DN = \langle D, M_i, I_i, Z_{ij}, select \rangle$$

여기서

D : coupled 모델의 구성요소 i 에 대한 이름의 집합.

M_i : 구성요소가 되는 basic 모델.

I_i : 모델 i 의 영향을 받는 모델들의 집합.

Z_{ij} : I_j 의 원소 각 j 에 대해서 i 에서 j 로의 출력 번역 함수.

$Select$: 타이 브레이킹 함수.

III. 연구방법

HLA에서의 분산 시뮬레이션은 DEVS 모델과 DEVS 시뮬레이션을 Federate와 Federation으로

하여 DEVS 시뮬레이션의 모델설계의 편리함과 RTI에서 제공하는 분산되어진 시뮬레이션의 동기화, 메세지 전송 서비스 등을 이용할 수 있는 분산 시뮬레이션의 환경을 구축하고자 한다.

3.1. 분산 시뮬레이터의 설계

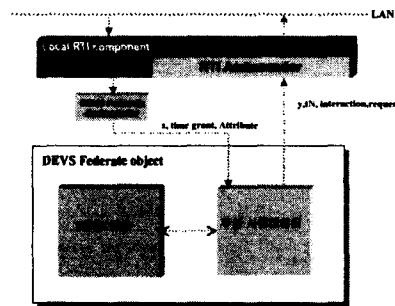


그림 3.1 분산 시뮬레이터의 설계

3.2. DEVS Federate Object의 구조

DEVS Federate Object는 기존의 모델의 구조와 작동을 나타내는 부분과 추상 시뮬레이터의 2부분으로 나누어 질 수 있다. 추상 시뮬레이터의 최상위 프로세서인 Root-co-ordinator에서 RTI와 연동하기 위한 DEVS Federate Ambassador의 객체를 갖고 있으며, 이 객체를 통해서 RTI에서 들어오는 메시지를 처리할 수 있으며, LRC (Local RTI Component)를 통하여 외부로 메시지를 전송할 수 있도록 RTIAmbassador 클래스의 객체도 포함하고 있다.

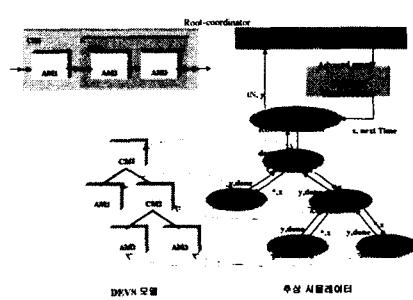


그림 3.2 DEVS Federate Object의 구조

분산 시뮬레이션 환경에서의 인터액션은 크게 외부에서 이루어지는 인터액션과 내부에서 이루어지는 인터액션으로 구분할 수 있다.

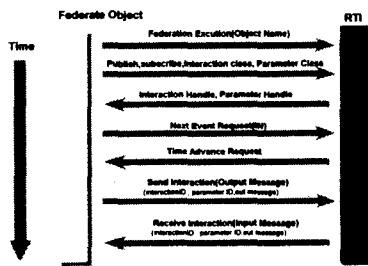


그림 3.3 외부 인터액션

내부에서의 인터액션은 DEVS 시뮬레이션에서 정해지는 사건의 시간 스케줄링과 *-메시지와 done-메시지, 그리고, 모델의 입출력이 되는 메시지의 전달을 통한 것이고, 외부의 인터액션은 Federate와 RTI에서 행해지는 인터액션으로서, 클래스와 파라미터 클래스의 정보 전달과 저장, Federate에서 발생하는 입출력 메시지의 전달로 인해 발생하는 인터액션과 RTI를 연동하는 Federate간의 주고받는 인터액션이다.

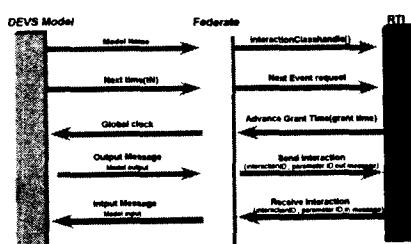


그림 3.4 외부 인터액션과 내부 인터액션의 관계

외부 인터액션은 DEVS Federate Object와 RTI 연동하기 위해 필요한 상호 작용이다.

내부 인터액션과 외부 인터액션 사이의 관계는 그림에서 볼 수 있듯이, 같은 시뮬레이션 시간에 발생하고, 또한 인터액션에 사용되는 변수들과 메

시지들도 동일하다. 즉, 두 인터액션은 따로 발생되어질 수 없다는 것을 의미한다.

