
파라미터 행위 모델링 기법을 이용한 전쟁게임 시뮬레이션

War Game Simulation Using Parametric Behavior Modeling Method

이완복*, 김재현**, 김탁곤**
충부대학교 게임학과*, KAIST 전자전산학과**

Wan-Bok Lee(wblee@jcongbu.ac.kr)*, Jae-Hyun Kim(jhkim@smslab.kaist.ac.kr)**,
Tag-Gon Kim(tkim@ee.kaist.ac.kr)**

요약

기존의 객체지향 통합 모델링 방법은 전쟁게임 모델을 두 계층으로 구분하여 개발할 수 있는데, 상위 계층에서는 객체의 추상화된 행위를 모델링하며 하위계층에서는 모델의 세부적인 행위를 표현하게 된다. 이 방법에서는 시뮬레이션 전문가와 군 관련 전문가가 각각의 계층에서 별도로 작업을 할 수 있는 장점이 있다. 본 논문에서는 통합 모델링을 발전시킨 파라미터 행위 모델링 방법을 이용하여 전쟁 게임 시뮬레이터를 구축할 수 있는 방안을 제시한다. 제안한 방법에서는 군 관련 전문가 또는 사용자가 모델링 방법에 대하여 별도의 지식이 없이도 새로운 알고리즘이나 전략을 시뮬레이터에 추가하거나 수정할 수 있는 장점이 있다. 제안한 방법의 효용성을 보이기 위해 구현된 해군 군함의 전쟁게임 시뮬레이터를 구체적으로 소개한다.

■ 중심어 : □전쟁게임 시뮬레이션 | 통합 모델링 | 파라미터 행위 모델링 | DEVS |

Abstract

The object oriented co-modeling methodology, previously introduced, employs a layered approach in war game models development in which an upper layer models abstract behavior of an object and a lower one models details of the object. Within the methodology military domain experts and simulation experts models an object at the upper and the lower layers, respectively in concurrent manner. This paper proposes a method of constructing a war game simulator using parametric behavior modeling technique, which provides a means for military domain experts/users to change model's detailed behavior with no knowledge on modeling semantics. The proposed simulator would support new algorithms or strategies with minimal cost and could be modified even by the users who are ignorant about modeling technique. To demonstrate the effectiveness of the proposed framework, a naval war game simulator is exemplified.

■ keyword : □War Game Simulation | Co-modeling | Parametric Behavior Modeling | DEVS |

1. 서론

시뮬레이션 전문가들은 주어진 모델링 목표에 따라 시스템을 추상화시키고 적합한 모델로 만들게 된다. 이러한 모델을 제작하는 과정에서 시뮬레이션 전문가들은 모델링하고자 하는 시스템에 대하여 자세한 지식을 겸비하고 있어야 하지만, 그렇지 못한 경우가 많다. 또한 실세계(Real World) 시스템을 잘 알고 있는 필드 전문가(Field Expert)는 시스템을 운영할 수는 있어도, 시뮬레이션 모델을 구성하기는 매우 어려운 경우가 많다. 예를 들어, 전쟁게임 시뮬레이션을 구성하는 경우, 전략과 전술을 세우고 실행하는 지휘관이나 군인들은 전쟁 수행 규칙은 잘 알고 있지만, 전쟁게임 시뮬레이션을 수행할 수 있는 시뮬레이션 모델을 잘 만들지는 못한다. 반면에 시뮬레이션 전문가들은 시뮬레이션을 할 수 있는 방법론은 잘 알고 있지만 시뮬레이션을 하고자 하는 대상 시스템에 대한 깊은 지식이 없어서 세밀한 모델 개발을 직접 하기 어렵다. 그러므로 모델링 시뮬레이션 전문가와 필드 전문가가 협력하여야만 시뮬레이션 모델을 쉽게 구축할 수 있다.

이처럼 다른 영역의 전문가들이 작업을 할 경우 각자의 전문성이 다르기 때문에 상호 의사소통이 원만하게 이루어지기 어려우며, 두 전문가 집단 간의 작업 분배가 쉽지 않은 경향이 있다.

UML(Unified Modeling Language)를 이용하여 위 두 전문가 집단 간의 의사소통을 원활히 하고 모델을 쉽게 구축할 수 있는 방안에 대하여 연구된 바가 있다 [1]. 제안된 방법은 수평적인 접근 방식을 취하고 있는데, 필드 전문가가 먼저 UML을 이용하여 사양(Specification)을 정의하면, 시뮬레이션 전문가가 그 사양을 시뮬레이션이 가능한 모델로 추가적인 정보를 붙여서 구현하는 방법이다. 그러므로 전체적인 시뮬레이션 모델은 순차적으로 구축되어진다.

