가상 에이전트 기반 시뮬레이션 소프트웨어의 쾌속 프로토타이핑 지원 환경

최창범*, 김탁곤*

Rapid Prototyping Application Environment for Virtual Agent based Simulation Software

ChangBeom Choi, TagGon Kim

Abstract

Checking the implementation and design of virtual agent and simulation software against the user requirements is important for the virtual agent based simulation software development. Therefore, analyzing the user requirements and checking the implementation of simulation software rapidly are required to the developers during the virtual agent based simulation software development process. Moreover, testing the software and reflecting the modified requirements rapidly are required when the user requirements are changed. This paper proposes the development application environment for rapid prototyping of virtual agent based simulation software. The proposed application environment utilizes the discrete event simulation formalism and C-Interpreter to review the user requirements and check the result of modified code in real time.

Key Words : Virtual Agent, Agent Based Simulation, Runtime Testing

* 한국과학기술원

1. 서론

최근 전 세계적으로 자원과 시간, 인력을 낭비하지 않고 제품과 실험을 수행하기 위하여 모델링 및 시뮬레이션 (Modeling & Simulation: M&S) 공학이 주목을 받고 있다. M&S 공학은 실세계를 다양한 수준으로 모사함으로써 수행된 시뮬레이션 결과를 다양한 분야에 활용할 수 있다. M&S의 활용 예로는 산업 쪽으로 다양한 산업 제품의 설계와 제조 과정에서 핵심적인 역할을 수행하고 있으며, 국방 분야에서는 워 게임 모델, 항공 시뮬레이터와 같이 훈련 시뮬레이터로 훈련을 수행하여 자원과 시간, 인력 낭비를 줄이고 있다. 특별히 M&S공학 분야에서 보다 실세계를 정확하게 모사하기 위하여 가상 에이전트 기반 시뮬레이션 기법이 사용되고 있으며 산업 체계, 경제 이론의 검증, 교통 체증 시뮬레이션 등과 같이 다양한 분야에서 활용되고 있다 [1-3].

가상 에이전트 기반 시뮬레이션의 소프트웨어의 개발 과정은 이상적으로 수행되었을 경우 모의 수행하려는 대상 시스템에 대하여 모의 목적에 따라 모의 수준을 결정하고 이로부터 사용자의 요구사항을 분석하여 사용자의 요구사항에 맞는 모델을 설계한 후에 시뮬레이션 소프트웨어로 구현하는 과정을 끝으로 종료된다. 하지만 사용자는 개발이 완료된 소프트웨어를 사용하면서 요구사항을 변경시키는 경우가 많기 때문에 변경된 요구사항에 따라 시뮬레이션 모델의 설계가 변화되고 설계의 변화는 구현의 변화를 발생시킨다. 또한 시뮬레이션 소프트웨어 개발과정에서 발생할 수 있는 수정과정은 사용자의 부주의로 인하여 소프트웨어 버그를 발생시킬 수 있으며 버그를 찾고 이를 수정 과정은 시뮬레이터를 개발 시간에서 많은 비중을 차지한다. 이와 같은 이유로 소프트웨어의 개발 과정에서 사용자의 요구사항을 빠르게 정립하고, 사용자의 요구사항이 실제로 구현 되었는지 신속하게 확인하는 것이 필요하다. 또한 사용자의 요구사항이 변경되었을 때 이를 빠르게 반영하고, 반영된 결과를 신속히 검사하는 환경이 필요하다.

이러한 문제를 해결하기 위하여 쾌속 프로토타이핑(Rapid Prototyping)이 제시되었다 [4]. 쾌속 프로토타이핑이란 사용자의 요구사항의 완성도를 높이기 위하여 대상 시스템의 프로토타입을 개발하여 사용자의 요구사항을 검증하는 방법이다. 이러한 과정은 사용자의 요구사항을 빠르게 확인하고, 이로 인하여 발생할 수 있는 시뮬레이션 소프트웨어의 설계 및 구현 변화를 최소화한다. 이러한 쾌속 프로토타이핑을 지원하기 위해서는 사용자의 요구사항의 변화를 에이전트와 시뮬레이션 모델에 빠르게 적용할 수 있고, 버그를 찾고 이를 수정하는 것을 지원할 수 있는 테스팅 환경이 필요하다.

