

HLA 기반 대공유도무기 시뮬레이션 프레임워크 설계 및 사례적용

조 병 규[†] · 김 세 환^{**} · 윤 청^{***}

요 약

예측 불가능한 미래 전략 환경에 부합하고, 보다 경제적이며 과학적인 수단으로 무기체계개발을 수행하기 위해 모델링 및 시뮬레이션 기술의 적용이 확대되고 있다. 본 연구는 모델링 및 시뮬레이션 표준연동구조인 HLA(High Level Architecture) 및 TCP/IP를 지원하고, 실시간 분산 시뮬레이션을 지원하는 소프트웨어 프레임워크인 ADSF(Air Defense Simulation Framework)를 개발한 경험을 기술하고 있다. 개발된 ADSF는 대공유도무기 M-SAM(Medium Range Surface to Air Missile) 체계시뮬레이터에 적용하였으며, 위성항법장치(GPS: Global Positioning System) 시계를 사용하여 요구조건에 만족하는 시험결과를 산출하였다. 그 결과 ADSF는 HLA를 지원하고 실무기 체계의 통신 프로토콜인 TCP/IP를 지원하는 통신마들웨어가 만들어지고, 실시간 분산 시뮬레이션의 핵심 기술인 정밀모사를 지원하는 시뮬레이션 엔진도 갖추게 되었다. 대공유도무기 체계시뮬레이터는 다수의 구성요소로 이루어진 실시간 분산 시뮬레이션 시스템이므로 실시간 분산 시뮬레이션을 지원해 주는 시뮬레이션 엔진 개발이 요구되었다.

키워드 : 소프트웨어 프레임워크, 실시간 분산 시뮬레이션, 표준연동구조

A Design and Application of HLA-Based Air Defense Simulation Framework

Byung Gyu Cho[†] · Sae Hwan Kim^{**} · Cheong Youn^{***}

ABSTRACT

To correspond with the unpredictable future tactical environment, we expanded the application of M&S(Modeling & Simulation) that is more scientific and more economic in a field of weapon system development process. This paper describes experience in development of ADSF(Air Defense Simulation Framework) that supports not only HLA(High Level Architecture) which is an international standard in M&S but also TCP/IP as well as real-time distributed simulation. ADSF has been applied to the M-SAM(Medium Range Surface to Air Missile) System Simulator, and satisfying test results through GPS(Global Positioning System) timer has been acquired. As a result, an ADSF which is able to support HLA and TCP/IP as well as precise real-time simulation has been successfully made. We were in need of a real-time simulation engine to support Air Defense System Simulators that were consisted of several subsystems.

Key Words : Software Framework, Real-time Distributed Simulation, HLA

1. 서 론

오늘날 국방 분야에 국제적 화해모드로 인한 국방예산 감축에 따라 저 비용 고 효율 요구가 증대하고 있다. 또한, 안보환경은 미래 새로운 위협에 따른 군 구조와 무기체계의 재설계 요구 및 교육/훈련 여건의 변화와 획기적인 지원을 요구하는 쪽으로 변화하고 있다. 이와 더불어, 정보 통신관련 기술의 발전은 모델링 및 시뮬레이션(M&S: Modeling

& Simulation) 기반기술의 성숙을 가지고 왔으며, 경제적으로 실현 가능한 개발환경을 제공하고 있다. 또한, 선진국에서는 시뮬레이션 기반 체계 획득(SBA: Simulation Based Acquisition) 기술을 각종 체계개발에 적용하는 새로운 패러다임으로 접근하고 있다[1]. 국방과학연구소에서도 한국형 중거리지대공유도무기(M-SAM: Medium Range Surface to Air Missile)체계 및 기타무기체계개발에 모델링 및 시뮬레이션 기술을 적극적으로 적용하고 있는 실정이다[2].

그러나, 많은 국가들에서 다양한 목적에 의해서 개발된 시뮬레이션 모델들은 모델 간 상호운용성의 미흡 등으로 인해 투자대비 성과는 만족할 만한 수준이 되지 못하고 있었다. 이를 극복하기 위해, 미국 국방성에서는 시뮬레이션 상

[†] 정 회 원 : 국방과학연구소 선임연구원

^{**} 정 회 원 : Nex1 Future(주) 책임연구원

^{***} 정 회 원 : 충남대학교 컴퓨터학과 교수

논문접수 : 2005년 5월 18일, 심사완료 : 2005년 7월 18일

호운용성을 보장하기 위한 표준연동구조(HLA : High Level Architecture)를 제안하였으며, IEEE 1516 표준(Standard for Modeling & Simulation)으로 2000년에 등록되었다. 이런 노력은 미국의 국방 분야 모델링 & 시뮬레이션에 HLA를 준수하도록 하고 있으며, 한국에서도 창조 21 연동화 모델 등이 HLA를 준수하고 있다[1, 3].

국방과학연구소에서는 지대공유도무기 시험을 목적으로 지대공유도무기 체계시뮬레이터(이하 체계시뮬레이터)를 개발하고 있다. 체계시뮬레이터는 모델링 및 시뮬레이션 기술을 이용하여 생성한 가상환경(Synthetic Environment)에서 실무기 체계와 시뮬레이터를 연동한다. 즉, 체계시뮬레이터는 HLA를 준수하고, 실무기체계의 대표적인 통신 방식인 TCP/IP 및 UDP를 지원하도록 개발하고 있다. 현재, 체계시뮬레이터는 레이더 실무기 체계, 교전통제 실무기 체계, 수직발사 실무기 체계 및 다수의 분산 시뮬레이터들로 구성되어 있다. 또한, 시험평가에 적용하기 위하여서는 실시간으로 동작 및 연동되어야 한다[4, 5].

일반적으로 HLA 연동은 복잡한 개발기술을 습득해야 하며, 개발 시 오류가 쉽게 발생하였다[3-5]. 또한, 실시간 분산 시뮬레이션을 위해서는 체계시뮬레이터 구성요소(Sub System)간에 메시지 전송 지연시간(Data Latency)을 50 ms(통신주기 20Hz) 이하가 되도록 통신관련 서비스를 개발하여야 했다. 더욱이, 정밀 무기체계를 정확히 모사하기 위해서는 유도탄 및 표적의 운동방정식 모사(Runge-Kutta 방식사용)에 적절한 실행주기 100Hz 이상을 지원해 주는 실시간 시뮬레이션 엔진개발이 필요하였다. 이와 같은 요구조건을 만족시키기 위해서는 각 구성요소 개발자들이 HLA 연동 기술을 포함한 복잡한 통신부분과 실시간 시뮬레이션 엔진을 개발해야 하였다.

