

석사학위논문

HLA/RTI기반 Test bed 구현의 실증적 연구

-분산 시뮬레이션 개발 및 구현을 중심으로-

2014년

한성대학교 국방과학대학원

국방 M&S학과

국방M&S학전공

박 희 준

석사학위논문
지도교수 이동준

HLA/RTI 기반 Test bed 구현의 실증적 연구

-분산 시뮬레이션 개발 및 구현을 중심으로-

An Empirical Study for the Implementation of Test bed
based on HLA/RTI: Focused on Distributed Simulation
Development and Implementation

2013년 12월 일

한성대학교 국방대학원

국방 M & S 학과

국방 M & S 학전공

박 희 준

석사학위논문
지도교수 이동준

HLA/RTI 기반 Test bed 구현의 실증적 연구

-분산 시뮬레이션 개발 및 구현을 중심으로-

An Empirical Study for the Implementation of Test bed
based on HLA/RTI: Focused on Distributed Simulation
Development and Implementation

위 논문을 국방M&S학 석사학위 논문으로 제출함

2013년 12월 일

한성대학교 국방과학대학원

국방 M & S 학과

국방M&S학전공

박 희 준

박희준의 국방M&S학 석사학위논문을 인준함

2013년 12월 일

심사위원장 _____ 인

심 사 위 원 _____ 인

심 사 위 원 _____ 인

국 문 초 록

HLA/RTI 기반 Test bed 구현의 실증적 연구

- 분산 시뮬레이션 개발 및 구현을 중심으로 -

한성대학교 국방과학대학원

국방 M&S 학과

국방 M&S 전공

박 희 준

미 국방성 표준시뮬레이션 기술구조인 HLA/RTI는 전 세계적으로 국방 M&S에 사용되는 분산 시뮬레이션 표준기술구조이다. 우리나라 국방M&S에서도 역시 HLA/RTI를 시뮬레이션 간 연동기술을 많이 채택하고 있는 실정이다.

RTI는 분산 시뮬레이션의 구현 및 실행을 위한 다수의 서비스를 사용자에게 제공하나 그만큼 시스템 리소스를 사용하고, 메시지 전송에 걸리는 부하가 강해지는 만큼 데이터 송수신 퍼포먼스가 크게 하락한다. 따라서 개발과정의 위험을 줄이기 위해서는 시뮬레이션 구현 전 설계단계부터 시뮬레이션 망에서 사용될 데이터를 결정하고 시스템이 수용 가능한 전송량을 예측한 후 구현에 들어가야 한다.

본 논문에서는 HLA/RTI 기반 분산시뮬레이션 구성 시 고려해야 할 요소를 식별하고 실제 분산시뮬레이션 M&S도구 구현사례 및 개발과정에 확인된 이슈들에 대해서 고찰하였다. 그리고 시뮬레이션의 적정 설계방안을 찾기 위한 HLA/RTI 기반 Test bed 구성을 제안하고 이를 적용한 시험을 수행하였다.

【주요어】 국방M&S, HLA, RTI, Test bed, 모델링, 시뮬레이션

목 차

I. 서 론	1
1.1 연구의 배경과 목적	1
1.2 연구의 범위와 방법	2
II. 시뮬레이션 연동의 이론적 배경	4
2.1 시뮬레이션 연동기술과 HLA/RTI	4
2.2 분산시뮬레이션 설계 및 구현 시 고려요소	16
III. Test bed 설계 및 구현	21
3.1 분산 시뮬레이션 개발 시 문제점	21
3.2 HLA/RTI 기반 Test bed 요구사항	23
3.3 HLA/RTI 기반 Test bed 설계 및 구현	24
IV. Test bed 운용 및 결과	32
4.1 HLA/RTI 기반 Test bed 운용절차	32
4.2 HLA/RTI 기반 Test bed 적용결과	34
V. 결론 및 향후 연구 방향	42
참고문헌	44
ABSTRACT	46

표 목 차

[표 2-1] 시뮬레이션의 분류	5
[표 2-2] 국방 M&S의 유형	6
[표 2-3] 모델과 시뮬레이션의 속성	7
[표 2-4] 국방용 연동구조 변천사(1983-)	11
[표 3-1] Object Class Structure Table	24
[표 3-2] Interaction Class Structure Table	24
[표 4-1] 33ms 주기로 객체정보 30회 갱신 3회 실시 결과	39

그 립 목 차

〈그림 2-1〉 계층구조 별 M&S 적용분야	8
〈그림 2-2〉 분산 시뮬레이션 기술 발전과정	10
〈그림 2-3〉 을지포커스렌즈 연습의 시뮬레이션 체계	11
〈그림 2-4〉 HLA Interface Specification 개념도	14
〈그림 2-5〉 RTI의 Publish - Subscribe 메커니즘	16
〈그림 2-6〉 시뮬레이터 구조 개념도	17
〈그림 2-7〉 Memory Que Overflow의 예	18
〈그림 2-8〉 인과성 오류의 예	19
〈그림 3-1〉 Test bed Simulator GUI	27
〈그림 3-2〉 페더레이트 가입 / 탈퇴	28
〈그림 3-3〉 페더레이트 정보 수신 창	28
〈그림 3-4〉 객체 생성 / 상태정보 갱신	28
〈그림 3-5〉 상호작용 메시지 송신	29
〈그림 3-6〉 반복 송신 기능	29
〈그림 3-7〉 송신 / 수신 로그화면	30
〈그림 3-8〉 Test bed Simulator 구성 개념도	31
〈그림 4-1〉 Test bed 시뮬레이션 망 테스트 개념도	33
〈그림 4-2〉 반사-Latency 측정모드 기능	33
〈그림 4-3〉 100개의 객체 상태갱신정보 송신화면	36
〈그림 4-4〉 100개의 객체 상태갱신정보 수신화면	36
〈그림 4-5〉 1개 객체 상태갱신정보 100회 송신화면	38
〈그림 4-6〉 1개 객체 상태갱신정보 100회 수신화면	38
〈그림 4-7〉 33ms 주기로 객체정보 30회 갱신 3회 송신화면	40
〈그림 4-8〉 33ms 주기로 객체정보 30회 갱신 3회 수신화면	40

I. 서 론

1.1 연구의 배경과 목적

현대의 무기체계는 획득과정에 있어서 국방분야의 특수성 상 대량생산 대량소비의 민수물품과는 달리 비교적 소량의 주문생산으로 이루어지며, 요구사항을 충족시키기 위해 고도의 최첨단 기술을 필요로 함은 물론이거니와 이를 구성하는 부품과 원자재 역시 고수준의 요구조건을 만족시켜야만 한다.

요구조건을 만족하는 무기체계를 개발하기 위해서는 반복적인 단위/통합시험이 필수적이다. 과거의 경우 시제품을 제작하고 실험하는 과정에서 파손, 망실, 설계변경에 따른 추가제작 등으로 인해 비용이 과다 발생되었고 이는 신규 무기체계 개발의 위험요소로 작용하였으나, 현대에는 미리 전산으로 수십, 수백 회 모의 시험을 실시하여 위험 및 결함요소를 사전에 식별, 제거할 수 있기 때문에 무기체계 개발기간과 비용이 대폭 감소되었다.

컴퓨터 기술의 발전은 무기체계 획득분야 뿐 아니라 분석 및 훈련 분야에도 큰 영향을 미쳤다. 이제는 전산으로 전쟁을 사전에 모의 수행 하여 전략 및 전술의 검증과 효과를 확인할 수 있을 뿐만 아니라 위게임 및 훈련용 시뮬레이터를 이용하여 가상으로 전투훈련을 실시하는 등 국방분야 전 범위에 걸쳐 컴퓨터 기반 M&S가 주는 이점이 매우 크다고 볼 수 있다.

이처럼 컴퓨터로 수행하는 M&S의 효과가 가시적으로 확인되는 가운데, 무기체계 획득, 훈련 및 분석에 있어서 M&S의 적용은 필수가 되었고 여기에서 한발 더 나아가 비용은 절약하면서 기 개발된 M&S도구를 재활용하여 원하는 결과 데이터를 산출해내기 위해 시뮬레이션 간 연동의 필요성이 대두되었다. 기 개발된 시뮬레이션 간 연동이 제대로 이루어지기만 한다면 처음부터 새로 시뮬레이션 도구를 개발하는 것 보다 비용과 시간, 노력을 훨씬 절약하면서도 원하는 결과 데이터를 얻을 수 있기 때문이다.

독립적으로 개발되는 M&S도구들 간의 상호운용성을 보장하기 위해 그동안 시뮬레이션 기술구조는 꾸준히 개발, 발전되어왔으며 그 중에서도 근래 가

장 널리 쓰이고 있는 기술을 꼽자면 바로 미 국방성 표준시물레이션 구조인 HLA/RTI 라고 할 수 있다. HLA/RTI는 국방 M&S도구의 재사용 및 효과적인 개발을 위해 정립한 기술구조로써 우리나라에서도 분산시물레이션 기반 M&S도구개발에 대부분 적용되고 있다.

단일 시물레이션과 같이 독립적으로 수행되는 M&S도구와는 달리, 다수의 시물레이터가 복합적으로 연동되는 분산시물레이션 기반 M&S도구를 개발하기 위해서는 몇 가지 추가적으로 고려해야 할 사항들이 있다. 그 중 하나는 결과물로 구현될 시스템이 앞으로 수행될 시물레이션 중 오가는 메세지량을 충분히 처리 가능한지의 여부이다.

현재 분산 시물레이션 기반 M&S도구 개발은 일반적인 SW개발 방법 프로세스와 같이 요구분석-설계-구현-테스트 의 순서로 개발이 된다. 그러나 시물레이션의 목적, 특성, 시나리오에 따라 시물레이션 네트워크에 걸리는 메시지 부하량이 매우 가변적이기 때문에 요구분석 단계에서 이를 미리 예측하기란 쉽지 않으며, 구현단계에서 확인된 시스템의 메시지 처리능력이 요구조건에 미달할 경우 개발일정의 지연은 물론 이를 수정하는데 소요되는 비용도 추가적으로 발생하는 위험이 있다.

본 논문에서는 시물레이션 수행 시 네트워크에 영향을 미치는 설계변수들의 종류와 그 영향에 대하여 확인하였다. 그리고 시물레이션 별 특성에 맞추어 사용자가 임의로 시물레이션 망의 부하 및 처리량을 테스트 해 볼 수 있는 HLA/RTI 기반 Test bed를 제안하고 이에 대한 실증적인 Sample Case를 제시한다.