사용자의 요구사항의 변화에 따라 에이전트와 시뮬레이션 모델을 수월하게 변경할 수 있어야 한다는 요구 사항에 있어서 시뮬레이션 모델의 기능과 역할에 따른 모듈화가 필요하다. 이러한 관점에서 집합론 기반의 모듈적이고 계층적인 특징을 가지고 있는 이산 사건 시스템 명세 (Discrete Event System Specification:DEVS) 형식론은 가상 에이전트 기반 시뮬레이션에 적합하다고 할 수 있다.

본 논문에서는 시뮬레이션 소프트웨어의 쾌속 프로토타이핑을 위하여 DEVS 형식론으로 구현된 가상 에이전트 기반 시뮬레이터에서 시뮬레이션 모델의 테스팅할 수 있는 환경을 제안한다. 본 개발 지원 환경은 사용자가 이산 사건 시뮬레이션 형식론과 C-Interpreter를 사용하여 개발된 가상 에이전트 기반 시뮬레이션 소프트웨어의 프로토타입에서 사용자의 요구사항을 검토하고, 코드의 변경을 실시간에 확인할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 쾌속 프로토타이핑 지원 환경을 구축하기 위한 관련 연구 및 배경 지식에 대하여 소개한다. 3장에서는 쾌속 프로토타이핑 지원 환경을 소개하고, 마지막으로 4장에서는 결론을 맺는다.

2. 관련연구 및 배경 지식

사용자의 요구사항의 변화에 따라 시뮬레이션 모델의 구조를 자유롭게 변경시킬 수 있으며 이를 지원하는 모델 표기 방법으로는 DEVS 형식론이 있다[5]. 또한 시뮬레이션 소프트웨어 개발 방법론에 있어서 시뮬레이션 모델의 재사용성과 시뮬레이션 소프트웨어의 빠른 개발을 위하여 시뮬레이션 소프트웨어의 개발 업무를 분장하여 개발 시간을 단축시킬 수 있는 협동 모델링 방법론이 제시되었다[6]. 본 장에서는 모델링 방법론으로 DEVS 형식론을 소개하고, 시뮬레이션 소프트웨어 개발 방법론으로 협동 모델링 방법론을 소개한다. 그리고 C++ 코드를 실시간에 수행할 수 있도록 개발된 C-Interpreter에 대해서 설명한다.

2.1 DEVS 형식론

DEVS 형식론은 집합론에 근거한 객체 지향적인 특징을 지닌 대상을 모델링할 수 있는 수학적 틀이다. DEVS 형식론은 시뮬레이션 모델의 행위를 모의하는 원자 모델과 여러 원자 모델을 결합하여 계층적으로 모의 대상 시스템을 표현할 수 있도록 하는 결합 모델[2]로 구성된다. 따라서 가상 에이전트의 시뮬레이션 모델들은 기능과 목적에 따라 세부 모델로 나뉘어 원자 모델로 표현될 수 있다. 그리고 사용자의 요구사항의 변화에 따라 해당 원자 모델만을 수정하거나 새로운 원자 모델을 구현하여 결합 모델에 추가하여 시뮬레이션을 수행하도록 하여 개발 시간을 단축시킬 수 있다.

2 본 논문에서는 DEVS 형식론의 원자모델에 대한 검증만을 고려하기 때문에 결합모델에 대해서는 다루지 않는다.

2.2 협동 모델링 방법론

협동 모델링 방법론은 서로 다른 두 분야에서 각각의 역할에 따라 시뮬레이션 모델을 구현하고, 이를 통합하여 시뮬레이션을 수행하는 방법으로 정의될 수 있다. 이러한 협동 모델링 방법론은 VLSI 시스템에서 HW/SW의 통합 설계를 중에 하드웨어 설계와 소프트웨어 설계를 구분하여 구현하고 이를 통합하여 시뮬레이션을 수행하는데 사용되었고, 워게임 시뮬레이션 소프트웨어를 개발하는 과정에서 도메인 전문가와 M&S 전문가가 하나의 시뮬레이션 모델을 구현함에 있어서 M&S 전문가가 시간과 사건에 따라 모델이 수행해야 하는 행동을 정의하고, 미분 방정식 등을 푸는 것과 같이 도메인에 맞도록 시뮬레이션 모델의 행동을 도메인 전문가가 구현하는 방법이다.

