

# 워 게임 시뮬레이터 개발 과정 개선을 위한 이벤트 제어 모델 Event Control Model for Improving War-game Simulator Development Process

권세중\* 최창범\* 김탁곤\*

Se Jung Kwon Changbeom Choi Tag Gon Kim

\* 한국과학기술원 전기 및 전자공학과

(발표자 연락처 : 042-350-5454, sjkwon@smslab.kaist.ac.kr)

### Abstract

The development for war-game simulation software may hardly be not the straightforward process. The process can be interrupted by modified scenarios and re-progressed repeatedly. Thus, a rapid prototyping approach is requisite to shorten the repeated process. This paper proposes an approach for enhancing rapid prototyping, which does not modify the model inside as far as possible. Instead, our approach enables the same simulators to generate the other execution results. This approach can be enabled with Event-based Simulation and additional Event Control Model(ECM). Although the simulation engine cannot access the models inside, the simulation engine can handle the Input/Output level data by accessing events. ECM consists of global states and control functions that are mapped to events. Users can control their simulation simply by describing the ECM, instead of modifying simulation models.

Keywords : Simulator Development Process, Event-based Simulation, DEVS

### 1. 서론

일반적으로 소프트웨어(S/W)를 체계적으로 개발하기 위해서는 S/W 개발 프로세스가 필요하다. 마찬가지로 모델링 및 시뮬레이션(Modeling and Simulation, M&S) 공학에서도 비슷한 개발 프로세스가 활용된다. 워 게임 시뮬레이터 개발도 소프트웨어 관점에서 비슷한 과정을 적용하지만 일반적인 S/W 개발과는 달리 특수한 측면이 있다. 워 게임 시뮬레이터 개발 시에 군사 전문가가 도출하는 요구사항을 M&S 전문가가 이해하기 쉽지 않으며 시뮬레이터의 결과에 따라서 요구사항이나 시뮬레이션 목적이 크게 변경될 수 있다. 따라서 워 게임 시뮬레이터는 일반 S/W에 비해 개발과정이 더 복잡해지거나 연장될 수 있다. 이러한 특성은 워 게임 시뮬레이터 개발 과정이 폭포수 모델과 같이 순차적으로 이뤄지기 보다 반복적으로 이뤄질 수 있다는 것을 뜻한다. 따라서 이를 보완하기 위해 애초에 반복적인 방법의 개발 프로세스를 사용하기도 한다. 반복적으로 프로토타입을 제작하여 요구사항을 수정해 나가는 Sawtooth 모델[1]과 같이 여러 번의 시뮬레이터 제작과 수정을 반복한다.

수 있다. 도출된 요구사항으로부터 개발과정을 통해 개발된 시뮬레이션 S/W는 실험 틀(Experimental Frame, EF), 해당 모델(Target Model), 시뮬레이션 엔진(알고리즘)으로 이루어지게 된다. 시뮬레이터 개발 과정에서 수정된 요구사항을 반영하는 것은 EF의 수정을 통해 이루어진다. 하지만 EF가 입력하는 시나리오로는 모든 것을 반영할 수 없기 때문에 모델을 수정하게 되는데, 수정된 모델은 신뢰도가 이전에 비해 떨어지게 되어 이를 검증하는 과정이 다시 진행되어야 하고 개발 시간이 증가한다.

본 논문은 이러한 반복되는 과정에서의 개발 시간을 단축하는 것에 목적을 두고, 빠르게 이산 사건 시뮬레이션 모델의 결과를 수정하는 이벤트 제어 모델을 제안한다. 이것을 추가하여 기존의 시뮬레이션 모델을 수정하지 않고 수정된 결과를 생성한다.