객체지향 통합모델링 방법(Object-oriented co-modeling methodology)[2]에서는 계층적인 방식으로 전체 모델을 구성하고 있다. 상위 레벨에서는 객체들의 추상적인 행위를 표현하고, 하위 레벨에서는 객체들의 세부적인 행위를 표현한다. 그러므로 시뮬레이션 전문가와 필드

전문가가 서로 다른 레벨에서 동시에 작업을 할 수 있다.

일반적으로 초기에 만들어진 시스템 사양서나 요구사항은 개발하고자 하는 시뮬레이터의 기능을 명백하게 제한하지만, 개발된 시뮬레이터는 새로운 알고리즘이나 실행규칙이 적용되도록 변형되어야 할 필요가 많다. 즉 새로운 군사 전략이나 무기 체계의 도입으로 인하여 전력의 증감을 평가하려면 시뮬레이터의 재구성이 필연적으로 요구되는 것이다.

파라미터 행위 모델링은 이처럼 새로운 알고리즘이나 실행규칙을 쉽게 적용할 수 있는 방법인데, 이것은 컴퓨터 프로그램에서 파라미터 또는 옵션 값들을 달리 지정함으로써 프로그램을 여러 용도로 사용하는 것과 비슷하다. 본 논문에서는 동정 링크 라이브러리(DLL)를 이용하여, 시뮬레이터의 행위를 실시간에 변경할 수 있도록 하고 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 통합모델링 방법에 대해 설명한다. 3장에서는 객체지향 모델링 기법인 UML 모델링, DEVS 형식론에 대해 소개하고, 4장에서는 제안하는 파라미터 행위 모델링 기법에 대해 소개하고 시뮬레이터를 자동으로 합성하는 방안에 대해 설명한다. 5장에서는 해군 전쟁게임 시뮬레이션을 예제로서 보이고 6장에서 결론을 맺는다.

II. 통합 모델링

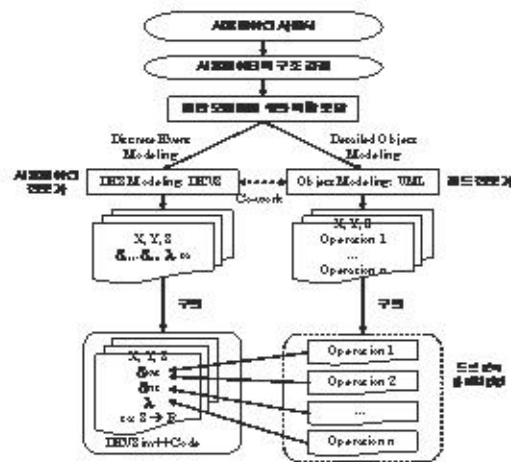


그림 1. UML과 DEVS 형식론을 이용하는 통합

통합모델링 기법은 VLSI 시스템 설계에서 자주 사용되는 HW/SW 통합설계(co-design) 기법과 유사하다 [2]. HW/SW 통합설계 기법은 하드웨어와 소프트웨어 부분을 동시에 고려하여 설계하는 기법으로써 시스템 레벨에서 최적의 설계를 추구하는 기법이다[3][4]. 통합 모델링 과정을 좀 더 자세히 살펴보면 다음과 같다.

먼저 시뮬레이터의 구조를 시스템 요구조건과 사양으로부터 결정한다. 시뮬레이션 전문가는 각 객체를 원자 모델(Atomic Model)로 모델링하는데, 입력 집합, 출력 집합, 상태 집합의 세 집합과 네 가지의 특성 함수들을 정의한다. 만일 모델링되는 객체에 대해 더욱 자세한 규칙이나 연산들이 정의되어야 할 경우 필드 전문가가 UML을 이용하여 그러한 제한 사항들을 정할 수 있다.

다음 단계에서는 시뮬레이션 전문가와 필드 전문가가 긴밀하게 연계되어 시뮬레이션 모델을 설계하게 된다. 이 과정에서는 어떠한 모델들이 이산사건 시스템 계층에서 모델링되어야 하는지 또 어떤 모델들이 이산사건 시스템 계층과 객체 모델링 계층 두 곳 모두에서 모델링되어야 하는지 정해야 한다. 전자의 경우, 필드 전문가가 잘 알고 있는 전문적인 지식이 필요하지 않다. 반면에 후자의 경우 객체의 구체적인 연산이나 행위에 대해서는 필드 전문가가 UML을 이용하여 객체 모델링을 제공하여야 한다. 이러한 관점에서 보면, 이산사건 시스템 모델링은 통합설계의 소프트웨어에 해당하며, UML을 이용하여 작업하는 객체 모델링은 하드웨어에 대응된다고 볼 수 있다. 결론적으로 통합 모델링 방법에서는 시뮬레이션 전문가와 필드 전문가가 동시에 모델 개발을 할 수 있는 여건을 제공한다.