따라서, 본 논문에서는 HLA를 포함하는 다수의 통신환경을 지원하면서, 실시간 시뮬레이션 엔진을 포함하는 대공유도무기 시뮬레이션 프레임워크(ADSF : Air Defense Simulation Framework)를 제안하였다. 유용성 검증을 위하여 제안한 프레임워크는 M-SAM 체계시뮬레이터 개발에 적용하여 그 결과를 분석하였다. 또한, 기존연구와 다르게 결과분석의 정확성을 높이고자 위성항법장치(GPS : Global Positioning System) 시간을 사용하였다. 결과적으로 다수의 개발자가 ADSF를 이용하여 기존보다 편하게 국제표준과 실무기체계 연동이 가능한 실시간 분산 시뮬레이션 시스템을 구축할 수 있도록 하였다. 구축한 시뮬레이션 시스템은 미국 국방부 모의분석국(DMSO : Defense Modeling Simulation Office)부터 HLA 인증을 획득하였다[1].

본 논문은 2장에서 HLA, 시뮬레이션 프레임워크 기존연구들, 실시간 분산 시뮬레이션 개념 및 ADSF의 필요성에 대하여 설명하고, 3장에서는 제안한 ADSF 기술하고, 4장에서는 M-SAM 체계시뮬레이터 개발에 적용한 연구결과를 분석하고, 마지막으로 5장에서 결론 및 향후 과제를 제시한다.

2. 관련 연구

2.1 HLA 개념

시뮬레이션 표준연동구조인 HLA는 1990년대 중반 미국 국방성에서 모든 유형의 시뮬레이션, 무기체계 및 C4I체계간의 상호 운용성을 보장하고, 각 모형간의 재사용성을 향상시키기 위해 개발한 표준 분산 시뮬레이션 아키텍처이다[1, 3].

분산 시뮬레이션은 최초 개별 시뮬레이터를 활용한 장비 조작 숙달 훈련의 한계를 극복하기 위해 개발된 1세대 분산 시뮬레이션 구조인 SIMNET(Simulator Network) 체계로부터 다양한 시뮬레이션 모형과 시뮬레이터를 하나의 동일한 가상환경으로 연동하는 2세대 분산 시뮬레이션 구조인 DIS(Distributed Interactive Simulation)로 발전하였다. 동시에 전구급 수준의 대규모 합동시뮬레이션을 위한 분산 시뮬레이션 하부구조 및 통신규약을 정의한 ALSP(Aggregated Level Simulation Protocol) 체계로 각각 발전해 왔다[1, 3].

HLA는 규칙(Rules), 객체모델 모형(OMT : Object Model Template), 및 인터페이스 명세(Interface Specification)로 구체적으로 정의되었다. RTI(Runtime Infrastructure)는 HLA 인터페이스 명세를 구현하고 있고, HLA에 호환되는 시뮬레이션 응용체계를 개발하기 위해서는 반드시 사용해야 하는 미들웨어이다. 미국 정부는 국방 분야에서 시뮬레이션을 개발할 때에 HLA를 준수하도록 하고 있으며, 상용분야에서도 게임 시뮬레이션, 교육용 시뮬레이터 등에 HLA를 적용하려는 시도가 이루어지고 있다[6]. HLA 표준을 기반으로 개발된 개별 분산시뮬레이션 체계들은 상호작용하여 하나의 시뮬레이션 목적을 위해 한개의 페더레이션(Federation)을 구성하게 된다. 페더레이션은 분산 시뮬레이션 객체인 페더레이트(Federate)와 연동기반체계인 RTI, 및 교환 자료를 기술하고 있는 FOM(Federation Object Model)로 구성된다[1, 3].

2.2 소프트웨어 프레임워크

소프트웨어 프레임워크는 특정응용분야 소프트웨어 대해서 재사용할 수 있도록 설계된 상호작용하는 클래스(Class)들의 집합이라고 정의할 수 있다[7]. 또는, 특정 응용분야 소프트웨어를 개발하기위해서 구체화될 수 있는 어느 정도 완성(Semi-Completed)되고 재사용 가능한 프로그램이라고 정의할 수 있다[8]. 개발자는 프레임워크 클래스들을 부 클래스(SubClassing) 및 인스턴스(Instance) 과정을 통해 특정 응용분야에 적용할 수 있다.

소프트웨어 프레임워크는 하부체계의 상세한 것을 개념화하고, 증명된 상부체계 소프트웨어 서비스를 제공한다. 이러한 소프트웨어 프레임워크는 재사용이라는 가장 큰 이익을 제공한다. 일반적으로 프레임워크를 통한 재사용은 생산성 향상, 오류 교정, 개발기간 단축, 유지비용 감소 및 전체개발 비용감소를 제공한다[7].

2.3 기존 HLA 시뮬레이션 프레임워크 관련연구

Kevin Cox[7]은 여러 가지 HLA 페더레이트 지원하는

HFC(HLA Federation Class) 및 HAT(HLA Automation Tool)를 제안하였다. HFC는 시뮬레이션 시간을 관리해 주는 시뮬레이션 관리부분, 객체모델모형 표현 지원 부분, 원격 패더레이트 객체를 위한 프록시(Proxy) 부분, 및 객체속성의 자동 갱신과 반영 부분으로 구성된 C++ 라이브러이다. HAT는 객체모델을 이용 자동적으로 패더레이트 골격코드를 생성해 주는 도구이다. Jean-Pierre[8, 9]는 HLA 호환 패더레이트 개발 도구인 OSim 프레임워크를 제안하였다. OSim 프레임워크는 HLA 호환 패더레이트 자동개발을 지원하여 개발비용을 감소시켰다. Jay Graham [10]는 HLA 객체모델모형에서 자동적으로 Java 골격코드로 형성된 프록시 패더레이트를 생성해 주는 FedProxy 도구를 제안하였다. FedProxy 시뮬레이션 엔진은 시간 및 사건 관리 서비스, 객체관리 서비스, 및 시뮬레이션 실행제어 서비스를 제공한다.

Robert-Jan[11]는 DIS (Distributed Interactive Simulation) 호환 시뮬레이션을 HLA 호환 시뮬레이션으로 전환해 줄 수 있는 ASF(Advanced Simulation Framework)을 제안하였다. ASF는 소프트웨어 재사용성을 높이기 위하여 공통적으로 사용되는 시뮬레이션 기능들을 소프트웨어 프레임워크로 구성하였다. ASF는 시뮬레이션 전장 환경을 구성하는 Environment와 RTI에 기초하는 Object Server로 구성된다. ASF개발 결과 새로운 시뮬레이션 모듈을 개발할 때 재사용성 및 융통성을 높여 개발시간 단축 및 개발비용을 감소시켰다.