1.2 연구의 범위와 방법

HLA/RTI 기반 시물레이션은 그 목적에 따라 모델 및 시물레이션의 구성과 특징에 각기 차이점이 있다. 그 중에서도 시물레이션 시간(Simulation Time)과 실제시간(Wall Clock)이 동일한 속도로 흘러가야만 하는 경우에는 어떠한 이유에서든 통신지연이 허락될 수 없는데 대표적인

경우로 훈련용 가상 시뮬레이션이 있다. 현실세계를 모사한 시뮬레이터는 이를 운용하는 훈련자에게 실제와 같은 몰입감을 주어야만 하는 것이 기본 요구사항이다. 그러나 어떠한 사유로 인해 시뮬레이션 시간에 지연이 일어나 화면이 끊긴다든지, 조작 입력이 반영되지 않는다든지 하는 문제가 발생할 경우에는 훈련수행에 중대한 문제가 생긴다.

본 논문에서는 광범위한 모델 및 시뮬레이션 분류 중에서도 실제 시간과 시뮬레이션 시간의 진행속도가 동일한 시뮬레이션에 대하여 연구를 하는 것으로 범위를 결정하고, 네트워크 부하로 인한 시뮬레이션 시간지연을 방지하는 시뮬레이션 설계/구현 이슈사항과 이를 사전에 시험해 볼 수 있는 HLA/RTI 기반 Test bed를 제안하였다.

이를 위하여 본 논문에서는 총 4개의 장으로 목차를 구성하였다.

1장에서는 연구배경 및 필요성과 연구 범위 및 방법에 대해서 설명하였다. 2장에서는 본 연구의 바탕이 되는 국방M&S의 소개 및, 시뮬레이션 연동기술, HLA/RTI의 메커니즘에 대해서 설명하였다. 3장에서는 실제 HLA/RTI 기반 분산시뮬레이션 개발 시 문제가 되었던 이슈사항을 확인하고 이를 고려한 Test bed 설계 및 구현내용을 기술하였다. 4장에서는 개발된 Test bed의 운용하는 절차와 이를 통한 간단한 시험을 수행하고 그 결과를 분석하였다. 마지막 5장에서는 연구내용의 요약과 함께 향후 발전/개선시킬 수 있는 분야에 대해 제안하고 결론을 맺었다.

II. 시뮬레이션 연동의 이론적 배경

2.1 시뮬레이션 연동기술과 HLA/RTI

2.1.1 국방M&S 소개 및 M&S종류별 특성

합동참모본부에서 발행한 “국방시뮬레이션 모델 목록(2007.12)”의 정의에 따르면, 국방 M&S(Modeling & Simulation)란 기존의 위게임 영역을 대폭 확대하여 국방기획관리상의 소요제기·결정, 획득, 분석평가는 물론 군의 교육훈련까지를 과학적으로 지원하는 도구 및 수단을 총칭을 뜻한다. 여기서 모델링(Modeling)이란 현실세계의 상황을 명료하고 개념적으로 표현하기 위해, 의도된 목적에 따라 단순화/이상화/추상화 하여 표현하는 과정을 말하는데 국방M&S에서의 모델링은 군사작전(전투)을 모의하기 위하여 적과 아군부대 및 전투체계, 작전지역을 포함한 전장 환경, 자연 및 인공 현상과 전술교리 등 전투에 미치는 각종 요소와 모의절차, 과정 등을 물리적, 수학적, 논리적 표현으로 만들어 가는 과정으로 한정지을 수 있다. 그리고 시뮬레이션(Simulation)이란 연속적인 시간의 흐름 속에서 모델을 통해 실험함으로써 현실세계의 시스템을 이해하고 문제들의 결과를 예측하는 것을 뜻하며 국방M&S에서의 시뮬레이션은 군사작전(전투)을 실제와 유사하게 실행하는 것으로, 복잡한 군사작전 과정을 수치적·물리적 모델로 표현하여 그 결과를 계산적으로 처리하는 기법을 뜻한다.

미국 국방과학위원회는 1992년에 [표 2-1]과 같이 M&S를 크게 구성 시뮬레이션, 가상 시뮬레이션, 그리고 실제 시뮬레이션으로 분류하였다.

구성 시뮬레이션(Constructive Simulation)은 모의되는 시스템과 모의되는 사람이 시뮬레이션을 수행하는 유형으로, 실제 사용자는 시뮬레이션 시작 전 시나리오 작성에 개입하나 시뮬레이션의 진행과정에는 일절 참여하지 않는다. 구성 시뮬레이션의 대표적인 사례로는 위게임이 있다. 시뮬레이션의 진행이 처음부터 종료시점까지 사용자의 개입 없이 모델을 통해 자동수행 되는 특성 때문에 반복적인 시뮬레이션 수행으로 통계적인 신뢰성을 확보하는데 이점이

[표 2-1] 시뮬레이션의 분류¹⁾

Classes of Models and Simulations

DESCRIPTION	CLASSES	EXAMPLES
Wargames, models, analytical tools	CONSTRUCTIVE	• CEM (Army) • Brawler (Air Force) • ENWGS (Navy) • MTWS (Marine Corps) • Engineering, Cost, Support models
Systems simulated both physically and by computer. Troops in simulators fight on synthetic battlefields	VIRTUAL	• Aircraft/Vehicle/Ship simulations • Virtual Prototypes(e.g. NLOS) • Appended Trainers
Operations with live forces and real equipment in the field	LIVE	• REFORGER • Red Flag • National Training Center • Strike University • Instrumented prototypes

있으며 다양한 시나리오를 구성하여 시험이 가능하다. 때문에 공학급 모델을 이용한 체계/부체계 모의분석, 교전 임무급 모델을 이용한 위게임, 전투지휘 훈련, 모의전장 분석 등에 적용되고 있다.

가상 시뮬레이션(Virtual Simulation)은 모의되는 시스템에 실제 사람이 참여하는 시뮬레이션 유형으로써 대표적인 예로써 비행기 시뮬레이터가 있다. 실제장비와 유사한 물리적인 환경(시뮬레이터)에서 운용자가 장비 운용/조작 숙달훈련을 하는데 적합한 형태의 시뮬레이션이며 실 체계 운용 훈련 시 감당해야 하는 위험요소나 비용 등을 줄일 수 있고 장소의 제약 없이 반복적인 훈련이 가능하다. 탑재되는 모델의 상세도에 따라 시뮬레이터가 운용자에게 주는 몰입감이 높아지고 이를 통해 가상으로도 현실감 있는 훈련이 가능하다.

실제 시뮬레이션(Live Simulation)은 실제 시스템 운영에 실제 사람이 참여하는 시뮬레이션 유형으로 대표적인 사례로는 사격 훈련장, 과학화 훈련장 및 야외 기동훈련, 모의 비행 전투 훈련 등이 있다. 실제 시뮬레이션은 실제 작전환경에서 실 체계 하드웨어, 소프트웨어를 운용하여 수행하는 시뮬레이션으

1) Lalit K. Piplani. (1994). "Systems Acquisition Manager's Guide for the use of models and Simulations", Defense systems management college press, p.4-2.

로써 시제 개발된 신규 무기체계의 시험평가 및 실 체계 기반 훈련에 주로 적용되는 시뮬레이션 형태이다. 실험환경의 시간적/공간적 제약이 있고 안전성/경제적인 문제도 있으나 실제적인 데이터가 반드시 필요한 시험평가/검증/훈련분야에서는 적용되고 있다.

[표 2-2] 국방 M&S의 유형

구분	특성	유형
구성 시뮬레이션	- 수리논리 및 시나리오 기반 - 모의훈련, 전장모의분석 모델 - 위게임 전투지휘 훈련, 체계임무, 효과도 분석 등	- 위게임 모델 - 위게임 분석모델 - 공학급 체계모의분석/검증모델
가상 시뮬레이션	- 가상 합성전장환경 기반 - 모의훈련, 체계 모형화, 성능검증용 모의분석장비 - 시뮬레이터, HILS	- 시뮬레이터 - 가상시제 3차원 목업 등
실제 시뮬레이션	실 장비를 상호 연계하여 실 기동으로 수집되는 데이터를 통한 각종 모의분석 및 실험검증	- 美 JRTC - 육군 KCTC 등

국방M&S는 무기체계의 획득단계부터 운용효과성 평가, 훈련에 이르기 까지 광범위한 활동을 지원한다. 사용자의 목적에 따라 이에 적합한 세부 모델 및 시뮬레이션의 특징과 요구사항이 달라지는데 크게 공학급 M&S, 교전급 M&S, 임무/전투급 M&S, 전구/전역급 M&S로 분류할 수 있으며 이렇게 분류된 모델과 시뮬레이션을 M&S계층이라고 한다.

공학급 모델과 시뮬레이션(Engineering M&S)은 체계나 부체계 및 구성품의 성능, 비용, 군수지원성 등을 분석하거나 이들 요소사이의 상쇄분석 시 사용되는 M&S로 획득단계에서는 체계, 부체계의 설계 및 성능검증 및 개발지원하고 훈련 및 분석단계에서는 수리/물리적 모델에 기반을 둔 정교하고 실제적인 현실모사 및 분석이 가능하며 각 체계/부체계의 성능척도 - MOP(Measure of Performance)를 제공한다.

교전급 모델과 시뮬레이션(Engagement M&S)은 특정 표적이나 적 대상 무기체계에 대한 개별 무기체계의 효과도를 평가하는데 사용하는 모델로써 제

한된 시나리오에 의한 일대일, 소수 대 소수 무기체계간의 전투효과를 모의한다. 이 수준의 M&S는 공학적 모델로부터 얻은 수치들을 이용하여 체계 효과도(MOE, Measure of Effect)를 제공한다.