이러한 협동 모델링 방법론을 사용하여 시뮬레이션 소프트웨어를 구현하는 경우 M&S 전문가가 담당한 부분과 도메인 전문가가 담당한 부분에 버그가 발생할 수 있으며, 디버깅 작업에서 M&S 전문가와 도메인 전문가가 직접 모델의 파라미터 값과 동작을 시뮬레이션 수행 중에 직접 확인할 수 있는 지원 환경이 필요하다.

2.3 CINT - C Interpreter

CINT는 C와 C++의 인터프리터로 시뮬레이션 등에 사용하기 위한 목적으로 실행시간보다 빠른 개발이 필요한 환경을 위하여 개발되었다[7]. CINT는 은행, 게임 환경 등에 사용되었으며 특별히 에너지 물리학자들을 위한 많은 자료들을 효율적으로 처리하기 하는 ROOT 시스템에 사용되었다. CINT는 표준

C/C++를 지원하지만 CINT는 배열의 정의나 typedef, function call stack의 깊이 등의 제약사항이 있기 때문에 시뮬레이션 소프트웨어 전체를 CINT 기반으로 수행시키는 것보다 각 시뮬레이션 모델을 CINT를 사용하여 프로토타입을 빠르게 개발하고 시뮬레이션 결과를 확인하는 데 사용하는 데 이점이 있다.

3. 쾌속 프로토타이핑 지원 환경

본 논문에서 제안하는 쾌속 프로토타이핑 지원환경은 DEVS 모델로 구성된 시뮬레이션 모델을 CINT 위에서 실행할 수 있게 하는 환경을 제안하는 것이다. 또한 시뮬레이션을 수행할 때만 확인할 수 있는 실행 시간 오류에 대해서 처리할 수 있는 환경을 제안한다.

3.1. 쾌속 프로토타이핑 지원 환경 구조

쾌속 프로토타이핑 지원 환경의 구조는 그림 1과 같다.

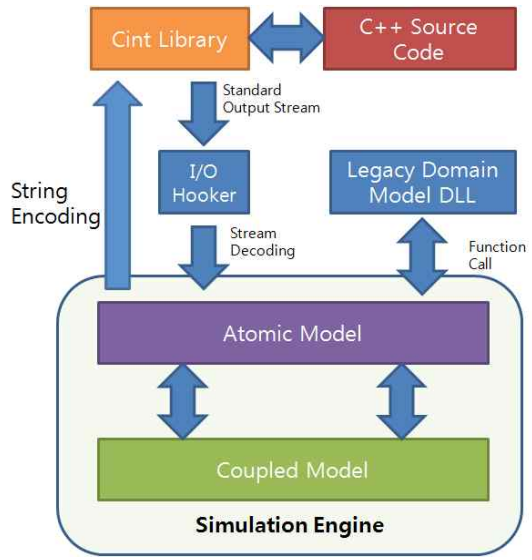

그림 1 쾌속 프로토타이핑 지원 환경 구조

쾌속 프로토타이핑 지원 환경은 기본적으로 DEVS 형식론으로 표현된 시뮬레이션 모델을 수행시키기 위한 시뮬레이션 엔진인 DEVSim++[8] 위에 원자 모델과 결합 모델들의 동작에서 도메인 모델의 함수 수행을 CINT를 사용하여 C++ 코드를 수행한다. 또한 이러한 쾌속 프로토타이핑 지원 환경은 시뮬레이션 도중에 발생하는 오류를 처리하기 위하여 도메인 모델의 DLL을 기본으로 가지고 있다.