Eral[12]는 HLA와 DIS 호환형 분산시뮬레이션을 지원하는 프레임워크인 SIMWORK를 제안하였다. SIMWORK는 소프트웨어공학방법을 이용하여 Ada95 언어로 구현하였다. SIMWORK은 소프트웨어 개발자에게 시뮬레이션 객체를 HLA 객체모델모형으로 표현하도록 했고, HLA 패더레이트 행동을 캡슐화 하였다. Zijng Yuan[13]은 논리적 프로세스(LP: Logical Process)를 이용하여 일의 부담을 균형화 하는 프레임워크를 제안하였다. 김대석[3]은 RTI 프로그램과 시뮬레이션 프로그램을 분리해 주는 ROM(RTI Object Model) 프레임워크를 제안하였다. ROM은 RTI 서비스 부분, 객체상태 관리, 및 데이터 관리 부분으로 구성된다. ROM 프레임워크는 창조21 연동화 모델 개발에 성공적으로 활용되었다.

기존연구를 종합적으로 정리하면 HLA를 포함한 통신 관련부분과 시뮬레이션 엔진부분을 분리하여 소프트웨어 프레임워크로 구성하는 추세이다. 이전 연구에서 미비한 분야는 정교한 실시간 시뮬레이션 지원 시뮬레이션 엔진과 실시간 분산시뮬레이션을 지원하면서 신뢰성 있는 통신지연시간을 갖는 통신관련 프레임워크 부분이다.

2.4 실시간 분산 시뮬레이션 개념

실시간 시뮬레이션 실행에 있어서 시뮬레이션 시간(Simulation Time)과 일상 시간(WCT: Wall Clock Time)이 일정한 시간간격을 유지하면서 진행될 때 시뮬레이션은 실시간으로 진행된다고 할 수 있다[14]. 시뮬레이션 시간은 시

뮬레이션을 전진시키는 시뮬레이션의 내부시간을 말하며 일상시간은 실시간에 대한 참조시간으로서 일반적으로 시스템 시간을 나타낸다. 실시간 시뮬레이션에서는 시간제약조건이 중요하다. 시간제약조건은 일정한 시간간격 내에서 시뮬레이션의 계산처리가 완료되는 조건을 의미한다. 실시간 제약조건은 충족에 따라 하드(Hard) 실시간 시뮬레이션과 소프트(Soft) 실시간 시뮬레이션으로 구분된다. 하드 실시간 시뮬레이션은 시간제약조건이 엄격하게 지켜지지 않으면 임무수행 혹은 기능상에 중대한 오류가 발생할 수 있는 시뮬레이션(예를 들면, 시뮬레이터, 시험평가용 시뮬레이션 등)을 말하며, 소프트 실시간 시뮬레이션은 시간제약조건이 지켜지지 않을 때 현저한 성능저하를 가져올 수 있는 시뮬레이션을 말한다.

분산 컴퓨팅 환경에서 지역적으로 서로 다른 분산 노드에서 실시간 시뮬레이션 들이 상호 협조적으로 수행될 때 이를 실시간 분산 시뮬레이션이라고 한다. 실시간 분산 시뮬레이션에서는 실시간 제약조건을 만족해야 되는 것 외에도 각 분산노드에서 시뮬레이션 수행 간에 사건메시지 송수신시 발생할 수 있는 인과성오류(Casualty Error)가 발생하지 않도록 해야 한다[14, 15].

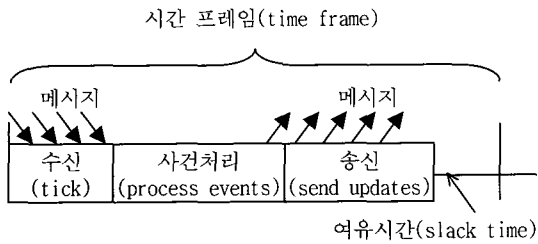
실시간 분산 시뮬레이션에서 주요 이슈에는 시간제약조건, 메시지 지연 등이 있다.

2.4.1 시간제약조건

실시간 분산 시뮬레이션의 수행에 있어서 임의의 시간간격(t) 내에서 이루어지는 작업에는 (그림 1)에서 보는 바와 같이 외부 메시지 수신(Tick), 메시지 관련 사건처리(Process Events), 그리고 외부 메시지 송신(Send Updates) 작업이 있다. 실시간 분산 시뮬레이션에서 t 내에서 이러한 연속적인 3가지 작업의 처리가 완료될 때, 시간제약조건이 만족되었다고 말한다[14].

2.4.2 메시지 지연

실시간 분산 시뮬레이션에서 메시지 지연(Message Delay)은 인과성 오류를 유발시키는 요인이 된다. 메시지 지연은 (그림 1)에서 외부 메시지 송신(Send Updates) 작업 구간에서 메시지를 송신한 시점으로부터 다른 시뮬레이션 노드에서 외부 메시지 수신(Tick) 작업구간에서 메시지를 수신한 시점까지의 시간차를 의미한다. 일반적으로 메시지 지연은 외부 메시지 송신시간, 네트워크 지연시간(Network Latency), 그리고 외부 메시지 수신시간에 의존적이다. 송신시간의 경우는 거의 무시할 수 있다. 네트워크 지연시간은 WAN(Wide Area Network)의 경우에는 무시하지 못할 정도이며 네트워크의 밴드 폭에도 크게 영향을 받는다. 외부 메시지 수신시간의 경우에는 수신 메커니즘을 어떤 것을 사용하느냐에 따라 달라진다. 특히, RTI와 같은 미들웨어를 통하여 수신되는 경우에는 많은 지연시간이 발생된다. 이런 지연시간을 가상지연시간(Virtual Latency)이라고 한다.



(그림 1) 시간 프레임

2.4.3 실시간 분산시뮬레이션 실행형태

실시간 분산 시뮬레이션 실행모델은 시간 전진을 어떻게 시키느냐에 따라 2가지로 구분한다. 시간간격(time-stepped) 실행모델과 시간샘플(time-sampled) 실행모델이 있다.

시간간격 실행모델에서는 일정한 시간간격(t)으로 시뮬레이션 시계를 전진시키면서 실행된다. 각 시간간격 내에서는 3가지 작업이 수행된다. 외부 메시지 수신, 메시지 관련 사건처리, 그리고 외부 메시지 송신 작업이다. 이 작업수행의 사이클이 끝나면 내부 시간 즉, 시뮬레이션 시계를 t 만큼 전진시킨다.

시간샘플 실행모델은 외부 메시지 수신, 메시지 관련 사건처리, 그리고 외부 메시지 송신 작업 수행 사이클이 끝나는 시점에 시뮬레이션 시계를 전진시키는 방법이다. 이는 가능한 빠른 시간에 사건 완료시점을 알려주도록 하는 방식이다. 여유시간에 대한 요구가 없이 바로 일상시간(WCT)과 동기화되도록 허용하는 것이다. 이는 설계상에 있어서 어떤 입력에 대하여 체계의 응답시간이 아주 중요한 요소가 될 때 적합하다[14].