III. UML과 DEVS 형식론

객체지향 모델링 기법은 이산사건 시스템의 요소들을 표현하기에 자연스러우면서도 강력한 패러다임이다. 여기에서는 객체지향 모델링 수단으로 사용되어지는 UML과, 객체지향 이산사건 시스템 모델인 DEVS 형식론에 대해 간략히 살펴본다.

1. UML 모델링

UML(Unified Modeling language)는 시스템을 객체 지향적으로 개발하고자 할 때 사양, 표현, 문서화 등의 과정에 사용될 수 있는 표준 언어이다[5-7].

UML은 소프트웨어의 설계단계에서부터 초기 원형(prototype)을 만들 때까지의 복잡한 단계를 간략히 할 수 있게 해준다. 통합 모델링 기법에서는 객체 모델링 단계에서 UML을 사용하는데, 유스케이스(Use case) 다이어그램, 클래스(Class) 다이어그램, 순차(Sequence) 다이어그램의 세 가지가 활용된다.

1.1 유스케이스 다이어그램(Use Case Diagram)

유스케이스 다이어그램은 시스템의 행위를 나타내는 것으로써 행위자와 시스템간의 관계와 그 유스케이스들의 집합으로 이루어진다. 이 다이어그램에는 행위를 세부적으로 구분하기 위해 액션(action)들을 순차적으로 정의하는데, 각 액션들은 하나 또는 여러 행위자 또는 시스템과 관련되어 발생하게 되며 다른 행위자들에게 알려지게 된다. 이 다이어그램은 시스템의 사용자 관점에서 바라보았을 때의 시스템 요구조건들이라 할 수 있다.

1.2 클래스 다이어그램 (Class Diagram)

클래스 다이어그램은 시스템을 구성하는 클래스들과 인터페이스 그리고 상호 관계들에 대해서 나타낸다. 이 다이어그램은 거의 대부분의 객체 지향 방법들에서 사용되어지는 핵심 요소이며 시스템의 정적인 구조를 표현한다.

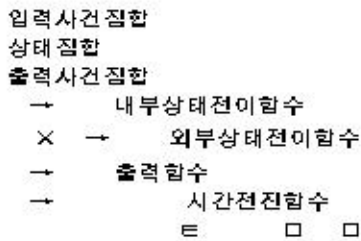
1.3 순차 다이어그램(Sequence Diagram)

순차 다이어그램은 객체들 사이의 메시지 전달 순서를 시간 순으로 나열함으로써 객체들 간의 상호 연동 관계를 나타내는 것이다. 이것은 시스템의 동적 행위를 쉽게 파악할 수 있도록 일련의 세부 동작들의 연속으로 나타낸다. 순차 다이어그램에서는 객체들의 수명을 가로축으로 나타내고, 객체들의 생성과 파생되는 순서를 세로축에서 보여준다.

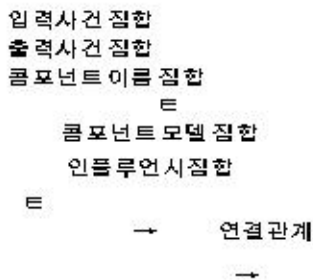
2. DEVS 형식론

DEVS 형식론은 이산사건 시스템을 모듈러하고도 계층적으로 모델링 할 수 있는 방법을 제시한다[8]. 이 형식론에는 모델을 계층적으로 분해할 수 있도록 원자 모델(Atomic Model)과 결합 모델(Coupled Model)의 두 가지 클래스로 나누어서 모델을 표현하고 있다.

원자 모델은 더 이상 나눌 수 없는 모델로서 각 구성 요소의 동적 특성을 시간병세 상태전이시스템(timed state transition system) 레벨에서 기술한다. 원자 모델에는 세 개의 집합과 4개의 특성함수가 있다. 원자 모델의 수학적 정의는 다음과 같다.



결합 모델은 구성요소 모델들 간의 연결 관계를 표현하는 모델로서 구성요소 모델들을 조립하여 더 큰 단위의 모델을 만들 수 있게 한다.



원자모델 또는 결합모델은 그 자체로 실세계 시스템의 객체와 대응된다. DEVS 형식론을 이용하여 모델을 설계할 때에는 top-down 방식으로 접근하는 것이 용이하며, 모델 구현 시에는 bottom-up 방식을 취하는 것이

유리하다. DEVS 형식론에 대한 자세한 설명은 [8]을 참조할 수 있다.