CINT 라이브러리는 기본적으로 C-Interpreter와 기존에 개발된 소프트웨어의 연동 기능을 지원하고 있다. 하지만 CINT에서 지원하는 기본 타입으로는 실제 시뮬레이션을 수행할 때의 이벤트를 표현할 수 없다. 따라서 struct와 같은 복잡한 형을 CINT가 인식하고 이를 사용하여 수행할 수 있도록 표준 입출력 스트림을 사용한 인코딩/디코딩 방법이 쾌속 프로토타이핑 지원 환경에서 구현되었다. 이는 CINT에 실행될 코드 블록에 입력 파라미터를 문자열 형태로 인코딩하고, CINT에서 수행되는 코드에서는 표준 입출력 스트림으로 원자 모델이 받아야 하는 값을 출력하면 표준 출력 스트림을 후킹하는 후커가 이를 받아 CINT에서 발생하는 출력 값을 디코딩하여 실제 원자 모델에서 필요한 값으로 변환하도록 설계되었다.

3.2. 쾌속 프로토타이핑 지원 환경을 사용한 검사 기법

쾌속 프로토타이핑 지원 환경은 도메인 전문가와 M&S 전문가가 협동하여 구현한 시뮬레이터의 프로토타입에서 개별 원자 모델의 동작을 검증한다. DEVS 형식론으로 모델링된 원자 모델은 다음과 같이 수학적으로 표현 된다.

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

X: 이산사건 입력 집합

Y: 이산사건 출력 집합
S: 일련의 이산사건 상태의 집합
δ_{ext} : Q × X → S: 외부 천이 함수
Q={(s,e)|s∈S, 0≤e≤ta(s)}:
M의 모든 상태
δ_{int} : Q → Q: 내부 천이 함수
λ : Q → Y: 출력 함수
ta: S → $R_{0,\infty+}$: 시간 진행 함수

사용자는 쾌속 프로토타이핑 지원 환경을 사용하여 시뮬레이션을 수행하고 개별 시뮬레이션 모델을 검사하기 위해서는 위에서 제시한 원자 모델을 쾌속 프로토타이핑 지원 환경에서 동작할 수 있도록 변환해야 한다.

3.2.1 DEVS 모델 변환 방법

DEVS 형식론의 원자 모델은 3개의 집합과 4개의 함수로 구성되며, 특별히 협동 모델링으로 모델링된 시뮬레이션 모델은 시간과 발생된 이벤트에 따라 도메인 모델의 행동, 즉, 함수를 원자 모델에서 호출해 주는 방식으로 동작한다. 따라서 쾌속 프로토타이핑 지원 환경에서는 원자 모델의 4가지 함수가 호출될 때 각 함수 내에서 호출되어야 할 도메인 모델의 함수를 호출해 주도록 구현해야 한다. 즉, 기존의 DEVS 모델의 동작과 쾌속 프로토타이핑 환경 내의 모델이 동일한 순서로 수행되도록 구현해야 한다.

외부 천이 함수의 경우 원자 모델의 정의와 같이 현재 상태에 대하여 입력된 이벤트를 쾌속 프로토타이핑 지원 환경의 모델에 전달하여 시뮬레이션이 수행되도록 해야 한다. 따라서 CINT에 이벤트가 발생하였을 때의 상태와 입력 이벤트에 따라 호출되어야 할 외부 천이 함수가 구현된 C++ 코드를 CINT가 수행하도록 C++ 코드와 입력 파라미터를 변환하여 처리할 수 있도록 한다.

내부 천이 함수의 경우 현재 상태에서 시간 진행 함수에서 지정한 시간을 지난 경우 다음 상태로 천이함으로 이를 CINT가 수행하도록 내부 천이 함수가 구현된 C++ 코드와 현재 상태를 CINT에 전달하고, CINT로부터 받은 값은 다음 상태로 변환한다.

출력 함수는 현재 상태에서 출력 이벤트를 발생시킨다. 따라서 CINT에서 수행되는 출력 코드를 위하여 CINT에 현재 상태를 입력하고, 출력 함수가 명시되어 있는 C++ 코드를 수행하여 그 실행 결과를 출력 이벤트의 형식으로 변환시켜 시뮬레이션 엔진에서 다른 모델로 이벤트를 전달한다.

시간 진행 함수도 다른 함수와 동일하게 현재 상태에 따른 시간 진행 값을 출력하도록 CINT를 사용한다.