2.5 ADSF 필요성

체계시뮬레이터는 HLA를 이용하는 군사용 시뮬레이터와 TCP/IP 및 UDP 통신을 이용하는 실 무기체계를 연결하여 동작하는 시스템이다. 따라서, HLA 및 TCP/IP, UDP 통신을 모두 지원하는 통신미들웨어가 필요하다.

체계시뮬레이터는 다수의 구성요소로 구성되어 있다. 이 구성요소들은 서로 네트워크로 연결되어 하나의 대공유도무기 패더레이션(Air Defense System Federation)을 이루게 된다. 각각의 부 구성요소들은 시뮬레이션 수행을 위해서 시간간격 실행모델을 지원하는 시뮬레이션 엔진이 필요하다.

개발자들이 통신미들웨어와 실시간 분산 시뮬레이션 엔진을 각각 개발하기 보다는 공통적인 프레임워크를 적용하여 재사용을 높일 필요성이 있다. 기존의 HLA 시뮬레이션 프레임워크들은 실 무기체계 포함을 고려하지 않았다. 또한, 체계시뮬레이터에 필요한 정밀한 모사주기인 100Hz를 지원하지 못하므로, 100Hz 이상을 지원하는 시뮬레이션 엔진 개발이 요구된다. 실시간 분산시뮬레이션을 지원하기 위해서 체계시뮬레이터 메시지 지연시간을 50ms이내 설계하여야 한다.

따라서, 체계시뮬레이터의 실시간 분산 시뮬레이션 요구조건을 만족시키는 통신부분 및 시뮬레이션 엔진을 포함하

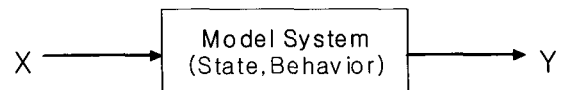
는 소프트웨어 프레임워크 적용이 필요하다.

3. ADSF(Air Defense Simulation Framework)

대공유도무기 각 패더레이터에서 사용되는 공통적인 기능들을 지원하는 것이 ADSF의 개발목적이다. ADSF는 큰 기능인 통신기능을 담당하는 미들웨어 부분과 실시간 시뮬레이션 엔진을 담당하는 모델구조(Model Architecture) 부분으로 구성된다.

3.1 모델구조

제한한 모델구조는 이산 모델시스템을 기반으로 개발하였다. (그림 2)의 상태(State)는 이산 모델시스템에서 정의한 상태변수(State Variables)에 해당된다. 행위(Behavior)는 이산 모델시스템 내부의 행동을 모사하는 사건을 수행하는 내부함수를 말한다.

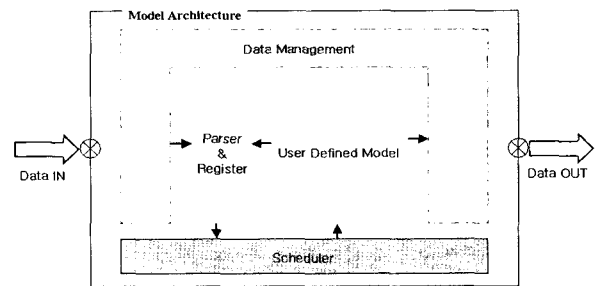


(그림 2) 모델구조에서의 모델 시스템

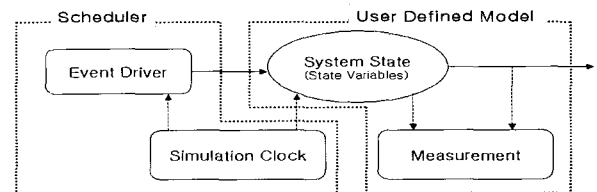
3.1.1 모델구조 구성

모델구조의 큰 구성은 (그림 3)과 같이 데이터관리(Data Management), 분배/등록(Parser/Register), 및 스케줄러(Scheduler)로 구성된다. 데이터는 모델의 인터페이스(Interface)를 통해서 들어온 후 사용자 정의모델로 전달된다. 전달된 데이터는 사용자 정의모델에서 처리하여 데이터인터페이스를 통하여 외부로 출력된다.

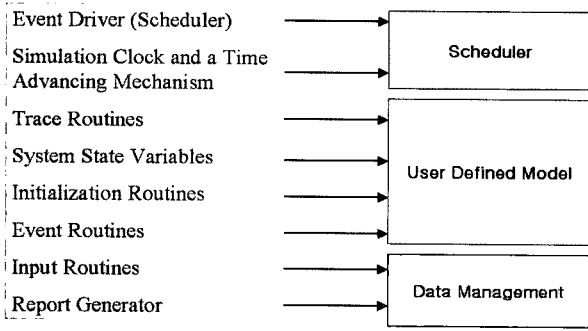
(그림 4)는 기본적인 이산-사건 시뮬레이션 모델이 구성 요소에 제안하는 모델 구조의 기능을 나타낸다.



(그림 3) 모델구조 구성



(그림 4) 기능분해도



(그림 5) 기능 분류도

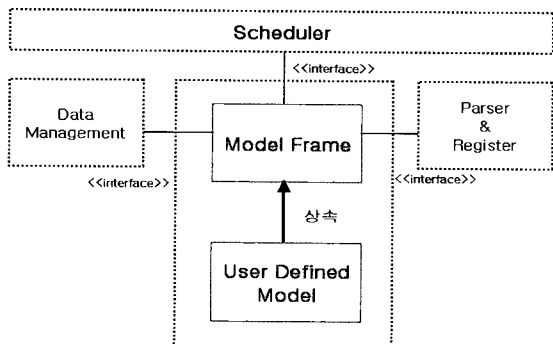
이산-사건 시뮬레이션의 주요 구성요소와 모델 구조의 상호 관계는 (그림 5)와 같다.

3.1.2 모델구조 기능

사용자 정의 모델(User Defined Model)을 통해 모델링하려는 모델은 각각 자신의 상태와 행위를 가지고 외부 입력이나 내부 연산을 통해 상태를 변화시키거나 행위를 수행한다. 따라서, 사용자 정의 모델은 외부 인터페이스 부분과 모델이 공통적으로 가지는 기능을 모아놓은 프레임이다. 나머지 부분은 이 프레임을 기반으로 모델을 정의하는 사용자가 작성하여야 한다. (그림 6)과 같이 사용자가 정의하는 모델은 기본적으로 모델프레임(Model Frame)의 기능을 상속하여 사용한다.