IV. 분산 시뮬레이션의 적용

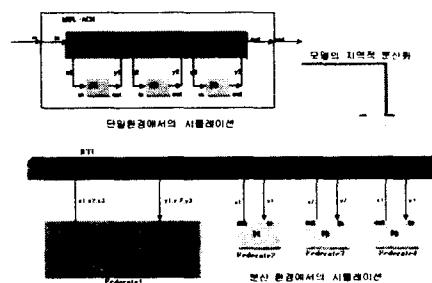


그림 4.1 대상 시스템의 분산

4.1. 시뮬레이션의 결과

본 연구에서 구현된 분산 시뮬레이션 환경은 제시되는 시뮬레이션의 결과를 통하여 시간의 동기화가 제대로 이루어진다는 것을 알 수 있다.

다음 그림은 각 모델에 입력이 발생된 시간을 보여주고 있다.

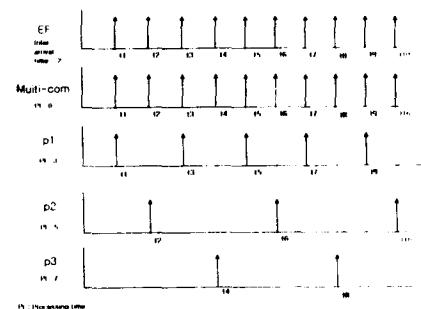


그림 4.2 시간의 진행에 따른 모델의 입력

이러한 결과로 볼 때 분산 시뮬레이션 환경은 grant time의 흐름을 시뮬레이션 시간으로 하여 올바르게 진행하며, 모델의 상태 변이에 따른 동작도 정확히 발생함을 알 수 있다.

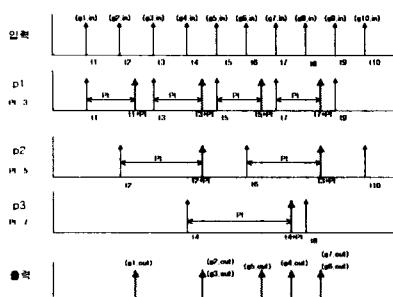


그림 4.3 모델의 출력

V. 결론 및 향후과제

본 연구에서는 DEVS-Obj-C 와 HLA를 기반으로 하는 분산 시뮬레이션 환경의 구현을 제안하였다.

검증 모델인 Multi-co-ordinator의 시뮬레이션을 통하여 환경 구현에 적용된 설계가 틀리지 않았다는 것을 알 수 있었다.

향후과제로는 도큐멘트 뷰 구조에 의해 입출력이 텍스트 형태로 나타나는 시스템에 GUI를 적용하여 그래픽을 통하여 시뮬레이션의 동작을 볼 수 있도록 하는 것과, 계층적 에니메이터의 추가를 통하여, 에니메이션을 이용한 분산 시뮬레이션 환경이 요구되어진다.

참 고 문 헌

- [1] Bernard P. Zeigler, Object-Oriented Simulation with Hierarchical, Modular Models. San Diego, CA, USA: Academic Press, 1990.
- [2] B. P. Zeigler, Multifaceted Modeling and Discrete Event Simulation. Orlando, FL: Academic, 1984.
- [3] B. P. Zeigler, Theory of Modeling and Simulation, John Wiley, NY, USA, 1976, reissued by Krieger, Malabar, FL, USA, 1985.
- [4] J. S. Dahaman, F. Kuhl, and R.

Weathrly, "Standards for Simulation: As Simple As Possible But Not Simple The High Level Architecture For Simulation", SIMULATION, 71(6), pp.378-387, 1998

[5] Ulrich Klein, et al., "Traffic Simulation Based on the High Level Architecture", in Proceedings of the 1998 Winter Simulation Conference, Washington DC, USA, Dec. 13-16, 1998, pp.1095-1103

[6] C. D. Pham and R. L. Bagrodia, "Building Parallel Time-Constrained HLA Federates: A Case Study with the PARSEC Parallel Simulation Language", in Proceedings of the 1998 Winter Simulation Conference, Washington DC, USA, DEC,13-16, 1998, pp.1555-1562

[7] B. P. Zeigler, Hyup Cho, J.S. Lee, H. S. Sarjoughian, G. Ball, "The DEVS/HLA Distributed Simulation Environment And Its Support for Predictive Filtering"

[8] B. P. Zeigler, G. Ball, Hyup Cho, J.S. Lee, H. S. Sarjoughian, "Implementation of the DEVS Formalism over the HLA/RTI : Problems and Solutions"

[9] DMSO, "HLA Rules version 1.3", 1998 IEEE Standards Draft, February 5, 1998

[10] DMSO, "HLA Interface Specification version 1.3", 1998 IEEE Standards Draft, April 20, 1998

[11] DMSO, "HLA Object Model Template Specification version 1.3", 1998 IEEE Standards Draft, April 20, 1998