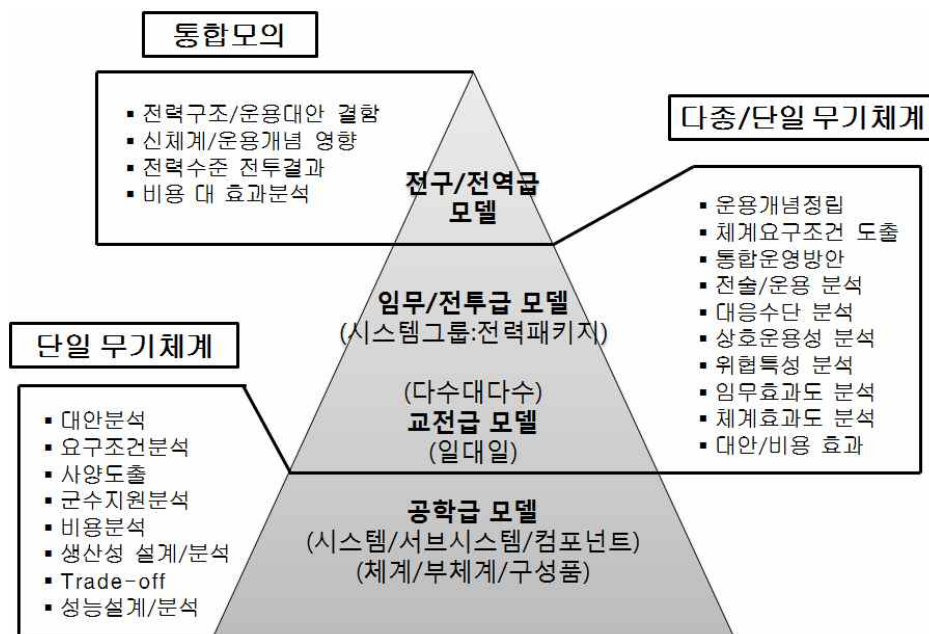
[표 2-3] 모델과 시뮬레이션의 속성²⁾

수준	공학급 모델	교전급 모델	임무/전투급 모델	전구/전역급 모델
전력대상	단일무기체계, 부체계, 구성품	일 대 일 소수 대 소수 다수 대 다수	다중플랫폼 다중 임무부대 패키지	합동/연합군
묘사수준	부분품/부품	개별개체/ 상세하부체계	개체 혹은 집약개체	고수준의 집약개체
모의시간	몇 달 - 몇 초	몇 분	몇 시간 - 몇 분	몇 주일 - 며칠
결과	-체계성능(MOP) (체계하부체계) 예: 탐지거리, 사거리 -비용, 지원성, 생산성	-체계효과도(MOE) 예: 살상확률, 손실, 생존성, 취약성	-임무효과도(MOE) 예: 손실교환율, 교전확률 -조우확률	-전쟁결과(MOO) 예: 공중우위, 전력기동, 전력소멸 -전력손실
사용분야	-설계 -하부체계/구성품 성능 및 상쇄분석 -명세요구 및 동의 -비용, 지원, 생산 성 분석	-대안평가(COEA) -임무/운용요구분 석 -체계효과분석 -체계상쇄분석 -전술, 교전규칙 분석 -시험평가지원	-대안평가(COEA) -임무/운용요구분 석 -전력배치분석 -무기통합분석 -전술, 운용개념 분석 -훈련/위게임	-대안평가(COEA) -임무/운용요구 분석 -전력배치/교리분석 -훈련/위게임 -참모훈련
사용목적	-체계/부체계/구성품 최적공학설계 -비용/지원성/생산성 분석	-체계효과도 -체계사양도출 -비용/성능/기술절충	-임무효과도 -부대구성분석 -소부대훈련	-합동전력평가 -합동작전분석 -대부대훈련
예	6-DOF CAD/CAM/CAE, HW/SWIL, Cost Factory Sim	Eagle, Janus SSTORM Brawler, EASAMS	Janus, Eagle, WEPTAC, EADSIM, SIMII Suppressor	CEM, ENWGS CAAM, AWSIM, THUNDER, MTWS

2) 김형현. (2009). “국방M&S개론”, 서울: 경성문화사, p.40.

임무/전투급 M&S(Mission/Battle M&S)는 위게임, 훈련 및 전술 개발 용도로 활용된다. 여기에 적용되는 대상은 개별 플랫폼이나 이를 구성하는 무기체계 수준이 아니라 부대 작전을 수행하는 다수의 전력 패키지 수준이다. 따라서 작은 소규모 교전의 단위가 아닌 목표임무수행과 관련된 범위이기 때문에 효과 척도(MOE)로는 손실교환비율, 교전확률 또는 특정임무 목표 달성 성공률 등으로 나타낼 수 있다.

전구/전역급 M&S(Campaign/Theater M&S)는 다양한 전력이 개입된 전쟁을 모의하며 장기적인 관점에서 전역 및 전쟁수준의 분쟁결과를 분석하는데 사용된다. 여기에 참여하는 개체들은 보다 낮은 단위의 전력과 체계를 종합하여 나타내게 되며 주로 장기간에 걸친 전쟁을 모사하기 때문에 체계, 부체계의 성능이나 효과도 보다는 좀 더 상위 집단이 되는 체계나 체계집단의 추세를 파악하고 군사적인 효용성을 상대적으로 비교하는데 활용된다.



〈그림 2-1〉 계층구조 별 M&S 적용분야³⁾

3) 김형현, 전제서, p.33.

2.1.2 시뮬레이션 연동기술

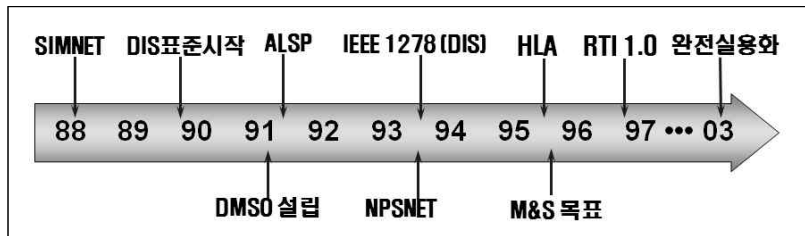
국방과학기술용어사전⁴⁾에서는 ‘연동’이란 개념을 ‘두 개의 체계가 물리적으로 직접 연결되어 자료를 교환할 수 있는 능력’ 또는 ‘컴퓨터와 통신을 활용하는 자동화 체계에서 서로 다른 기능을 수행하는 시스템 간, 장비와 장비 간, SW와 SW간의 기계적 또는 기능적 접속’이라고 정의하고 있다⁴⁾. 네트워크로 연결된 여러 컴퓨터들의 처리 능력을 이용하여 거대한 계산 문제를 해결하려는 분산처리 모델을 뜻하는 분산 컴퓨팅(Distributed computing)과 비슷하면서도 조금 다른 점은, 분산 컴퓨팅은 처음 구성할 때부터 하나의 목적을 가지고 이를 해결하기 위함을 전제로 다수의 시스템이 네트워크로 연결되었다면, 시뮬레이션이 연동된 분산 시뮬레이션(Distributed Simulation)은 처음 시뮬레이션을 구성할 시에 하나의 목표를 두고 다수의 시스템으로 분산처리를 위해 구성할 수도 있고, 처음에는 각기 다른 용도로 개발되어 사용되었던 별도의 모델들을 어떠한 목적을 위해 서로 데이터를 교환하여 얻고자 하는 결과를 도출하기 위해 연동시키는 구성을 할 수도 있다는 것이다.

1980년대 초반까지 시뮬레이터는 개별무기를 모의하여 승무원의 장비 조작 및 숙달 훈련을 실시하는데 중점을 둔 시뮬레이터 형태가 대부분이었으나 M&S에 대한 개념발전이 이루어지고 다수의 훈련자가 참여하는 조직적 전술 훈련방법으로의 이용가치를 인지하게 되면서 단독 시뮬레이터를 네트워크로 연결, 다수의 시뮬레이터가 함께 연동 되는 분산 시뮬레이션(Distributed Simulation)이 탄생하게 되었다. 이것이 바로 시뮬레이션 네트워킹의 시초가 된 SIMNET(Simulation NET-working)이다.

기존의 시뮬레이터는 훈련자 개인 차원의 훈련에 적합한 고정밀도의 훈련 장비라고 여겨져 왔으나 실제 작전수행은 혼자서 하는 것이 아닌, 다수의 부대와 편제가 함께 임무에 참여하기 때문에 현실성이 부족하다는 문제가 있었다. 따라서 보다 현실적인 훈련방식의 필요성이 미 국방부에 대두되었고 이것이 SIMNET 개발의 계기가 되었다.

SIMNET은 소부대급 훈련에서 저렴한 비용으로 효과적인 훈련을 제공하였으나 훈련에 필요한 다른 필수 묘사 요소, 지형, 플랫폼 추가 수정이 제한적

4) 국방기술품질원. (2011). “국방과학기술용어사전”, 서울: 미래미디어, p.588.



〈그림 2-2〉 분산 시뮬레이션 기술 발전과정⁵⁾

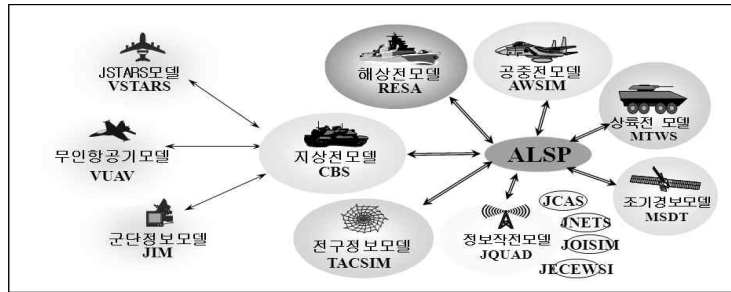
이었으며 주로 전차 위주의 전투에 편중되어 있었다. 또한 LAN 기반으로 개발되어 있기 때문에 보다 더 큰 규모의 분산시뮬레이션 환경으로 확장 시키는 데는 제한점이 있었다.

DIS(Distributed Interactive Simulation)는 이러한 SIMNET의 제한사항을 극복하기 위해 1989년 후반부터 미 육군에서 개발하기 시작하여 1993년에 IEEE 1278-1993으로 공인된 표준연동규약이다. DIS 시스템은 추후 미래에 개발될 분산 시뮬레이션들에 대한 요구사항들을 수용할 수 있도록 지원하고 이를 위한 확장성을 보장하기 위해 통합표준을 작성하였다. 다양한 유형의 시뮬레이션 모형과 시뮬레이터를 하나의 가상전장환경에 연결할 수 있게 하였으나 DIS는 기본적으로 SIMNET 시스템의 구조를 확장한 형태였기 때문에 제약사항 역시 그대로 가지고 있었고 따라서 대규모 사용자들이 참여하는 시뮬레이션은 제한되었다. 또한 같은 가상세계의 DB정보를 중복하여 가지고 있어야 했기 때문에 비효율적인 네트워크 사용의 문제가 있었다.

ALSP(Aggregated Level Simulation Protocol)는 기 존재하는 여러 전쟁 시뮬레이션을 상호작용 할 수 있도록 규약하고 있는 기술표준으로 DIS보다 좀 더 확장된 개념이다. ALSP 개발 이후 전구급 수준의 연동 시뮬레이션이 가능하게 되었다. 〈그림 2-3〉은 지난 2002년 시행된 을지포커스렌즈(UFL)에 참가한 시뮬레이션으로써 ALSP를 통해 다양한 모델들이 함께 연동되는 것을 확인할 수 있다.

HLA(High Level Architecture)는 미 국방부가 DM&S(Defense M&S) 체계와 전투 및 C4I 체계 간 상호운용성을 촉진하고, 기관 간 모형의 재사용성을 향상시키기 위해서 개발한, 향후 모든 국방M&S 체계가 준수해야 할 시뮬

5) 김형현, 전제서, p.47.



〈그림 2-3〉 을지포커스렌즈 연습의 시뮬레이션 체계⁶⁾

레이션 표준기술구조이다. DIS와 ALSP를 통해 배운 경험과 기술적 진보를 바탕으로 모든 종류의 M&S 상호운용성을 보장하기 위해 만들어진 HLA는 시뮬레이션 운용에 적용 가능한 설계규칙, 인터페이스 등을 정의하여 고수준에서 모든 M&S들이 준수해야 할 사항들을 정의하였다. 또한 HLA에 정의된 규약을 구현한 RTI(Real Time Infrastructure)를 통해 시뮬레이션 간 상호작용 정보를 송수신 할 수 있도록 하였다.