데이터 관리 부분은 모델프레임에서 기본적으로 제공하는 데이터구조(Data Structure)들을 정의하고, 그 데이터들을 미들웨어인 네트워크가 전송/수신할 수 있게 포장(Packing)/정리(Unpacking)한다. (그림 7)은 미들웨어로부터 데이터가 입력되어 사용자정의 모델로 전달되는 구조를 나타낸다. 데이터관리에 대한 세부 기능에 대한 설명은 <표 2>와 같다.

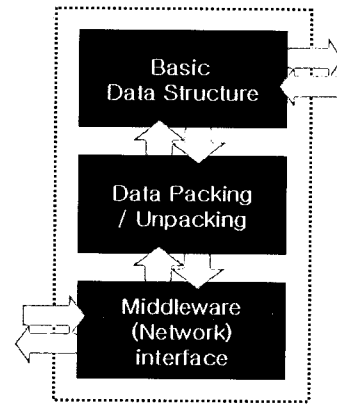
분배기는 외부 시스템을 통해 들어오는 데이터를 데이터 관리 모듈을 이용하여 가져와 그 데이터를 분석하여 사용자 모델을 구동하는 스케줄러로 호출함수(Callback Function)를 등록한다. 등록할 때는 등록기를 이용한다. 마찬가지로 외부 데이터가 등록취소를 요구하는 것으로 분배기가 분석한 경우도 등록기를 이용하여 등록취소를 하게 된다. 내부모델에서 주기적인 호출과 비 주기적 호출을 스케줄러에 등록하거나



(그림 6) 사용자 정의 모델

<표 2> 사용자 정의 모델 세부 기능

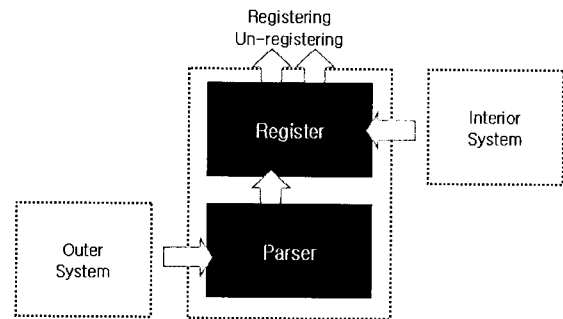
세부기능	설명
기본적인 Data 구조	Data Management에서 제공하는 Data 구조를 이용하여 제공
CallBack 등록	등록/분배기를 이용하여 제공
CallBack 등록제거	등록/분배기를 이용하여 제공
Data Output 제공	Data Management를 이용하여 제공
Data 초기화	모델 Frame에서 기본기능 구현



(그림 7) 데이터관리

<표 3> 데이터관리 세부기능

세부기능	설명
기본 자료구조 제공	사용자정의모델이 사용할 수 있는 기본적인 자료구조 제공
자료 포장/정리 제공	미들웨어를 통해 송수신되는 자료의 포장/정리 제공
미들웨어 인터페이스 제공	미들웨어를 통한 자료 송수신 인터페이스 제공



(그림 8) 등록/분배기 구조

취소하게 되는 경우는 등록기를 이용하여 직접수행하게 된다. 단, 공통적인 모델프레임이 이를 수행할 수 있는 인터페이스를 제공한다. (그림 8)은 분배/등록기의 구조를 나타낸다.

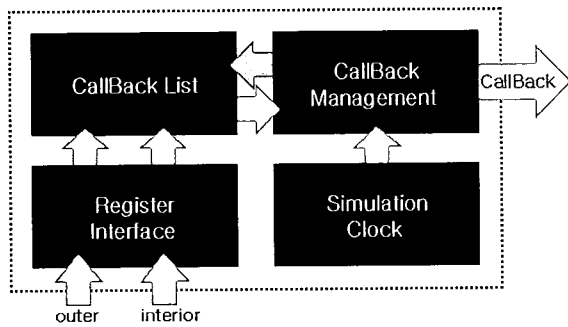
스케줄러의 기본적인 기능은 모델의 시뮬레이션 시간 진행에 따라 사용자 정의 모델 상태에 영향을 주는 사건을 발생시키는 것이다. 여기서 사건은 사용자가 정의한 행위를 시뮬

레이션 시간 흐름에 따라 호출해주는 것으로 정의하였고, 스케줄러는 외부와 모델에서 요구하는 호출함수를 등록기 인터페이스를 통해 등록시키거나 취소시킨다. 그것을 시뮬레이션 시계(Simulation Clock)를 참조하여 사용자 정의 모델의 행위를 호출한다. 등록된 행위 리스트는 기본적으로 시뮬레이션 시간 흐름에 따라 정렬된다. 각 호출마다 우선순위를 결정하여 호출관리가 호출을 할 것인지를 결정하게 된다. (그림 9)는 사용자 정의 모델로부터 사건등록 과정과 등록된 사건에 대해서 사용자 정의모델의 행위가 호출되는 과정을 나타낸다. 스케줄러의 세부기능에 대한 설명은 <표 4>와 같다.

ADSF 모델 스케줄러는 정밀한 실행모사 주기를 수행하고자 GPS Board Timer 또는 Multimedia Timer을 사용한다.

<표 4> 등록/분배기 세부기능

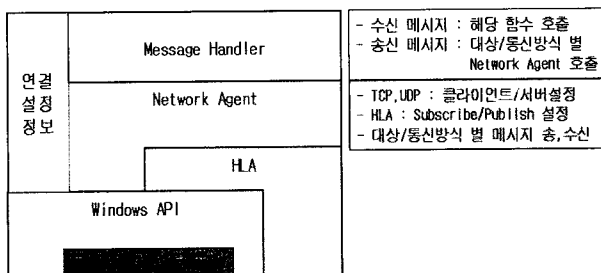
세부기능	설명
Parsing(분석)	데이터관리를 통해 들어온 외부 시스템의 자료를 분석하여 어떤 자료인지 판단
Callback 등록	스케줄러 Callback등록
Callback 등록제거	스케줄러 Callback 등록제거



(그림 9) 스케줄러 구조

<표 5> 스케줄러 세부기능

세부기능	설명
Callback List	등록된 Callback들이 시간과 우선순위를 바탕으로 정렬되어 있는 자료 구조집합
Callback 관리	Callback를 Simulation Clock를 바탕으로 하나씩 꺼내서 사용자 정의모델을 실제로 호출
등록기 인터페이스 제공	등록기를 통해 등록 또는 취소 요구에 대한 인터페이스



(그림 10) 미들웨어 구조

3.2 미들웨어

체계시뮬레이터는 분산 시스템으로 실 체계와 시뮬레이터를 통합하여 실체계의 성능을 분석하는 목적으로 설계되어 있다. 실 체계 장비의 경우 TCP/IP나 UDP를 이용하며 시뮬레이터 연동을 위하여 별도의 HLA 인터페이스를 제공하지 않으므로, TCP/UDP와 HLA를 모두 지원하는 미들웨어를 작성하여 일관된 품질의 네트워크 서비스를 제공해야 한다.