IV. 파라미터 행위 모델링을 이용한 시뮬레이터의 자동 생성 방안

1. 파라미터 행위 모델링

전산학에서 파라미터라는 것은 서브 모듈이나 함수 또는 객체에 정보를 전달하기 위해 사용되는 변수나 값을 의미한다. 특히 함수 호출에 사용되는 파라미터들은 인수라고 불린다. 파라미터는 사용자가 조정할 수 있는 값으로써 프로그램의 행위에 영향을 미치며, 프로그램은 파라미터의 값에 따라 정해진 동작을 보이게 된다.

이와 유사하게 시뮬레이터에 내재되어 있는 모델에서도 파라미터들의 값에 의해 여러 다양한 동적인 특성을 보일 수 있다. 본 논문에서는 이와 같이 파라미터들의 값에 의해 다양한 동적 행위를 보이는 모델들을 일반적으로 파라미터 행위 모델로 지칭한다. 파라미터 행위 모델에서는 파라미터 값에 따라 모델이 다른 동작을 보일 수 있으므로, 미리 파라미터별 행위에 대한 정의를 가지고 있어야 한다.

일반적으로 시뮬레이터는 개발된 이후 그것을 사용하는 중에 새로운 알고리즘이나 전략을 추가하거나 구성할 필요가 있다. 특히 전쟁게임 시뮬레이션에서는 새로운 무기체계가 구축되거나, 그것을 사용하는 전략 시나리오가 달라질 경우, 전력상의 변화가 발생하게 되는데, 시뮬레이션을 통하여 정확한 변화를 예측하려면 시뮬레이터의 구조가 유연해야 한다. 파라미터 모델은 이러한 경우에 새로운 알고리즘이나 전략을 쉽게 수용할 수 있기 때문에 유지 보수가 용이해지는 특징이 있다.

통합 모델링 방법에서는 모델의 인터페이스와 세부적인 행위를 분리하여 명세할 수 있기 때문에 파라미터 행위 모델링과 일치하는 부분이 많다. 통합 모델링 방법에서는 시뮬레이션 전문가와 필드 전문가가 전체 시스템을 이산사건 시스템 계층과 객체 모델링의 두 부분으로 분리한 후 모델링을 진행하게 된다. 이산사건 시스템 계층의 모델에서는 추상화된 행위를 상태 변화 다이어그램(State Transition Diagram)을 이용하여 나타

내는데, 상태 변화과정에서는 전문 분야의 지식과 판단이 요구된다.

예를 들어서 큐 시스템을 생각해보자. 큐 시스템 내에는 입력을 임시로 저장할 버퍼를 가지고 있으며, 외부의 프로세서가 한가한 상태(idle state)에 있으면 저장되어 있는 항목을 전달하게 된다. 이러한 행위는 enqueue, dequeue 라는 추상화된 연산으로 모델링할 수 있다. enqueue와 dequeue 연산의 구현은 여러 가지 방식으로 이루어질 수 있다.

이들 두 연산에 대한 구현은 그 동작방식과 적용 알고리즘에 따라 구체적인 지식을 요구한다. 구체적으로 적용 알고리즘에 따라 선입선출(First In First Out), 후입선출(Last In First Out) 방식의 두 가지로 세분화될 수 있다.

통합 모델링 방식에서는 시뮬레이션 전문가가 이산 계층 시스템 계층에서 큐의 추상화된 상태 변화 다이어그램을 그리고, 큐에 대해 세부적으로 알고 있는 전문가가 enqueue, dequeue 연산에 대해 구체적으로 구현하게 된다.

파라미터 행위 모델링 환경에서는 시뮬레이션 모델에서 이러한 연산들을 구체적으로 정할 수 있게 한다. 그러므로 다수 개의 enqueue, dequeue 연산 중 특정 연산들이 선택되어지면 큐 시뮬레이터는 FIFO 또는 LIFO의 한 구체적인 방식으로 동작하게 된다. 또한 이러한 연산들 중 하나로 선택되어지는 과정은 동적 링크 라이브러리를 이용하여 실현되기 때문에 시뮬레이터가 로딩 시 또는 실행 시에 즉각적으로 이루어진다.