[표 2-4] 국방용 연동구조 변천사(1983-)⁷⁾

구조	연동체계	기간	표준	특징
하위 레벨	SIMNET (SIMulator NETworking)	1983~ 1990		1. 네트워크로 연결된 여러 대의 탱크 시뮬레이터 (팀 훈련용) 2. 여러 개의 동일한 시뮬레이터를 연결, 소부대 훈련
중간 레벨 (프로토콜)	ALSP (Aggregate Level Simulation Protocol)	1989~		1. ALSP Infrastructure Software(AIS)가 분산 시뮬레이션 환경 제공 2. 메시지 기반 프로토콜 3. 기존 서로 다른 시뮬레이션을 연결하여 연동 4. Joint Training Confederation(JTC) 체계의 기반 규약
	DIS (Distributed Interactive Simulation)	1990~	IEEE 1278	1. 실시간 시뮬레이션을 위한 UDP 기반의 통신 프로토콜 2. 확장성(Scalability) 문제 3. 비효율적 네트워크 자원 사용
상위 레벨 (공동기반)	HLA/RTI (High Level Architecture)	1995~	DMSO v1.3 IEEE 1516	1. 가능한 모든 종류의 시뮬레이션 사이의 연동을 위한 “공동된” 기반 구조 2. 시뮬레이터의 인터페이스 및 데이터 모델 규약

6) 김형현, 전제서, p.52.

7) 김탁곤. (2012). “국방 M&S이론 및 기술(II)”, 연동기술-기본개념 편, p.43.

2.1.3 HLA/RTI 개념 및 특징

2.1.3.1 HLA(High Level Architecture)

HLA에서는 시뮬레이션 목적 달성을 위해 구현된 모델을 제공하는 모의 소프트웨어의 재사용 단위를 페더레이트(Federate)라 하고, 이들이 모여 시뮬레이션을 이루고 시뮬레이션 수행 중 페더레이트들이 정보교환을 할 수 있는 네트워크 환경을 페더레이션(Federation) 이라고 한다. HLA는 HLA 규칙(HLA Rules), 인터페이스 명세(Interface Specification), 객체모델형판(Object Model Template)의 3가지 요소로 구성되어있다.

1) HLA 규칙(HLA Rules)⁸⁾

HLA 규칙은 HLA의 연합과 연합참여 체계가 지켜야 할 10가지 규칙으로 페더레이션과 페더레이트가 따라야 규칙을 각각 5가지씩 정의하였다. 세부 규칙의 내용은 다음과 같다.

가) 페더레이션 규칙

- ① 페더레이션은 HLA OMT에 따라 문서화 된 HLA FOM을 가져야 한다.
- ② 페더레이션에서는 모든 시뮬레이션 객체들이 RTI가 아닌 페더레이트가 가져야 한다.
- ③ 페더레이션 실행 중에는, 모든 FOM data 교환이 RTI를 통해 이루어져야 한다.
- ④ 페더레이션 실행 중에는, 참여한 페더레이트들은 HLA Interface Specification을 준수하여 RTI로 상호작용 하여야 한다.
- ⑤ 페더레이션 실행 중에는, 모든 객체속성은 단 하나의 페더레이트만 소유하여야 한다.

나) 페더레이트 규칙

- ① 페더레이트는 HLA OMT에 따라 문서화 된 HLA SOM을 가져야 한다.
- ② 페더레이트는 SOM에 명기된 대로 객체정보를 갱신하거나 반영할 수 있어야 하며, 상호작용을 송/수신할 수 있어야 한다.
- ③ 페더레이트는 SOM에 명기된 대로 페더레이션이 실행되는 동안 객체의

8) IEEE, (2000). "IEEE Standard for Modeling and Simulations (M&S) High Level Architecture (HLA) - Framework and Rules", p.15.

소유권을 동적으로 이양하거나 받을 수 있어야 한다.

④ 페더레이트는 SOM에 명기된 객체정보의 속성갱신을 제공하는데 있어서 조건을 변경할 수 있어야 한다.

⑤ 페더레이트는 시뮬레이션 시간을 관리 할 수 있어야 한다.

2) 인터페이스 명세(Interface Specification)⁹⁾

인터페이스 명세는 HLA를 따라 실체화 된 구현물인 RTI가 제공해야 할 서비스를 정의하고 있다. 그 서비스의 내용은 크게 7가지로 나뉘는데 기타 부수적인 서비스를 포함한 Support Service를 생략한 주요 6대 서비스는 다음과 같다.

가) 페더레이션 관리 서비스(Federation Management)

Federation Execution을 관리하는 서비스로써, 페더레이션 생성 및 삭제, 페더레이션 가입 및 탈퇴, 페더레이션 동기화, 페더레이션 저장 및 복구 등의 서비스를 포함한다.

나) 선언관리 서비스(Declaration Management)

페더레이션에 가입되어 있는 페더레이트 간 교환정보 선언에 관련된 서비스이다. 페더레이트의 공유객체에 대한 Publish/Unpublish, 페더레이트의 공유객체 정보에 대한 Subscribe/Unsubscribe, 페더레이트의 상호작용 정보에 대한 Publish/Unpublish, 페더레이트의 상호작용 정보에 대한 Subscribe/Unsubscribe 서비스를 포함한다.

다) 객체관리 서비스(Object Management)

페더레이션 내에서 모의 될 공유객체(Object)와 상호작용(Interaction) 메시지에 관련된 서비스이다. 주요 기능으로는 공유객체 생성 및 발견(Create/Discover), 공유객체 갱신 및 반영(Update/Reflect), 상호작용 메시지 송신 및 수신(Send/Receive), 공유객체 제거>Delete/Remove)가 있다.

라) 소유권 관리 서비스(Ownership Management)

페더레이션 내 공유객체의 갱신 및 삭제 권한을 획득하거나 이양하는 것과 관련된 서비스이다. 주요기능으로는 소유권 포기(Unconditional, Negotiated), 소유권 획득(Intrusive, None Intrusive), 소유권 포기 및 획득 취소(Cancel),

9) IEEE, (2000). "IEEE Standard for Modeling and Simulations (M&S) High Level Architecture (HLA) - Federate Interface Specification", pp.18-210.

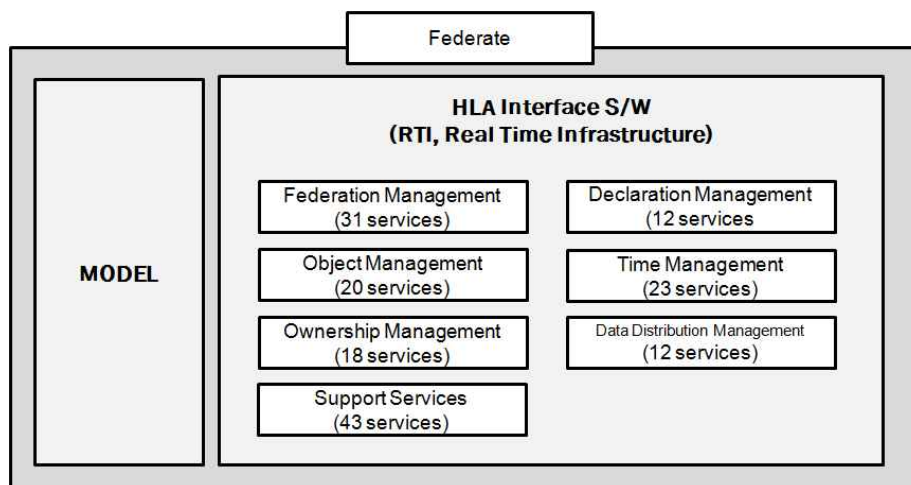
소유 여부 질의(Query) 등이 있다.

마) 시간관리 서비스(Time Management)

페더레이션 내에서 진행되는 시뮬레이션 시간관리 서비스이다. 페더레이트의 TSO 메시지 송신모드(Regulating), 페더레이트의 TSO 메시지 수신모드(Constrained), 시간진행 요청 및 수락(Time advance/Grant), 메시지 철회(Retraction) 등이 있다.

바) 메시지 분배관리 서비스(Data Distribution Management)

페더레이션에서 교환되는 메시지 양을 제어하기 위한 라우팅(Routing) 서비스이다. 주요기능으로는 Region(구역) 생성 및 삭제(Create/Delete Region), 공유객체와 Region 연관(Associate Region), Region 내 정보 송신(Send/Update Region), Region 내 정보 수신(Subscribe Region) 등이 있다.



〈그림 2-4〉 HLA Interface Specification 개념도

3) 객체모델형판(Object Model Template)¹⁰⁾

객체모델형판은 시뮬레이션 객체를 기술하기 위한 하나의 양식이다. 페더레이션에 참여하여 모의될 모든 객체들은 이 형식에 따라 표현되고 정보교환이

10) IEEE, (2000). "IEEE Standard for Modeling and Simulations (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification", pp.17–63.

이루어지는데 크게 3가지로 구별된다.

가) SOM(Simulation Object Model)

페더레이트 당 하나만 존재하며, 개별 시뮬레이션의 정보를 기술한 템플릿이다. 페더레이트의 객체/상호작용 메시지를 정의한다.

나) Federate Object Model(FOM)

페더레이션 당 하나만 존재하며 여기에 참여하는 모든 페더레이트의 메시지 교환정보를 모두 기술한다. 다시 말해 페더레이션에 가입하는 페더레이트의 SOM에 대한 합집합이다.

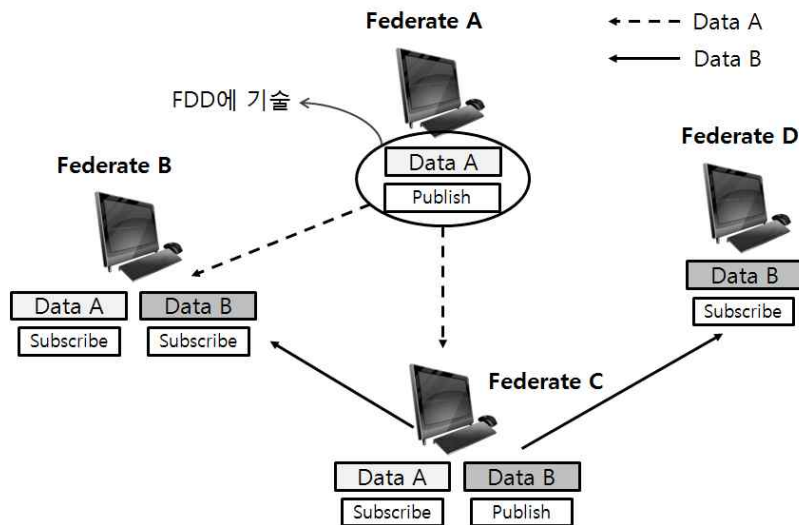
다) Management Object Model(MOM)

페더레이션 내 존재하는 모든 객체와 상호작용 정보를 식별하며 페더레이션 자체의 정보(상태 등) 또한 기술되어 있다.