ADSF 미들웨어의 구조는 (그림 10)과 같다.

미들웨어는 미 국방성 4계층의 정의에 의하여 프로그램 계층에 속하는 모듈로 다음과 같은 기능을 제공한다.

- ① TCP/IP 통신의 자동설정
 - 가. 서버 및 클라이언트 생성
 - 나. TCP/UDP 메시지 전달
 - 다. 메시지 전송 및 입력 메시지 감시/전달
- ② HLA 통신의 자동설정
 - 가. Publish 및 Subscribe
 - 나. HLA 메시지 전달
 - 다. 메시지 전송 및 입력 메시지/전달

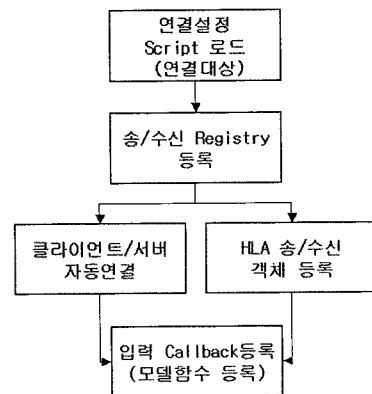
HLA에서 Publish 라는 것은 정보를 전송한다는 의미이고, Subscribe는 정보를 수신한다는 의미이다.

제안된 미들웨어가 HLA 와 TCP/IP를 이용하여 통신을 하기 위한 기본 설정은 (그림 11)과 같다.

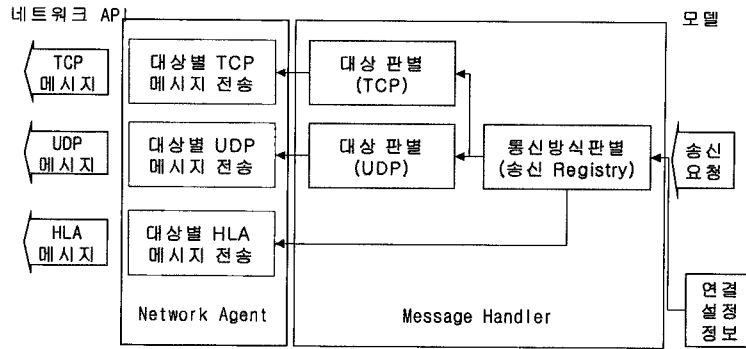
사용자는 스크립트나 미들웨어 제공 API(Application Program Interface)에 의하여 클라이언트-서버 및 HLA 송/수신 객체에 대한 등록을 수행할 수 있으며 해당정보가 들어왔을 때 호출되는 사용자 함수를 지정 할 수 있다.

미들웨어를 이용한 데이터 전송은 (그림 12)와 같다. 데이터를 전송하기 위해서는 사용자가 스크립트에 정보를 기입한 순서나 해당 연결의 이름을 이용하여 미들웨어에 전송을 요청한다. 미들웨어는 연결 설정정보를 이용하여 사전에 정의된 방식에 의하여 데이터를 전송한다.

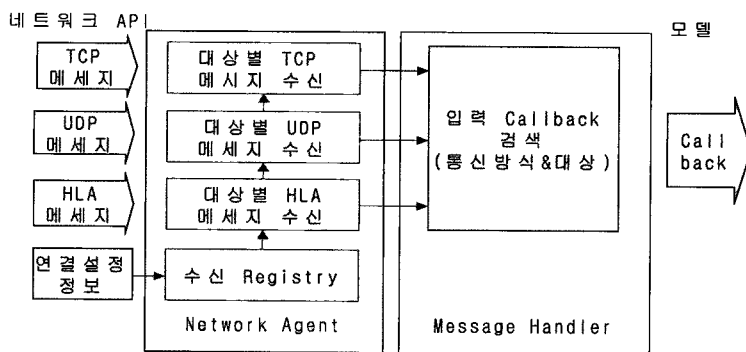
미들웨어를 이용한 데이터 수신은 (그림 13)과 같다. 수신된 데이터는 프로토콜에 따라 각 다른 방식으로 사용자 지정 한 호출함수를 호출하는 방법에 의해 사용자에게 네트워크를 통해 입력된 정보가 전달된다.



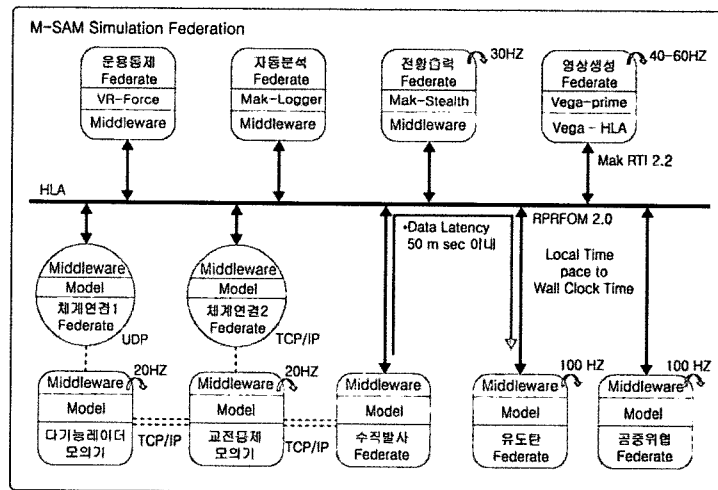
(그림 11) 미들웨어 통신 설정



(그림 12) 미들웨어 이용 정보 전송



(그림 13) 미들웨어 이용 정보 수신



(그림 14) M-SAM 시뮬레이션 패더레이션

4. ADSF 적용사례

ADSF를 사용하여 M-SAM 체계시뮬레이터를 작은 자원 (인력, 예산, 기간)을 사용하여 실시간 분산시뮬레이션을 위한 요구조건을 만족하게 설계하고 구현하였다. 요구조건은 선진국의 유사 분산시뮬레이션 시스템과 통신지연시간은 동일하고, 작업수행간격은 M-SAM 체계시뮬레이터가 더 어려운 조건이다[16]. M-SAM 체계시뮬레이터는 모델링 및 시뮬레이션 기술을 이용하여 체계 시험평가에 사용할 시험평가 도구이다. 2004년 탐색개발 시제품을 개발 완료하였다.

4.1 개발환경

M-SAM 체계시뮬레이터 모델은 Mak RTI 2.1을 사용하며, Window 2000 운영체제에서 C++ 언어로 구성하였다. 전체 패더레이션 구조는 (그림 14)와 같다.

상용제품을 사용하는 영상생성 장치를 제외한 모든 구성 요소에 ADSF를 적용하여 성공적으로 개발 하였다.