2. 동적 링크 라이브러리(DLL)

라이브러리는 대규모의 소프트웨어를 개발하거나, 또는 모듈의 재사용을 용이하게 하기 위해 만들어진 서브 프로그램들의 묶음이다. 라이브러리 안에는 코드와 데이터들이 포함되어 있으며, 이들은 링크되는 프로그램에서 참조될 수 있다. 라이브러리가 하나의 실행파일로 링크되는 방식은 크게 정적 링크(Static Link)와 동적 링크(Dynamic Link)의 두 가지로 구별된다. 정적 링크는 라이브러리 안에 들어있는 코드와 데이터들이 실행파일 내부로 복사되는 방식인데 비해 동적 링크는

라이브러리와 사용하는 서브 프로그램들의 위치정보만이 실행파일에 기록되고, 프로그램이 로딩 또는 실행되는 단계에서 메모리 공간으로 읽혀지는 방식이다. 본 연구에서는 마이크로소프트사의 DLL이라는 동적 링크 라이브러리를 이용하고 있다.

특히, 객체 모델링에서 각 연산들을 DLL 형식으로 구현하기 때문에 모든 파라미터별 연산들이 DLL로 구성되면 컴파일 과정 없이 즉시 시뮬레이터가 구성될 수 있게 된다.

3. 시뮬레이터의 자동 합성

통합 모델링 방법에서는 시뮬레이션 전문가가 시스템의 동작성을 상태 천이 다이어그램으로 나타내고, 필드 전문가가 특정 알고리즘들을 개별적인 함수들로 표현하게 된다. 시뮬레이션 전문가와 필드 전문가가 가장 많이 협력해야 하는 부분은 하위 레벨의 함수들에 대한 원형(Prototype)을 정의하는 일이다.

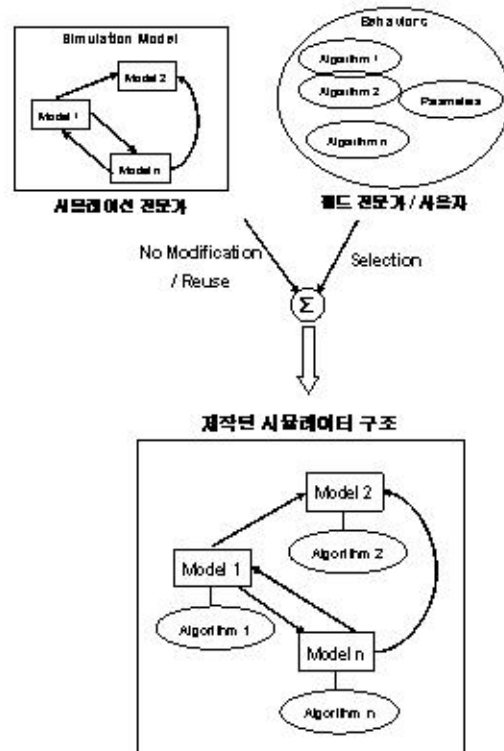


그림 2. 시뮬레이터의 자동 합성

상태 천이 다이어그램의 각 상태에서는 하위 레벨의 개별적인 함수를 호출하게 된다. 이 함수들은 필드 전문가에 의해서 DLL 형태의 동적 링크 라이브러리로 구현되게 된다.

최종적으로 시뮬레이터는 DLL 라이브러리 안에 구현된 개별적인 함수들 중 실제로 사용될 함수들을 선택함으로써 합성될 수 있다. 이러한 과정을 통해 시뮬레이터 사용자들은 다양한 종류의 알고리즘이나 규칙을 따르는 시뮬레이터를 사용할 수 있다. [그림 3]는 이러한 과정을 보여준다.

V. 통합 모델링을 이용한 전쟁 게임 시뮬레이터의 구축

파라미터 행위 모델링 방법을 이용하여 전쟁게임 시뮬레이터를 [그림 3]과 같이 구성할 수 있다. 이 시뮬레이터를 이용하면 군함의 방어장비인 SAM(Surface-to-Air Missile)과 EW(Electronic Warfare) 그리고, CIWS(Close-In Weapon System)의 방어력을 다수개의 ASM(Anti-Ship Missile) 공격으로부터 평가하여 군함의 생존가능성을 다양한 방어 전략 시나리오에 대하여 분석해 볼 수 있다.



그림 3. 해군 군함의 미사일 방어 전략

전쟁게임 시뮬레이터의 시나리오는 다음과 같다.

- 1) 군함이 특정 목적지로 이동하려고 항해중이다.
- 2) 이동 과정에서 군함은 외부의 적기 출현을 감지한다.
- 3) 하나 또는 여러 개의 ASM이 군함을 향해 주어진

시간 간격으로 발사된다. 군함 운영자들은 상대적인 위치 정보를 이용하여 미사일의 발사 지점을 찾아낼 수 있다.