3.2.2 쾌속 프로토타이핑 환경을 사용한 검사 기법

쾌속 프로토타이핑 지원 환경은 CINT를 사용하여 사용자가 시뮬레이션 도중에 모델의 소스 코드를 변경할 수 있게 지원한다. 특별히 쾌속 프로토타이핑 환경은 사용자의 요구사항의 변경으로 발생할 수 있는 시뮬레이션 모델의 변화 또는 사용자의 부주의로 인한 오류를 수정하는 경우 현재까지 진행 중이던 시뮬레이션을 종료시키지 않고, 즉시 코드를 수정하여 시뮬레이션을 수행할 수 있어, 변경된 소스코드의 변경을 즉시에 반영하여 시뮬레이션 결과를 확인할 수 있다. 이를 위하여 쾌속 프로토타이핑 환경에서는 CINT에서 발생할 수 있는 오류로 인하여 시뮬레이션이 종료되지 않도록 기존에 협동 모델링에서 시뮬레이션 모델을 구현하는 경우와 동일하게 도메인 모델을 사용한 시뮬레이션 수행도 지원한다. 즉, CINT에서 사용자가 수정한 코드를 수행하는 중에 예외가 발생하는 경우 시뮬레이션이 종료되지 않고, 기존의 도메인 모델이 수행된다. 따라서 사용자는 시뮬레이션 수행 중

에 코드를 변경하여 실시간으로 검사를 수행할 수 있다. 또한 이렇게 CINT로 구현된 시뮬레이션 모델은 실제 시뮬레이션 소프트웨어의 시뮬레이션 모델을 개발할 때 사용될 수 있는 장점을 지닌다.

4. 결론

쾌속 프로토타이핑은 사용자의 요구사항이 가상 에이전트 기반 시뮬레이션 소프트웨어를 개발하는 과정에서 변경되지 않도록 빠르게 프로토타입을 개발하고, 사용자의 요구사항을 프로토타입에서 신속하게 반영하여 사용자의 요구사항을 명확하게 하는 방법이다. 특별히 본 논문에서는 가상 에이전트 기반 시뮬레이션 소프트웨어에서 하기 위해서는 에이전트의 구조를 사용자의 요구 사항에 따라 쉽게 변경할 수 있고, 시뮬레이션 모델을 검증할 수 있는 가상 에이전트 기반 시뮬레이션 소프트웨어를 위한 쾌속 프로토타이핑 지원 환경을 제안한였다.

본 논문에서 제안하는 쾌속 프로토타이핑 지원 환경은 가상 에이전트 기반 시뮬레이션 소프트웨어의 시뮬레이션 모델을 DEVS형식론으로 모델링하고, C-Interpreter를 사용하여 컴파일되지 않은 소스 코드를 실시간에 수행될 수 있도록 하여 시뮬레이션을 중지시키지 않고, 시뮬레이션 모델을 수정하고, 시뮬레이션 모델을 수행할 수 있다.

참고문헌

[1] Eric Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," Proceedings of the National Academy of Sciences. May 14, 2002

[2] Kutluhan Erol, Renato Levy, and James Wentworth, "Application of Agent Technology to Traffic Simulation," United States Department of Transportation, May 15, 2007

[3] Amnah Siddiqah, Muaz Niazi, Farah Mustafa, Habib Bokhari, Amir Hussain, Noreen Akram, Shabnum Shaheen, Fouzia Ahmed &Sarah Iqbal, "A new hybrid agent-based modeling decision support system for breast cancer research", IEEE ICICT, IBA, Karachi, August 15-16, 2009.Breast Cancer DSS

[4] Steven D. Tripp and Barbara Bichelmeyer, "Rapid prototyping: An alternative instructional design strategy," Journal of Educational Technology Research and Development, vol. 38, num. 1, pp.31-44, Mar., 1990

[5] Bernard P. Zeigler, Herbert Praehofer and Tag Gon Kim, Theory of Modelling and Simulation (2nd Edition), Academic Press, 2000.

[6] Chang Ho Sung, Su-Youn Hong and Tag Gon Kim, "Layered Structure to Development of OO War Game Models Using DEVS Framework," SCSC2005, Philadelphia, USA, July, 2005.

[7] CINT, url:http://root.cern.ch/drupal/content/cint

[8] Tag Gon Kim, DEVSimHLA User's Manual, 2007. Available: http://smslab.kaist.ac.kr