2.1.3.2 RTI(Real Time Infrastructure)

RTI는 시뮬레이션에 참여하는 모든 시뮬레이터의 데이터 동기화 및 교환을 위해 HLA federation이 실행되는 동안 공통의 인터페이스 서비스를 제공하는 소프트웨어이다. HLA의 구성요소 중 하나인 Interface Specification이 실체화로 구현된 RTI는 시뮬레이션 개체의 재사용성(portability)과 상호운용성(interoperability)을 제공하며, RTI서비스와 사용자가 직접 개발하는 소프트웨어의 분리개발이 용이하고, HLA기반 분산시뮬레이션의 개발을 위한 서비스를 제공한다.

RTI는 페더레이트 간 데이터 송수신을 위해 Publish - Subscribe 메커니즘을 사용한다. RTI를 통해 데이터를 송수신하기 위해서는 먼저 FOM을 정의해야 한다. FOM(Federation Object Model)이란 HLA의 구성요소 중 하나인 OMT를 적용하여 기술된 객체모델 정보로써, 현재 페더레이션에서 사용될 객체, 속성, 상호작용, 파라미터를 정의하고, 각 페더레이트 간의 정보 송수신 관계를 기술한 HLA 객체모델이다. 그리고 이를 기술한 문서를 FDD(FOM Document Data)라고 하며 일반적으로 XML 의 형태로 기록이 된다.



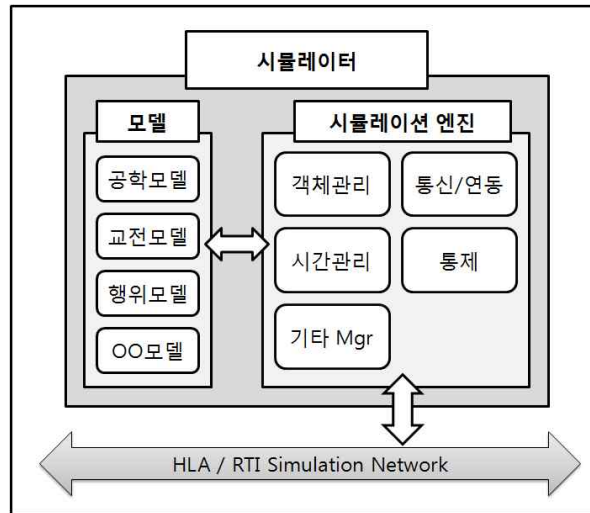
〈그림 2-5〉 RTI의 Publish - Subscribe 메커니즘

〈그림 2-5〉를 참고하면, FOM에 정의된 정보 송수신 관계에 따라 Data A는 Federate A에서 Publish, 송신하며 Federate B와 C에서는 Data A를 Subscribe, 수신하게 된다. 각 페더레이트의 RTI는 FDD에 기술된 데이터 송수신 정의를 참조하여 시뮬레이터에서 생성한 데이터를 송/수신한다. 시뮬레이터는 자신이 보내려고 하는 데이터를 어느 페더레이트에 보내야 할지 고민하지 않고 Publish 하면 수신측 페더레이트의 RTI가 FDD를 참조하여 자신이 Subscribe 해야 하는 데이터는 수신하고 그렇지 않은 데이터는 수신하지 않는다.

2.2 분산시뮬레이션 설계 및 구현 시 고려요소

분산시뮬레이션 설계에 앞서 일반적인 시뮬레이터의 기본 구조를 확인할 필요가 있다. HLA/RTI 기반 시뮬레이터는 크게 모델과 시뮬레이션 엔진(시뮬레이션 관리자)로 구성될 수 있다. 모델에는 목적대상 체계/부체계를 모사할 수 있도록 공학급 모델이나 교전급 모델, 행위모델, 피해평가모델, 수리기반 물리모델 등 다양한 모델이 단수, 복수로 포함될 수 있다. 그리고 시뮬레이션 엔진에는 모델로부터 데이터를 송수신 받아 현재 객체의 상태를 업데이트

트하고 관리할 객체관리모듈, 페더레이션에 참여시키고 다른 페더레이트와 상호작용을 할 수 있도록 하는 통신/연동 모듈, 페더레이션의 시간관리 및 통제에 대한 수행을 담당할 모듈 등이 있을 수 있다. 위와 같은 구조는 일반적으로 생각해 볼 수 있는 구성이며 시뮬레이터나 시뮬레이션의 성격, 목적에 따라 다르게 구성될 수 있다.

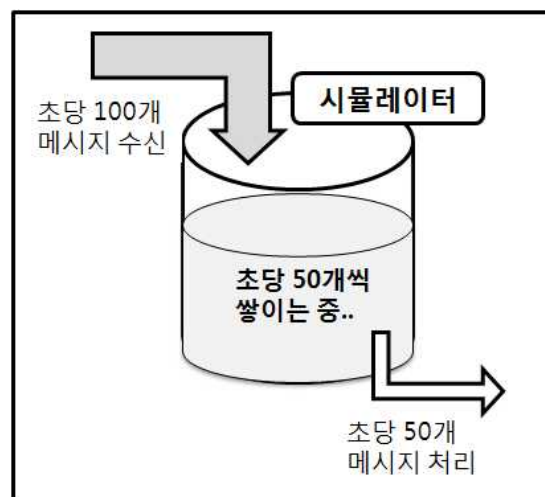


〈그림 2-6〉 시뮬레이터 구조 개념도

분산 시뮬레이션 구성 시 고려해야 할 부분 중 하나는 바로 네트워크 메시지 송수신 처리이다. HLA/RTI에서는 시간관리 서비스를 제공하는데, 이 서비스를 사용할 경우 메시지 순서 및 이벤트의 인과관계를 엄격히 통제하여 메시지가 모두 들어온 것을 확인할 때까지 시뮬레이션 시간을 진행시키지 않는다. 만일 시간을 진행 시킨 후에 이보다 더 과거의 메시지가 늦게 도착했을 경우에는 해당 시간으로 Roll-back 해서 다시 시뮬레이션을 수행하기 때문에 인과율이 깨지는 상황은 생기지 않는다. 그러나 이러한 상황에서 시뮬레이터의 메시지 처리 속도가 요구조건보다 느리다면 가상 시뮬레이션이 경우에는 문제가 심각할 수 있다. 가상 시뮬레이션은 모의된 체계에서 실제 사람이 운용하는 형태의 시뮬레이션인데 그 특성상 주로 훈련용 시뮬레이터가 이와 같은 형태를 취하는 경우가 많다. 훈련용 시뮬레이터에서는 훈련자가 느끼기에

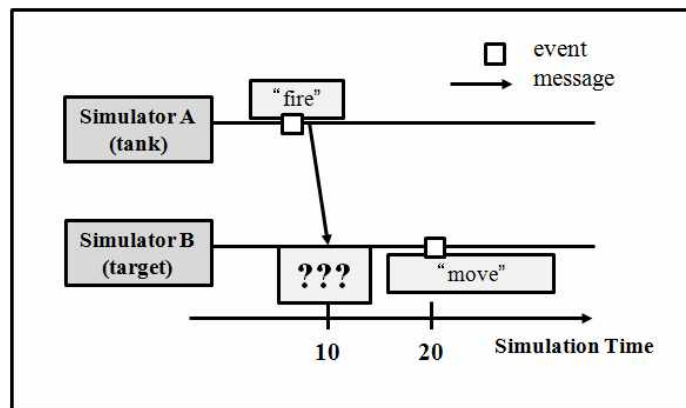
실제와 유사해야만 하는 요구조건 때문에, 시뮬레이션 시간이 실제 시간과 동일한 속도로 진행되어야 함에도 불구하고 메시지 처리에 병목현상이 일어나 더 이상 시뮬레이션이 진행되지 못하고 멈추거나 느리게 움직인다면 훈련용 시뮬레이터로서의 역할을 수행하기 어렵다. 이러한 문제 때문에 사실 훈련용 시뮬레이터에서는 이벤트의 인과관계를 엄격하게 검증/확인하지는 않는다. 훈련용 시뮬레이터는 분석용 시뮬레이션이나 획득용 시뮬레이션과 같은 수준으로 정밀한 데이터와 신뢰성을 요구하지 않으며 그보다는 원활한 훈련수행을 위하여 실시간성을 더 우선하기 때문이다. 시간 관리를 엄격하게 수행하면 좋겠지만 HLA/RTI의 시간관리 서비스는 구현이 복잡하고 시스템 자원도 많이 차지하기 때문에, 송수신 메시지가 많은 경우 실시간성을 보장하기 어렵다.

시간관리 서비스를 사용하지 않는 경우에도 대용량의 메시지 송수신 처리의 병목현상은 문제를 발생시킬 수 있다. <그림 2-7>과 같이 외부로부터 수신되는 메시지량이 초당 100개라고 가정해보자. 시뮬레이터의 처리능력이 초당 50개라면 매 초마다 50의 데이터가 내부에 쌓이게 된다. 컴퓨터의 메모리 자원은 유한하기 때문에 언젠가는 메모리 Que가 넘치게 될 것이고, 최악의 경우 시스템이 다운되게 된다.



〈그림 2-7〉 Memory Que Overflow의 예

또 다른 문제는 바로 인과성의 오류이다. <그림 2-8>과 같이 탱크 시뮬레이터 A에서 이동 중인 표적 B를 향해 ‘발사’ 이벤트 메시지를 발생시켰다고 가정해보자. 정상적인 상황이라면 포탄이 날아가는 시간을 감안하여 시뮬레이션 시간 10이 되었을 때 시뮬레이터 B가 ‘발사’ 메시지를 수신하여 ‘충돌’ 이벤트 메시지를 생성해야 할 것이다. 그러나 어떠한 문제로 ‘발사’ 메시지가 시뮬레이션 시간 10일 때 시뮬레이터 B에게 전달되지 못하고 시뮬레이션 시간 20이 되었을 때 도착한 경우에는 시뮬레이션 시간이 10만큼 진행되는 동안 시뮬레이터 B의 타겟이 이동한 상태이기 때문에 충돌 검출이 일어나지 않는 것이다. 다시 말해 시뮬레이터 B 입장에서는 포탄이 느리게 날아와서 피한 것이고 시뮬레이터 A 입장에서는 명중하여야 할 목표가 피격당하지 않고 이동한 것처럼 보이게 된다.