4) 군함에는 세 종류의 방어 시스템이 갖추어져 있다: SAM (Surface-to-Air Missile), EW (Electronic Warfare), and CIWS (Close-In Weapon System). 각각의 시스템은 상호 다른 유효 범위를 가지고 있다. 군함은 미사일을 감지하는 순간 여러 방어 시스템을 이용하여 미사일을 격추시키려 하는 방어 전략을 세운다.

5) 만일 ASM이 군함을 맞추거나 또는 모든 ASM이 빔나가게 될 경우 시뮬레이션이 종료한다.

6) 옵션으로 헬리콥터가 군함을 호위할 수 있는데 이러한 경우에는 적기를 감지할 수 있는 반경이 증가하게 된다.

시뮬레이션 모델에 내포되는 세부 행위는 다음과 같은 항목들에 대한 것으로서 해군의 세부적인 지식을 알아야지만 구현할 수 있다.

- 1) 작전
 - 군함 헬리콥터, SAM, ASM
 - 군함의 공격 탈출 작전
- 2) 감지
 - 군함 헬리콥터의 감지 과정
- 3) 교전
 - 무기 배치
 - SAM-to-ASM
 - EW-to-ASM
 - CIWS-to-ASM
- 4) 피해량 분석
 - SAM-to-ASM
 - EW-to-ASM
 - CIWS-to-ASM
 - ASM-to-군함

각 세부 행위는 두 부분으로 구성된다. 하나는 시뮬레이션 시간의 제어와 관련된 부분이며, 또 다른 부분은 각 상태에서 의사결정과 계산을 하는 것이다. 시뮬레이션 시간 제어 부분은 이산사건 시스템으로 모델링되어 이루어지며 의사결정이나 세부 계산 과정에는 전

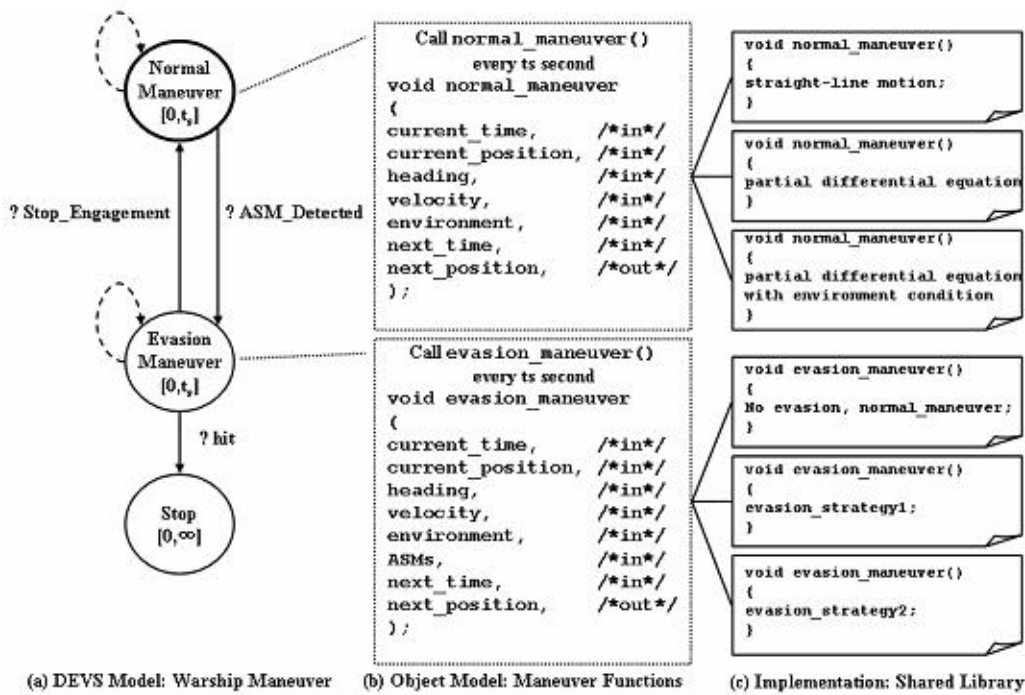


그림 4. 군함 전략의 통합 모델링

문적인 군사 지식이 요구된다. 그러므로 후자의 작업은 군사 전문가에 의해 정확히 구현되어야 할 수 있다.

예를 들어 군함을 이동하는 경우에는 시뮬레이션 시간을 전진시키면서 새로운 위치를 주기적으로 계산하게 된다. 또한 시뮬레이션 시간이 전진될 때마다 적기 감지도 매번 이루어진다. 교전 과정에서는 매우 정교하면서도 복잡한 계산과정을 거치게 된다. 이것은 각각의 무기 시스템이 서로 다른 제어 시간 스텝을 가지기 때문에 매우 많은 시뮬레이션 스텝을 거치면서 계산되어지기 때문이다.