### 2. 이벤트 제어 모델 및 적용 사례

본 논문은 이벤트 제어 모델을 위해서 이벤트 기반 시뮬레이션 개념을 도입한다. 이벤트 기반 시뮬레이션은 기초적인 이산 사건 시뮬레이션 알고리즘으로 이벤트를 기본 단위로 하여 시뮬레이션을 진행한다[2]. 이벤트 기반 시뮬레이션은 객체지향적인 접근이 아니지만 알맞은 해석기(Interpreter)를 구현하면 DEVS와 같은 객체지향 시뮬레이션 모델들도 실행할 수 있다[3].

이벤트 기반 시뮬레이션은 리스트를 통해 이벤트를 시간 순서로 관리하여 실행한다. 실행된 함수는 다시 다음 이벤트를 발생시키는 형태로 시뮬레이션이 진행된다. 이벤트 제어 모델은 이때 발생하는 이벤트에 매핑된다. 이벤트는 데이터 이벤트와 시간 이벤트로 이뤄지는데, 데이터 이벤트를 통해 모델간의 메시지가 전달되고 시간 이벤트를 통해 모델의 실행 시간이 스케줄링된다. 시뮬레이션 엔진은 객체지향 모델의 내부에 접근

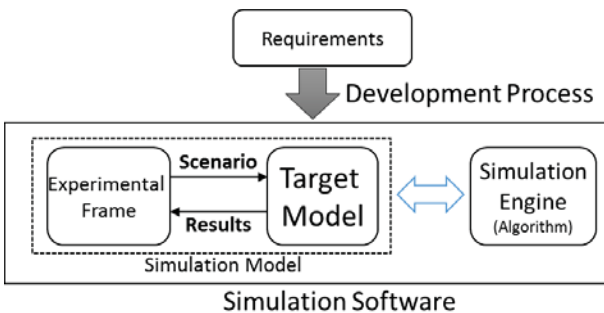


그림 1. 시뮬레이션 소프트웨어의 구성

일반적인 시뮬레이션 S/W는 그림 1과 같이 그려볼

할 수 없지만 모델 밖으로 공개되는 I/O 수준의 정보는 그림 2와 같이 이벤트에 접근함으로써 관리할 수 있다.

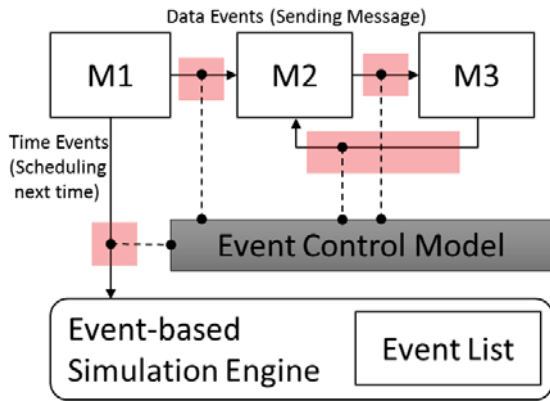


그림 2. 이벤트 제어 모델의 동작점(Access Point)

- $E = \{E_{time}, E_{data}\}$  is a set of events
- $E_{time} = \langle T, t_N \rangle$   
 $T$  is a target function(model)  
 $t_N$  is a next scheduled time
- $E_{data} = \langle T, \{V_i\} \rangle$   
 $T$  is a target function(model)  
 $\{V_i\}$  are variables
- $ECM = \langle \{f_n\}, S \rangle$   
 $f_n : E \times S \rightarrow \{E \cup \emptyset\} \times S$  is a control function  
 $S$  is set of global states

이벤트 제어 모델은 위와 같이 제어 함수와 상태 변수로 이루어져 있으며 제어 함수는 이벤트를 입력으로 받아서 변조된 이벤트를 내놓는다. 기본적으로 이벤트 지향 모델[2]의 형태를 따랐으며 상당히 자유도가 높은 명세 방법을 제공한다. 이러한 모델들은 이벤트 기반 실행 시에 함께 실행된다. 사용자는 필요에 따라 이벤트 제어 모델을 기술함으로써 시뮬레이션 중에 이벤트를 접근할 수 있어 다양하게 시뮬레이션을 제어할 수 있다. 대표적인 예를 들어보면 다음과 같다.