<그림 2-8> 인과성 오류의 예

이와 같이 시뮬레이터 엔진이 처리할 수 있는 능력과 시뮬레이션 망에서 송수신될 메시지의 양을 정확히 파악하고 조정하는 일은 신뢰성 있는 시뮬레이션 결과데이터를 얻어내기 위해서 중요하다. 이를 위해 분산시뮬레이션 설계 시 고려할 사항은 다음과 같다.

첫째, 시뮬레이션 진행 중 모의할 객체의 적정 수량을 결정해야 한다. 시뮬레이션 내의 가상 세계에서 모의할 객체 수가 많아지면 많아질수록 갱신해야 할 각 객체의 상태정보 메시지도 그와 비례하여 늘어난다.

둘째, 시뮬레이션에서 송수신되는 상호작용 메시지의 종류와 빈도를 결정해야 한다. 한 객체에서 다른 객체로 직접 메시지를 전달하는 경우, 예컨대 사격 또는 전문 송신 등을 할 경우 부가적으로 네트워크 부하량이 늘어난다. 특히 워게임 시뮬레이션의 교전상황에서는 굉장히 많은 추가적인 트래픽이 발생하게 된다.

셋째, 객체상태메시지 갱신 주기이다. 객체의 상태를 1분에 한번 씩 갱신하느냐, 1초에 한번 씩 갱신하느냐 등 객체상태의 갱신주기에 따라서 네트워크 부하량의 큰 차이가 있다. 이는 시뮬레이션의 성격에 따라 요구사항이 달라질 수 있으나 훈련용 가상시뮬레이션인 경우 화면 주사율과 같은 빈도로 하게 되면 굉장히 부하가 커지기 때문에 빈도를 낮추고 Dead Reckoning 기법을 적용하기도 한다.

넷째, 송수신 되는 메시지/데이터 크기이다. RTT로 송수신 하는 모든 데이터/메시지는 일정한 규칙에 의해 인코딩/디코딩 처리되어 송수신 된다. 데이터의 크기가 커질수록 인/디코딩 프로세스에 걸리는 시간이 길어지기 때문에 가능한 불필요한 데이터는 메시지에 포함시키지 않는 것이 좋다.

III. Test bed 설계 및 구현

3.1 분산 시뮬레이션 개발 시 문제점

HLA/RTI 기반 분산시뮬레이션 개발 시 발생했던 네트워크 이슈의 실제 사례를 소개하고 이를 통해 확인했었던 문제점과 대처방안 및 시사점을 제시한다.

사례로 제시한 개발 과제는 HLA/RTI 기반 시뮬레이션 개발 프로젝트로써, 다수의 페더레이트가 연합하여 가상의 전장상황을 모의하고 무기체계의 효과도를 사후 분석하는데 그 목적이 있었으며, 공학급 모델 기반 가상 시뮬레이션과 교전급 모델 기반 구성 시뮬레이션의 연동이 부가 요구사항으로 주어졌다. 가상 시뮬레이션은 공학급 모델을 기반으로 사용자가 무기체계를 직접 실시간으로 조종하여 시뮬레이션에 참여하는 형태였으며 구성 시뮬레이션은 개략 묘사된 다수의 CGF¹¹⁾객체들을 생성시켜 가상전장에 참여시켰다.

개발 초기단계의 요구사항으로 시뮬레이션 중에 모의될 객체 수는 가상 시뮬레이션 측과 구성 시뮬레이션 측 모두 합하여 최소 300여개 이상으로 정하였다. 그리고 가상 시뮬레이션 측의 운용자 시뮬레이터에서는 부드러운 화면 프레임 수를 유지하기 위해서 객체정보 갱신주기를 초당 60회로 결정하였으며 시뮬레이션 진행시간 속도는 실제 시간속도와 동일하게 설정하였다. 또한 시뮬레이션 상호운용성을 보장하기 위하여 미국 SISO¹²⁾의 Reference FOM인 RPR-FOM¹³⁾을 따랐다.

요구사항을 확정하고 개발을 진행하면서 RTI 연동시험을 수행한 결과 설계 단계에서 미리 예측하지 못한 상황들이 연출되었다. 실제 시뮬레이션을 수행하면서 교전 시 상호작용을 위한 발사, 충돌, 전문 송수신 메시지가 폭증하였다. 또한 포탄이나 유도무기를 묘사할 경우, 사격 시 많은 추가 객체가 생성되었고 이들 객체 또한 각자의 상태갱신메시지를 생성하였다. 이로 인해 네트

11) CGF : Computer Generated Forces, 모델에서 생성한 가상군

12) SISO : Simulation Interoperability Standard Organization

13) RPR-FOM : Real-time Platform-level Reference Federation Object Model

워크 지연이 발생하여 시뮬레이션 진행에 어려움이 발생하였다. 또한 다양한 서비스를 제공하는 만큼 시스템의 리소스를 많이 사용하는 RTI의 메시지 수신처리능력이 요구 성능을 하회 하였을 뿐 아니라, 여기에 함께 연동되는 사후분석기나 기타 모델들도 초당 60회의 갱신주기로 300여개의 객체정보를 수신 하기는 현실적으로 어려웠다. 이는 1초당 18,000개(300개 객체×초당 60회)의 메시지를 수신하고 기록해야 하는 상당히 부하가 큰 수치이기 때문이다.

상기와 같은 제한사항 때문에 원활한 시뮬레이션 수행을 위해서 설계 및 구현상의 수정이 불가피하였다. 먼저 가상전장에서 활동하는 객체의 수를 대폭 축소하여 보다 소규모의 전투를 상세히 시뮬레이션 하는데 초점을 두었다. 그리고 모의 시뮬레이터의 전시를 위해 객체정보 갱신을 초당 60회로 설정했던 것을 초당 30회로 낮추고, 화면이 끊겨서 이질감이 느껴지는 부분은 Dead Reckoning 기법으로 처리하여 해결하였다. 또한 구성모의모델에서 모사하는 정밀도 낮은 다수의 CGF객체들은 상태갱신 주기를 초당 10회 수준으로 낮추었다. 뿐만 아니라 객체정보 송수신의 기본 틀이 되는 RPR-FOM도 수정하여, 본 과제에서 사용하지 않는 객체속성들은 전부 제외하고 데이터 크기를 줄여 메시지 인코딩/디코딩 하는데 소요되는 시간을 단축시킴으로써 시뮬레이터 엔진의 처리속도를 높였다.

이러한 대안들을 통해서 결국에는 시뮬레이션을 원활히 진행하면서도 사후분석에 필요한 충분한 데이터를 산출해내는데 성공을 하였다. 그러나 개발기간 중 수많은 시행착오와 요구사항 변경, 시험 및 수정에 소요되는 많은 물적, 인적, 시간적 자원의 소모는 피할 수 없었다. 본격적인 구현단계에 들어가기 전에 실현 가능한 시뮬레이션의 성능한계를 미리 알 수 있었더라면 요구사항 결정이나 시뮬레이션 설계단계에서 먼저 반영하거나 다른 대안을 찾을 수 있었겠지만, 동시에 다수의 컴포넌트가 병렬적으로 개발되는 소프트웨어 개발상황에서, 연동데이터를 생성하는 모델의 개발이 완료되기도 전에 RTI 연동 네트워크 테스트를 하기란 어렵다. 따라서 별도의 HLA/RTI 기반 네트워크 연동 테스트가 가능한 Test bed가 없이는 위와 같은 시뮬레이션 개발의 위험성을 완전히 회피 할 수 없다는 결론을 얻게 되었다.

3.2 HLA/RTI 기반 Test bed 요구사항

시뮬레이션 망 네트워크 테스트를 하기 위해서는 첫째로 시뮬레이션 객체의 정보를 생성해주는 모델과, 모델이 생성해주는 데이터를 받아서 시뮬레이션 중에 객체를 만들고 정보를 업데이트 시켜주는 시뮬레이션 엔진, 그리고 송수신 메시지가 언제 송신되고 수신되었는지, 메시지 내용은 정확히 들어왔는지 등의 확인을 위한 사후분석기가 필요하다.

위와 같은 요구사항을 만족시키기 위해 사전에 시뮬레이션의 요구사항 및 제한사항을 확인/검증 할 수 있도록 HLA/RTI 기반 시뮬레이션 망의 네트워크 부하량을 시험할 수 있는 Test bed가 요구되었다.

Test bed의 목적은 현재 시스템으로 구성된 시뮬레이션 네트워크의 데이터 송수신 능력 측정이기 때문에 실제 모델은 필요하지 않다. 일반적인 시뮬레이션 환경에서는 모델의 객체갱신정보 생성속도가 시뮬레이션 엔진의 메시지 송수신 속도보다 우위에 있어서 시뮬레이션 망 네트워크 성능에 영향을 미치지 않는다고 가정한다. 따라서 Test bed의 모델은 실제 모델이 아닌 임의의 dummy 데이터를 반복적으로 시뮬레이션 엔진에 입력해주는 기능으로 구현하면 된다.

또한 HLA/RTI 기반 Test bed이기 때문에 시스템에서 앞으로 사용할 상용 RTI의 라이브러리를 사용해야만 한다. 이 Test bed에서는 MAK社의 MAK RTI를 사용하였다. MAK RTI는 유료 라이선스가 없어도 최대 2개의 페더레이트를 사용할 수 있도록 지원하여 RTI를 이용한 메시지 송수신을 할 수 있는 최소한의 요건을 충족시켜 줄 수 있다.

또한 메시지의 정확성과 송수신 시간을 확인하기 위해서 이에 특화된 로그 저장기가 필요하다. 수신된 메시지의 내용을 디코딩하여 보여줄 뿐 아니라 어떤 종류의 메시지가 언제 몇 개가 들어왔는지 시각과 수량으로 보여줄 수 있어야 한다.

3.3 HLA/RTI 기반 Test bed 설계 및 구현

3.3.1 FOM 설계

HLA/RTI 기반 Test bed를 설계하기 위해서는 RTI 기반 하에 구현되기 때문에 먼저 Test bed가 운영할 시뮬레이션을 정의하는 FOM을 설계하여야 한다. 시뮬레이션을 개발할 때는 원칙적으로 FEDEP¹⁴⁾라는 절차를 따라서 시뮬레이션의 목적과 특징에 따라 FOM을 설계해야 하지만, Test bed는 목적의 특성상 가장 일반적이고 공통적인 특성을 지녀야 하기에 RPR-FOM을 기본으로 따랐다. 그 중에서도 가장 일반적으로 쓰이는 Object Class와 Interaction Class 항목을 선별하여 시험을 지원하는데 그 목록은 [표 3-1], [표 3-2]와 같다.