[그림 4(a)]는 군함의 전략적인 행위를 보여주는 이산 사건 모델이다. 군함은 일반 기동(normal maneuver), 회피 기동(evasion maneuver), 정지(stop)의 세 가지 상태를 가진다. 일반 기동 상태에서는 공격을 받지 않고 어떤 한 목적지를 향해 움직이고 있는 상태이다. 만일 미사일 발사가 감지되면 ASM_Detected라는 이벤트가 발생하고 군함의 상태는 회피 기동으로 바뀌게 된다. 만일 미사일이 모든 방어시스템을 뚫고서 군함을 요격

하게 되면 명중(hit) 이벤트가 발생하고 군함의 상태는 정지 상태로 변하게 된다.

[그림 4(b)]에서는 군함의 다음 위치를 계산하게 되는 함수의 원형을 보여준다. 함수의 원형은 시뮬레이션 전문가와 필드 전문가가 협력하여 정의하여야 한다.

[그림 4(c)]는 군함이 이동할 경로를 결정하는 작전 함수들의 여러 가지 대안들을 보여준다. 군함의 이동은 다각선 경로를 따라 이루어질 수도 있으며, 미분 방정식의 해에 의해 그 위치가 결정되어 질 수도 있다. 만일 날씨나 지리학적 여건을 모두 고려한다면 미분방정식의 모델이 매우 복잡해 질 것이며, 위치정보도 매우 정확하게 계산될 수 있을 것이다. [그림 4(c)]는 세 가지 전략이 마련되어 있는 모델이다.

각각의 함수들은 별도의 DLL로 구현될 수 있으며, 실행되는 시뮬레이터는 이 중에서 관련된 라이브러리를 동적으로 링크하여 구성한다. 그러므로 실시간에 시뮬레이터를 재구성할 수 있으며, 시뮬레이션 결과 또한 시뮬레이터의 구성에 따라 새로이 도출될 수 있다. 이

러한 방식으로 군함에 탑재될 수 있는 다양한 전략과 알고리즘에 대해 매우 유연하고도 강력한 시뮬레이션을 수행할 수 있다.

또한 파라미터 행위 모델링 기법을 이용하였기 때문에, 다양한 군함의 전략과 알고리즘들을 시뮬레이션 전문가의 도움이 없이 해군 전략가가 하위레벨의 객체 모델링 부분에 반영하여 별도의 DLL로 구현함으로써 손쉽게 시뮬레이터의 구성을 변경하거나 확장할 수 있다.

예제 시뮬레이터는 마이크로소프트사의 Visual Studio를 이용하여 윈도우 XP 환경하에서 개발하였다. [그림 5]는 개발한 시뮬레이터의 화면을 보여주고 있다. 그림 안의 여러 무기 시스템을 표현하는 객체주위에는 텍스트와 버튼으로 구성된 파라미터들을 볼 수 있다. 사용자가 버튼을 눌러서 파라미터를 변경하면 DLL 파일내의 다른 전략 함수가 링크되어 전체 시뮬레이터가 구성 된다.

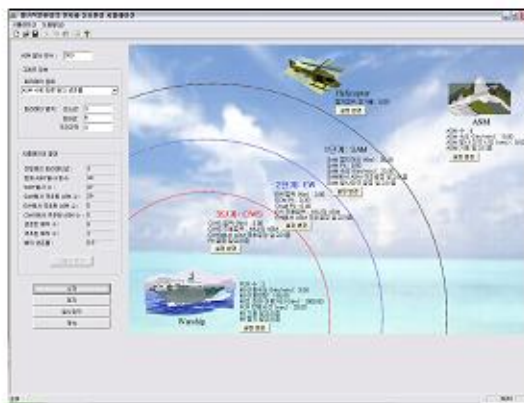


그림 5. 구현한 시뮬레이터의 실행 화면

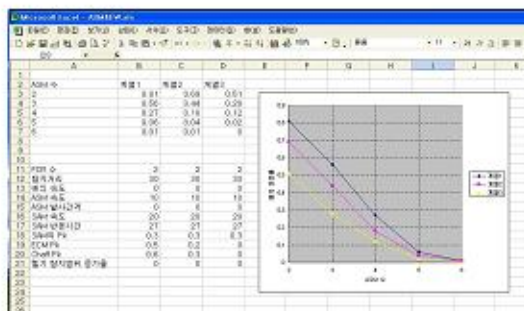


그림 6. 시뮬레이션 결과 분석 (Excel 연동)

시뮬레이션 결과는 [그림 6]에서 보이는 바와 같이 분석을 용이하도록 하기 위해 Excel을 이용하여 표와 차트로서 보여질 수 있다.