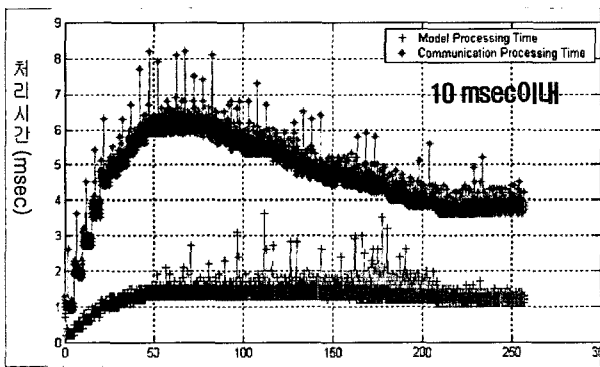
4.2 시험 및 결과

일반적인 시뮬레이션 소프트웨어 프레임워크 성능 측정을 위해서는 통신 지연시간 및 시뮬레이션 엔진 측면에서 본

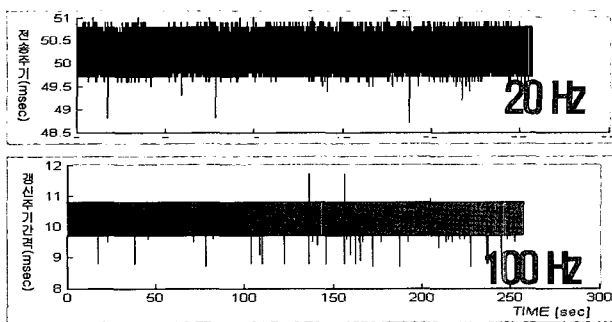
갱신주기를 측정하여야 한다[8, 9].

본 연구에서는 다음과 같이 3가지 항목에 대해서 시험하였다. 또한, 기존 성능측정 논문에서 사용한 시간 동기화 방법인 NTP(Network Time Protocol)와 시간측정에 사용된 윈도우 타이머는 정확성이 1ms~30ms의 오차를 가지고 있었다[6]. 기존방법은 ADSF의 성능측정에 부적합하다. 따라서, 본 논문에서는 1μs 이상의 정밀도를 갖는 bc637PCI GPS Board타이머를 사용하였다[17].

일반적으로 논리적으로 검증하기 곤란한 경우 분산 시뮬레이션의 실시간성을 검증하기 위해서는 시스템이 갖는 최악의 조건을 생성한 후 시간적 요구조건을 만족하는지를 보고 판단한다. 본 연구의 시험 결과들은 M-SAM 체계시뮬레이터의 최악의 조건(worst case)인 최대다수의 적 공중위협을 모의했을 때이다. M-SAM 체계시뮬레이터는 적 공중위협을 최소 1대부터 최대 00대 까지 동시에 모의할 수 있다. 1대의 경우 컴퓨팅 부하도 최소이고 통신 메시지양도 적어서 최적의 조건이라고 볼 수 있다. 일반적으로, 최대의 경우 컴퓨팅 부하 및 통신 메시지 양이 최대로 된다. 따라서, 최대 다수의 공중위협을 모의했을 때를 처리시간 및 메시지 지연시간이 최악의 조건이라고 볼 수 있다. 체계시뮬레이터 요구조건에 따라 공중위협모의 패더레이트는 다수의 적 공중위협에 대하여 100Hz (처리시간 10ms 이내)로 처리하여야 한다. ADSF 적용결과 (그림 15)와 같은 요구조건에 만족한 결과를 얻었다.



(그림 15) 공중위협 패더레이트 처리시간



(그림 16) 갱신 주기 및 전송주기 정확성

4.2.1 모의 정보 갱신 주기 정확성

적 공중위협 정보는 시뮬레이션 수행을 위해서 100 Hz 단위로 갱신되어야 한다. 갱신 주기 정확성에 대한 시험 결과는 (그림 16)과 같다.

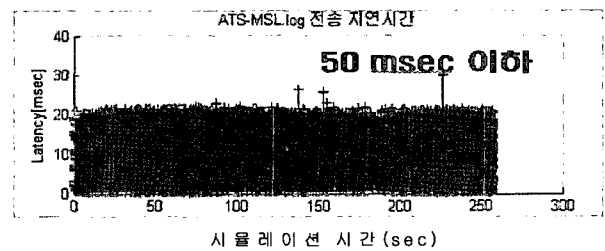
4.2.2 데이터 전송 주기 정확성

적 공중위협 정보는 시뮬레이션 수행을 위해서 주기적으로 이 정보를 필요로 하는 패더레이트에 전송하여야 한다. M-SAM 체계시뮬레이터의 설계요구조건은 20Hz 단위로 전송해야 한다. 데이터 전송 주기 정확성에 대한 시험결과는 (그림 16)과 같이 표현되어 있다.

4.2.3 주기적 데이터 통신 지연시간

실시간 분산 시뮬레이션 수행을 위해서는 필요한 정보들이 네트워크를 통해서 전송/수신 되어야 한다. 적 공중위협에 대한 정보는 (그림 16)과 같이 주기적 전송되었다. HLA를 준수하려면 RTI를 이용하여 통신을 수행해야 한다. 하지만, RTI는 내부적으로 처리하는 일 때문에 많은 계산 부하가 발생하여 통신 지연시간에 영향을 미친다. 따라서, RTI 성능 분석을 위한 많은 연구들이 진행되어 오고 있다[6, 18, 19].

M-SAM 체계시뮬레이터 공중위협 패더레이트에서 유도한 패더레이트까지 메시지 지연시간 요구조건은 50 msec 이하이다. ADSF를 적용하여 시험한 결과 (그림 17)과 같이 만족한 결과를 얻었다. 시험결과를 자세히 살펴보면 평균적인 메시지 지연시간은 20msec 정도이고 최악의 경우에도 요구조건인 50msec 이내로 만족하는 결과를 얻었다.



(그림 17) 메시지 전송 지연시간

4.3 ADSF 적용 M-SAM 체계시뮬레이터 개발 효과

M-SAM 체계시뮬레이터는 HLA를 기반으로 국내에서는 최초로 시도되는 실무기 체계와 시뮬레이터를 연동하는 시스템이다. 즉, Live 시뮬레이션과 Virtual 시뮬레이션을 연동하는 시스템이다. 기존의 나와 있는 국내 HLA 연동시스템들은 위게임과 같은 구조적(Constructive) 시뮬레이션이었다. M-SAM 체계시뮬레이터개발에 ADSF를 개발하여 적용함으로써 일반적인 모델개발자가 복잡한 HLA에 개발기술에 대해서 익숙하지 않아도 된다는 장점을 제공하였다. 또한, 기존에 검증된 HLA 적용기술을 소프트웨어 프레임형태로 적용함으로써, 검증 및 확인에도 많은 도움을 제공하였다. 개발된 ADSF를 지대공유도무기 사업 모델링 및 시뮬레이션 분야에 확대 적용할 수 있는 기회를 가지게 되었다.