[표 3-1] Object Class Structure Table

Class 1	Class 2	Class 3	Class 4
BaseEntity	EnvironmentalEntity		
	PhysicalEntity	Platform	Aircraft
			GroundVehicle
		Lifeform	Human
		Munition	
Embedded System	RadioTransmitter		

[표 3-2] Interaction Class Structure Table

Interaction Class	비고	Interaction Class	비고
Acknowledge	응답	Event Report	전문
Action Request	행위요청	Remove Object Request	객체삭제 요청
Action Response	행위요청 응답	Remove Object Result	객체삭제 응답
Collision	충돌	Start / Resume	시작 / 재개
Create Object Request	객체생성 요청	Stop / Freeze	중지 / 일시정지
Create Object Result	객체생성 응답	Weapon Fire	사격

14) FEDEP : Federation Development and Execution Process, 페더레이션 개발 및 실행 표준절차

RPR-FOM은 SISO에서 제안하는 권장 Reference FOM으로써, 모든 HLA/RTI 기반 연동 시뮬레이션의 표준 FOM은 아니지만, 앞으로 개발된 많은 페더레이션들이 기본적으로 RPR-FOM과의 상호운용성을 고려하여 설계되기 때문에 이를 차용하였다. 그러나 시험에서의 사용빈도가 낮거나 한국 군 실정과 맞지 않는 Object, Interaction Class는 제외하였다

3.3.2 Test bed 기능 설계

Test bed가 수행할 시뮬레이션을 정의하는 FOM을 설계 한 다음에는 Test bed Simulator의 요구사항을 만족시키기 위한 기능 설계를 수행해야 한다.

Test bed Simulator의 목적은 시뮬레이션 망에서 송수신 되는 데이터를 동일하게 모사하여 시스템이 처리 가능한 메시지 부하량이 얼마나 되는지 확인할 수 있도록 시험하고 그 결과를 제공하는 것이다. 이를 수행하기 위하여 필요한 요구사항을 세부적으로 식별하면 다음과 같다.

- 1) Federation 내 객체수량 별 데이터 송수신 시험 및 결과 확인
- 2) Federation의 정보갱신주기 별 데이터 송수신 시험 및 결과 확인
- 3) Object Class, Interaction 메시지 크기 별 데이터 송수신 속도 시험 및 결과 확인

다음은 위 요구사항을 만족시키기 위하여 필요한 기능을 상세설계단계에서 식별, 구현한 항목이다.

- 1) Federation 생성/파괴, Federate 가입/ 탈퇴기능

HLA/RTI를 이용하여 시뮬레이션을 하기 위하여 전체 시뮬레이션에 해당하는 페더레이션과 시뮬레이션에 참여하는 개별 시뮬레이터인 페더레이트를 생성하고 삭제하는 기능

- 2) Federate Status 확인 기능

현재 페더레이트가 페더레이션에 정상적으로 가입되어있는지, 정상적으로 탈퇴하였는지 확인하기 위해 Event Report Interaction 메시지를 활용하여 페더레이트의 상태정보

를 송수신 가능하는 기능

3) Object 생성, 삭제, 갱신 기능

시뮬레이션 상에서 모사될 모의객체를 생성하고 표현, 삭제하기 위한 Object 생성, 삭제, 갱신 기능

4) Object 객체 생성 수량, 객체 ID번호 부여기능

복수의 모의객체 생성을 위한 객체생성 수량 설정 및 ID 부여 기능

5) Interaction 메시지 송/수신 기능

RTI를 통한 상호작용 메시지 송수신을 위한 Interaction 메시지 송/수신 기능

6) 묶음단위 반복 송신 기능

대량의 메시지를 자동으로 송신하여 네트워크 부하량 시험을 하기 위한 기능

7) 밀리세컨드 단위 주기별 반복전송 기능

느슨한 시간관리 정책을 사용하는 시뮬레이션을 모사하여 정해진 주기별로 객체 속성정보를 정해진 시뮬레이션 시간동안 송신하기 위한 기능

8) 반복전송 횟수 지정 기능

정량적인 설정값을 설정하여 시험을 하기 위해 사용자가 원하는 횟수만큼 메시지를 반복적으로 송신하는 기능

9) 로그 전시 기능

송수신 된 데이터 확인, 송수신 객체 ID 정보, 송수신 시각, 반복회차 등을 확인하기 위한 데이터 송수신 로그 전시 기능

10) 전시된 로그를 text 파일로 저장하는 기능

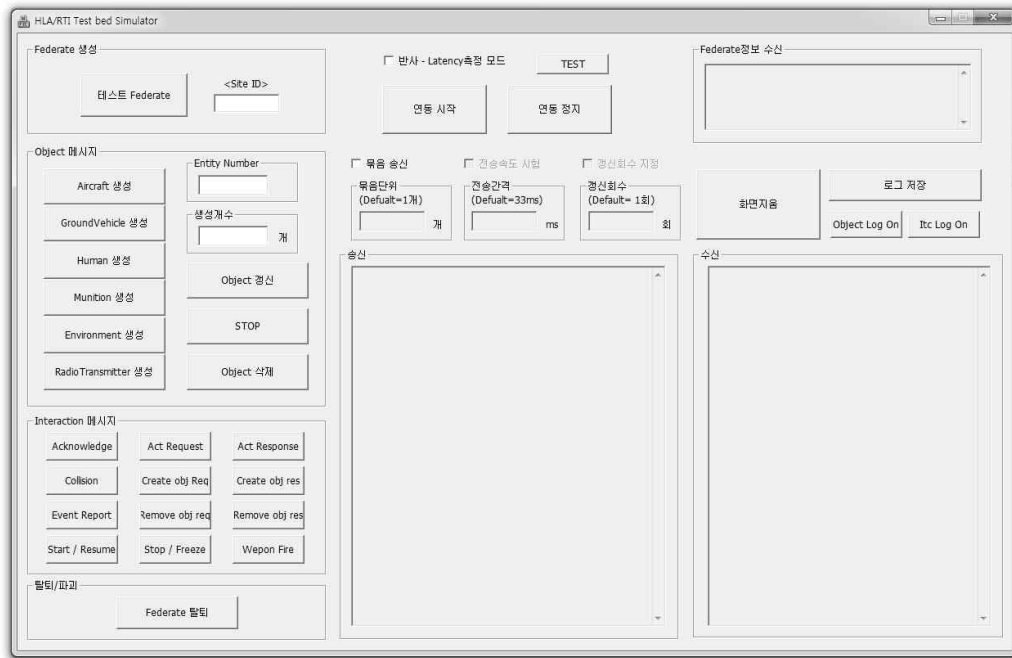
GUI상의 로그를 사후 분석에 별도 활용할 수 있도록 텍스트 파일로 저장하는 기능

11) 1:N 데이터 송수신 시험 기능

사전에 지정된 단일 페더레이트로 복수의 페더레이트가 동시에 다수의 메시지를 송신하는 시험을 수행하기 위한 기능

3.3.3 GUI 설계

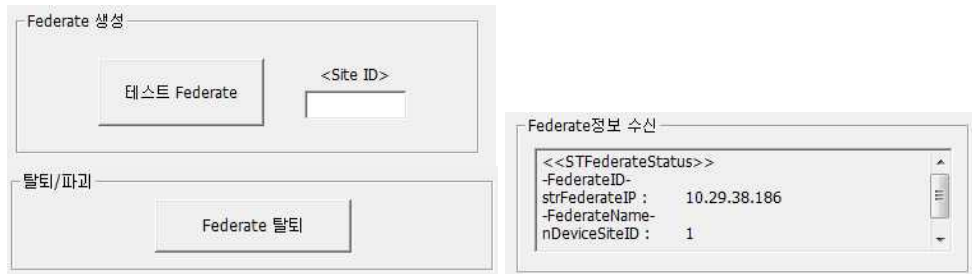
Test bed Simulator의 요구사항과 필요기능을 정의 한 다음, 이를 기반으로 GUI를 설계/구현하였다. Test bed Simulator는 페더레이트 관리, 종류별 객체 생성, 갱신, 상호작용 메시지 송신, 송/수신 로그 창, 반복송신 설정 메뉴 등을 사용자가 간편하게 사용할 수 있도록 구성되어야 한다.



〈그림 3-1〉 Test bed Simulator GUI

1) Federate 가입 / 탈퇴

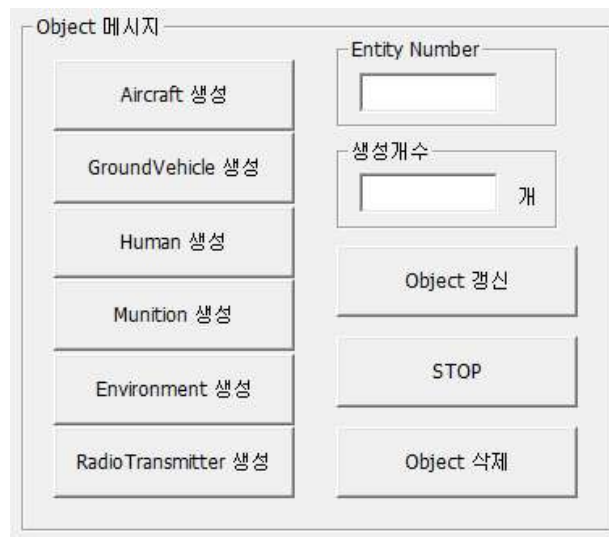
페더레이트 가입 시 페더레이션이 생성되지 않았다면 먼저 페더레이션을 자동으로 만들고 페더레이트가 가입되게끔 설계하였다. 다수의 페더레이트가 참여할 경우 서로를 구분할 수 있도록 Site ID를 입력하여 우측의 페더레이트 정보 수신 창에 현재 페더레이션에 가입되어있는 페더레이트 정보 메시지를 수신 받아서 전시할 수 있게 하였다.



〈그림 3-2〉 페더레이트 가입 / 탈퇴 〈그림 3-3〉 페더레이트 정보 수신 창

2) 객체 생성 / 상태정보 갱신

FOM에 정의된 6종의 객체를 종류별로 생성할 뿐 아니라 객체의 수를 원하는 만큼 생성하여 각각의 객체마다 상태정보를 갱신할 수 있도록 구현하였다. 또한 객체의 ID Number도 설정하여 구별할 수 있게끔 하였다.



〈그림 3-4〉 객체 생성 / 상태정보 갱신

3) 상호작용 메시지 송신

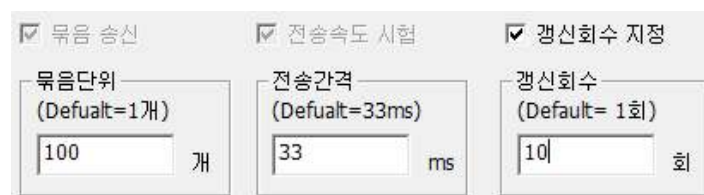
FOM에 정의한 상호작용 Interaction 메시지 12종을 클릭과 동시에 송신할 수 있도록 구현하였다.