VI. 결론

본 논문에서는 파라미터 행위 모델링 기법을 이용한 전쟁게임 시뮬레이터의 구성방법에 대하여 제안하였다. 제안한 방법에서는 시뮬레이터의 구성이 매우 유연하며, 새로운 알고리즘이나 게임의 전략이 쉽게 적용될 수 있는 장점이 있다.

특히, 전쟁게임 시뮬레이터를 구성할 때에는 군 전략을 잘 알고 있는 전문가가 시뮬레이션 모델에 대해 잘 몰라서 기존 전쟁게임 시뮬레이터를 제작하기 곤란한 경우가 많았다. 본 논문에서 제안하는 방법에서는 전체 시뮬레이터에 탑재된 모델을 두 계층으로 구분하여 구성하고, 상위 레벨의 이산사건 시스템 계층은 시뮬레이션 전문가가 설계 및 구현하고, 하위 계층의 군 전략 시나리오가 들어있는 함수들은 군사 전문가가 DLL 파일 형태로 제작하도록 하여 시뮬레이션을 잘 모르는 사람도 쉽게 수정 및 확장할 수 있게 되었다.

제안한 방법을 이용하여 해군 군함의 방어력 평가가 효율적으로 이루어질 수 있음을 예제로서 확인할 수 있었으며, 다양한 방어 무기 시스템과 시나리오에 대하여 군함의 성능을 평가할 수 있었다.

파라미터 행위 모델링 기법은 전쟁게임 시뮬레이션 뿐만 아니라, 구성하는 요소가 많고 복잡한 대규모 시스템의 시뮬레이션에도 적합하다. 특히 시뮬레이션 도중 도출되는 의사결정에 따라 시뮬레이터가 실시간으로 재구성되어야 한다면 더욱 효과적으로 이용될 수 있다. 이러한 시뮬레이션의 예로써 비즈니스 프로세서 시뮬레이션, 환경 시뮬레이션 등이 있으며, 이러한 분야에 앞으로 많이 활용될 수 있다.

참고 문헌

[1] S. Y. Hong and T. G. Kim, "Embedding UML

Subset into Object-oriented DEVS Modeling Process," Proceedings of the Summer Computer Simulation Conference, San Jose, California, USA, pp.161-166, July 2004.

[2] C. H. Sung, S. Y. Hong, and T. G. Kim, "Layered Approach to Development of OO War Game Models Using DEVS Framework," Proceedings of the Summer Computer simulation Conference, 2005.

[3] J. K. Adams and D. E. Thomas, "The Design of Mixed hardware/software systems," Proceedings of the 33rd annual conference on Design automation, Las Vegas, Nevada, USA, pp.515-520, June 1996.

[4] D. W. Franke and M. K. Purvis, "Hardware/Software CoDesign: A Perspective," Proceedings of the 13th international conference on Software engineering, Austin, Texas, USA, pp.344-352, 1991.

[5] Booch, Rumbaugh, and Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesely, 1998.

[6] Rumbaugh, *Unified Modeling Language Reference Manual*, Addison-Wesely, 1999.

[7] M Priestley, *Practical Object-Oriented Design with UML*, The McGraw-Hill Companies, 1996.

[8] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation*, Academic Press, 2000.

[9] [http://en.wikipedia.org/wiki/Library_\(computer_science\)](http://en.wikipedia.org/wiki/Library_(computer_science))

저자 소개

이 완 북(Wan-Bok Lee)

총신회원



- 1993년 2월 : KAIST 전기및전자공학과 (공학사)
- 1995년 2월 : KAIST 전기및전자공학과 (공학석사)
- 2004년 2월 : KAIST 전자전산학과 (공학박사)

• 2005년 7월 ~ 현재 : 중부대학교 게임학과 교수
 <관심분야> : 시뮬레이션, 컴퓨터 게임, 정보보호, 이산 사건 시스템

김 재 현(Jae-Hyun Kim)

정회원



- 1998년 2월 : KAIST 전기및전자공학과 (공학사)
- 2000년 2월 : KAIST 전기및전자공학과 (공학석사)
- 2006년 2월 : KAIST 전자전산학과 (공학박사)

• 2006년 2월 ~ 현재 : KAIST 정보통신연구소 연구원
 <관심분야> : 이산사건/분산 시뮬레이션, HLA/RTI

김 탁 곤(Tag-Gon Kim)

정회원



- 1998년 : Univ. of Arizona 컴퓨터공학과 (공학박사)
- 1999년 ~ 현재 : 한국과학기술원 전자전산학과 교수
 <관심분야> : 이산사건 시스템, 모델링 및 시뮬레이션