1) 장비의 오차 반영: 장비의 실동작은 약간의 오차를 포함하기 때문에 시뮬레이션 과정에서 이를 확률변수로 표현한다. 개발과정에서 중요하지 않은 요소라 생각되어 확률 변수화 하지 않았던 변수의 오차 반영이 필요할때 이벤트에 담겨 있는 해당 변수 값을 토대로 오차를 반영할 수 있다. 예를 들어 어뢰 발사과정[3]을 모의 하고 있다 가정하면 개발 초기에는 반영하지 않던 발사각 오차를 이벤트 모델을 통해 반영하여 기존의 값을 오차가 반영된 값으로 실행중에 수정할 수 있다. 이때 이벤트 함수는 C++ 언어에서 다음과 같은 형태로 간단히 구현될 수 있다. 이벤트에 담긴 값의 이름이 'Angle'이라고 할때 이 이름에 해당하는 값을 기준으로 지수분포를 따르는 오차를 반영한다.

```
boolean ControlFunction( CEvent ev ){
    double newValue = genExponential(ev.GetValue("Angle"));
    ev.SetValue("Angle", newValue);
}
```

2) 환경 변수의 반영: 온도나 습도, 바람 등과 같은 환경 변수가 있을 때 환경에 따른 변화를 일괄적으로 적용해야 할 수 있다. 이를 위해 이벤트 제어 모델을 구현하여 해당하는 이벤트에 일괄 적용할 수 있다.

3) 새로운 모델의 대체: 요구사항이 추가되어 새로운 (부)모델을 추가해야 할 때 이벤트 제어 모델로 대신 할 수 있다. 대어되진 시뮬레이터에 재머를 추가하는 예[5]를 생각해보면, 재머 모델을 추가하기 보다 소나가 발생시키는 위치 데이터를 일정 확률함수를 가지고 삭제 시킴으로 마치 재머가 동작하여 신호가 전달되지 않는 것처럼 수정할 수 있다.

3. 결론

지금까지 위게임 시뮬레이터 개발 과정을 개선할 수 있는 이벤트 제어 모델을 제안하였다. 위 게임 시뮬레이터의 반복적인 개발과정의 높은 개발 비용과 시간을 줄이는 것이 이벤트 제어 모델의 적용 효과이다. 이벤트 제어 모델을 이벤트 기반 시뮬레이션 알고리즘과 함께 실행하여 기존 모델을 수정하지 않고 수정된 결과를 발생시킨다. 이러한 시뮬레이션 동작 제어는 모델을 수정하는 노력을 줄일 수 있으며, 즉각적으로 수정된 요구사항에 대한 반영함으로 개발 시간을 줄일 수 있다. 또한 적절한 해석기 개발을 통해 다양한 이산 사건 시뮬레이션 모델에 적용될 수 있다는 장점이 있다.

참고문헌

[1] R.B. Rowen, Software Project Management under Incomplete and Ambiguous Specifications. IEEE Transactions on Engineering Management, 37(1), 10-21 (1990).  
 [2] B.P. Zeigler, H. Praehofer, and T.G. Kim, Theory of Modeling and Simulation. 2<sup>nd</sup> ed. USA: Academic Press (2000).  
 [3] S.J. Kwon and T.G. Kim, Design and Implementation of Event-based DEVS Execution Environment for Faster Execution of Iterative Simulation. Proceedings of SpringSim, TMS'12. March 26-29, Orlando (2012).  
 [4] K.M. Seo et al., Measurement of Effectiveness for an Anti-torpedo Combat System Using a Discrete Event Systems Specification-based Underwater Warfare Simulator. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 8 (3), 157-171 (2011).  
 [5] S. J. Kwon et al., "Effectiveness Analysis of Anti-torpedo Warfare Simulation for Evaluating Mix Strategies of Decoys and Jammers," AsisSim '2011, Seoul (2011).