5. 결론

대공유도무기 체계시뮬레이터 개발을 위해서 모델링 및 시뮬레이션 표준인 HLA를 적용 했으며, 실무기 체계와 연동을 위해서 TCP/UDP 표준도 지원되어야 한다. 체계시뮬레이터는 다수의 구성요소로 이루어진 실시간 분산 시뮬레이션 시스템이므로 실시간 분산 시뮬레이션을 지원해 주는 시뮬레이션 엔진 개발이 요구되었다. 따라서 본 연구에서는 HLA 및 TCP/UDP를 지원하고, 실시간 분산 시뮬레이션을 지원하는 소프트웨어 프레임워크인 ADSF를 개발하였다. 개발된 ADSF는 M-SAM 체계시뮬레이터에 성공적으로 적용하였다. 또한, 기존연구보다 더 정확한 시험장비로 시험하여 요구조건에 만족하는 시험결과를 산출하였다.

결론적으로 ADSF는 모델링 및 시뮬레이션의 표준인 HLA를 지원하고 실무기 체계의 통신 프로토콜인 TCP/UDP를 지원하는 통신 미들웨어를 갖추게 되었고, 실시간 분산 시뮬레이션의 핵심인 시뮬레이션 엔진도 갖추게 되었다. 또한, ADSF는 앞으로 개발된 다른 M-SAM 모델링 및 시뮬레이션 분야에 적극적으로 적용할 수 있을 것이다. 또한, 국방과학연구소 유도무기체계 모델링 시뮬레이션 분야에도 적용 가능할 것이다.

연구 결과로 제시된 ADSF는 윈도우 2000, XP 환경에서 지원되고 있으므로 광범위 한 적용을 위해서는 다른 운영체제를 지원하는 프레임워크로 발전해야 할 것이다.

참 고 문 헌

- [1] Defense Modeling and Simulation Office, <http://www.dmsomil>
- [2] 조병규, 김진호, 이승만, "M-SAM 체계개발에서의 M&S 기술적용", 제2회 무기체계 모델링 및 시뮬레이션 발표대회, pp.C1-C17, 2003.
- [3] 김대석, 배종환, 류재철, "HLA 패더레이터 개발을 위한 ROM 프레임워크 설계 및 구현", 정보처리학회논문지D, Vol. 9-D, No.6, pp.1137-1144, 2002.
- [4] 조병규, 이승영, 유양선, 김세환, "M-SAM 체계시뮬레이터 모델구조에 관한 연구", 제12차 유도무기 학술대회 논문집, pp.133-136, 2003
- [5] 조병규, 정석형, 김세환, "M-SAM 체계시뮬레이터 분산 통신에 관한 연구", 제 12차 유도무기 학술대회 논문집, pp.130-132, 2003.
- [6] 최상영, 조병규, 이길섭, "HLA 기반 실시간 분산 M-SAM 시뮬레이션에서 RTI 성능 측정 및 분석", 정보과학회 논문집: 커뮤니티의 실제, Vol.11 No.2, pp.149-156, 2005.
- [7] Kevin Cox, "A Framework-based Approach to HLA Federate Development", Simulation Interoperability Workshop, 1998, Fall
- [8] Jean-Pierre Belanger, Philippe Fortier & Lily Lam, "Lessons Learned from a framework approach to HLA-Compliant Federation Development", Simulation Interoperability Workshop, 1998, Fall
- [9] Jean-Pierre Belanger, Philippe Fortier & Lily Lam, "OSim Framework. Experience of an In-flight Refueling Federation Development Using Commercial Tools", Simulation Interoperability Workshop, 1997, Fall
- [10] Jay Graham, Jim Foscue, Dannie Cutts, "HLA Object Models as Software Object Models", Simulation Interoperability Workshop, 1998, Spring
- [11] Robert-Jan Elias, Wim Huiskamp, "Advanced Simulation Framework : A Generic Approach to Distribution Simulation", Proceeding of the 8th International Training and Education Conference(ITEC '97), 1997, April
- [12] Capt. Earl C. Pilloud, Maj. Mark A. Kanko, "SIMWOR K : An Ada 95 distributed simulation application framework support HLA and DIS", pp.732-739, IEEE, 1997.
- [13] Zijing YUAN, Wentong CAI, Malcolm Yoke Hean LOW, "A framework for executing parallel simulation using RTI", Proceedings of the Seventh IEEE international symposium on Distributed Simulation and Real-Time applications, 2003.
- [14] Thom McLean, "Hard Real-Time Simulation using HLA", Simulation Interoperability Workshop, 2003, Fall.
- [15] Herbert Tietje, "Benchmarking of RTIs for Real-Time Applications", Simulation Interoperability Workshop, 2003, Euro
- [16] Robert S. Manthy, "Navy Theater Wide Hardware-in-the-Loop End-to-End Simulation using Raytheon's HLA based Standard Missile-3 System Testbed", Simulation Interoperability Workshop, 1999, Fall
- [17] Symmetricom bc637 PCI GPS synchronized PCI Time & Frequency User's Manual
- [18] Pamela Knight, Ron Liedel, Melanie Klinner, "WBT RTI Independent Benchmark Tests : Design, Implementation, and Updates Results", Simulation Interoperability Workshop, 2002, Spring.
- [19] Paul Kuiper, Arjan Lemmers, Rene Verhage, "Performance Measurements of a HLA Component-based Fighter Pilot Station", Simulation Interoperability Workshop, 2002, Euro



조 병 규

e-mail : bgcho@naver.com

1988년 인하대학교 전산과(이학사)

1990년 인하대학교 전산과(이학석사)

1990년~현재 국방과학연구소 선임연구원

2003년~현재 충남대 컴퓨터학과

박사과정

관심분야 : 모델링 & 시뮬레이션, 소프트웨어 공학



김 세 환

e-mail : shkimc@nex1.co.kr
1985년 경북대학교 전자과(공학사)
1987년 경북대학교 전자과(공학석사)
1987년~현재 Nex1 Future(주) 시스템
연구소 책임연구원 M&S 담당
관심분야 : 모델링 & 시뮬레이션, CMMI,
HLA



윤 청

e-mail : cyoun@cs.cnu.ac.kr
1979년 서울대학교 물리과(이학사)
1983년 일리노이 주립대학교 Computer
Science(석사)
1988년 Northwestern 대학교 Computer
Science(박사)
1988년~1993년 Bell Communications Research 선임연구원
(MTS)
1993년~현재 충남대학교 컴퓨터과학과 교수
관심분야 : 소프트웨어 공학, 소프트웨어 형상관리, 객체지향 모델링