〈그림 3-5〉 상호작용 메시지 송신

4) 반복 송신 기능

시뮬레이션 중에는 메시지 송수신이 연속적으로 들어오며, 다수의 페더레이트가 연합할 경우 발생하는 네트워크 부하를 재현하기 위해서 기 생성된 객체의 갱신에 대한 반복송신 조건을 지정해 줄 수 있도록 구현하였다. 묶음단위는 현재 생성된 객체를 반복 업데이트 하는 것이고, 전송간격은 반복 송신을 할 경우 각 회당 전송 간격을 설정하는 기능이다. 갱신회수는 앞에서 설정된 객체정보 업데이트를 반복하는 회수이다. 〈그림 3-6〉은 기 생성된 객체들을 1회 업데이트 할 때 개당 100번씩 업데이트 하고, 전송간격은 33ms, 갱신회수는 총 10회로 설정한 화면이다.



〈그림 3-6〉 반복 송신 기능

5) 송신/수신 로그화면

RTI를 통해 오가는 메시지의 내용 확인과 누락메시지가 있는지 여부, 송수신 한 시각을 확인하기 위해서는 별도의 로그창이 송신/수신 각각 필요하다. 로깅 방식은 두 가지 방식이 있다. 첫 번째는 <그림 3-7>의 송신화면처럼 메시지 내용을 전부 보여주는 방식으로써 송수신 내용을 확인하고 내부 속성데이터가 누락되거나 손실된 부분은 없는지 확인하는 방식이며, 두 번째는 수신화면처럼 다량의 메시지에 대한 송/수신 정보를 기록으로 나타낸 것이다. 여기에는 송/수신 시각, 메시지 종류, 객체의 ID, 반복 송신일 경우 총 회차 중 현재 회차를 나타내는 형식으로 보인다.

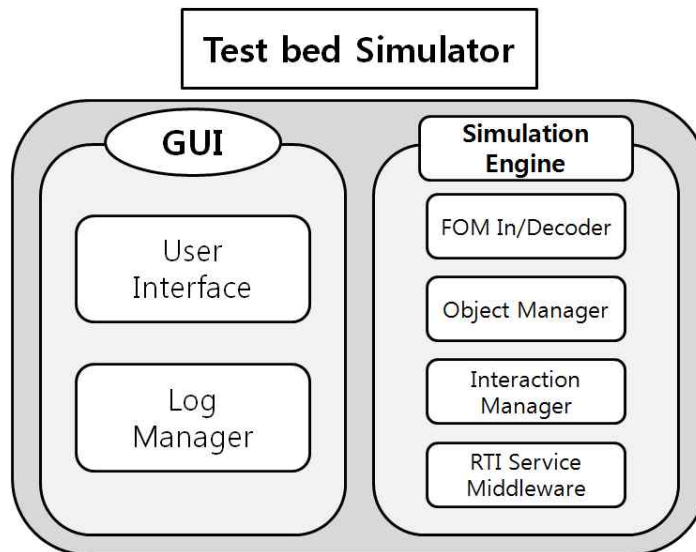
로깅 방식은 메시지 송/수신 전 반복송신 기능을 활성화 시키면 대량 메시지 확인 방식으로 자동 변경된다.

송신		수신			
<<STGroundVehicle>>		16:13:15.501	Human	1	Reflect 1 / 2
-AccelerationVector-		16:13:15.501	Human	2	Reflect 1 / 2
fAccelerationX : 1.000000		16:13:15.501	Human	3	Reflect 1 / 2
fAccelerationY : 2.000000		16:13:15.502	Human	4	Reflect 1 / 2
fAccelerationZ : 3.000000		16:13:15.502	Human	5	Reflect 1 / 2
-AngularVelocityVector-		16:13:15.502	Human	6	Reflect 1 / 2
fAngularVelocityX : 4.000000		16:13:15.503	Human	7	Reflect 1 / 2
fAngularVelocityY : 5.000000		16:13:15.503	Human	8	Reflect 1 / 2
fAngularVelocityZ : 6.000000		16:13:15.503	Human	9	Reflect 1 / 2
-DeadReckoningAlgorithm-		16:13:15.503	Human	10	Reflect 1 / 2
eDeadReckoningAlgorithm : 4		16:13:15.509	Human	1	Reflect 2 / 2
-EntityType-		16:13:15.510	Human	2	Reflect 2 / 2
enumEntityKind : 1		16:13:15.510	Human	3	Reflect 2 / 2
enumDomain : 1		16:13:15.510	Human	4	Reflect 2 / 2
enumCountryCode : 200		16:13:15.511	Human	5	Reflect 2 / 2
enumCategory : 0		16:13:15.511	Human	6	Reflect 2 / 2
enumSubcategory : 0		16:13:15.511	Human	7	Reflect 2 / 2
enumSpecific : 0		16:13:15.512	Human	8	Reflect 2 / 2
enumExtra : 0		16:13:15.512	Human	9	Reflect 2 / 2
-EntityIdentifier-		16:13:15.512	Human	10	Reflect 2 / 2
usSiteID : 1					
usApplicationID : 200					
usEntityNumber : 1					
-Orientation-					
fPsi : 30.000000					
fTheta : 60.000000					
fPhi : 90.000000					
-WorldLocation-					
dbIX : 123.456789					
dbIY : 234.567898					
dbIZ : 345.678987					

<그림 3-7> 송신 / 수신 로그화면

3.3.5 구현 결과

Test bed Simulator는 Visual Studio를 이용하여 C++ 개발환경 하에 MAK社에서 개발한 RTI의 라이브러리를 사용하는 형태로 개발되었으며 개발기간은 약 6개월이 소요되었다. 이 Test bed Simulator를 통해 HLA/RTI 기반 분산시뮬레이션 설계 시 고려해야 할 설계변수들을 추정하기 위한 다양한 시험이 가능하도록 기능을 구현하였다. 또한 사용자의 요구사항에 따라 원할히 수정/보완 할 수 있도록 GUI와 시뮬레이션 엔진을 독립적으로 분리시켜 구현하였다. GUI 부분은 사용자 인터페이스와 데이터 저장기능으로 구성하였으며 시뮬레이션 엔진 부분은 FOM 인/디코더, 객체 생성/삭제/관리모듈, 상호작용 메시지 모듈, RTI서비스 관리/이용 모듈로 구성하였다.



〈그림 3-8〉 Test bed Simulator 구성 개념도

IV. Test bed 운용 및 결과

4.1 HLA/RTI 기반 Test bed 운용절차

Test bed의 운용은 Test 준비, Test 실행, Test 결과분석의 3단계로 나뉜다.

1) Test 준비

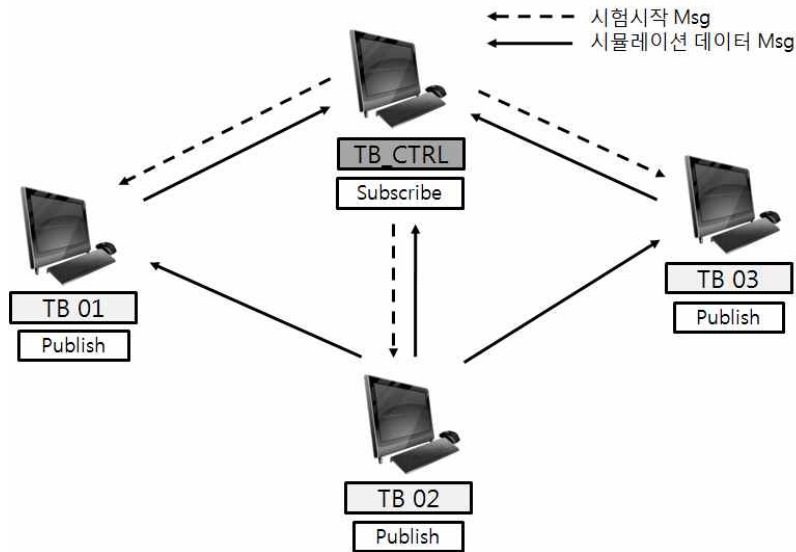
시뮬레이션 망 테스트를 하기 위해서는 먼저 시험하고자 하는 페더레이션의 구성을 결정하여야 한다. 몇 개의 페더레이트가 참여를 할 것인지, 각 페더레이트에서 모사 할 객체의 종류와 객체 수는 몇 개인지, 시뮬레이션 정보의 갱신 주기 시간은 얼마로 결정할 것인지 등이다.

Test 구성을 결정하였으면 결정에 따라 Test bed 페더레이션과 Test bed Simulator를 구성한다. 참여하는 페더레이트 수만큼 각 시뮬레이터에 Test bed Simulator를 실행하고 조건에 따라 객체를 미리 생성해둔다.

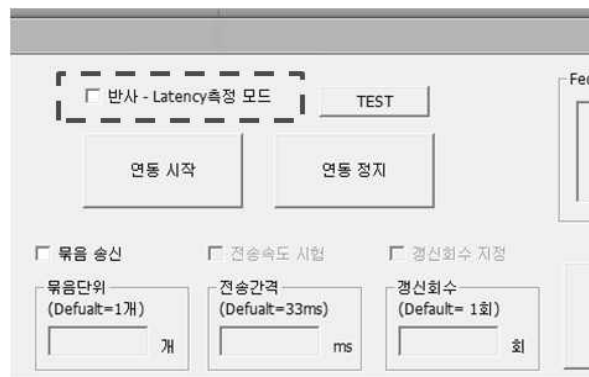
2) Test 실행

제대로 된 시험을 하려면 시뮬레이션 시작과 동시에 각 페더레이트에서 객체 정보를 전송해야만 한다. 실제 시뮬레이션 시스템이 아닌 Test bed로 이러한 상황을 재연해내기 위해 다음과 같은 방법을 사용한다.

〈그림 4-1〉와 같이 중앙에 Test bed를 통제하는 페더레이트와 네트워크 부하를 주기위한 객체정보생성 페더레이트로 페더레이션을 구성한다. Test bed Simulator에는 〈그림 4-2〉의 ‘반사-Latency 측정모드’ 기능이 있는데, 〈그림 4-1〉의 객체정보생성 페더레이트(TB_01 ~ 03)의 Test bed Simulator의 기능을 체크해주면 외부에서 Start/Resume 메시지를 받는 순간 ‘Object 갱신’ 기능이 동작하게 된다. 따라서 사전에 객체를 생성해두고 묵음송신 기능 등을 설정해 두면 〈그림 4-1〉의 TB_CTRL 페더레이트에서 Start/Resume 메시지를 받는 즉시 객체정보 송신이 시작된다.



〈그림 4-1〉 Test bed 시뮬레이션 망 테스트 개념도



〈그림 4-2〉 반사-Latency 측정모드 기능

객체정보생성 페더레이트가 객체정보를 송신하면 모든 데이터는 중앙의 Test bed 통제 페더레이트가 메시지를 수신하게 되고 자동으로 수신 로그 화면에 기록이 된다. 지정된 수량만큼 메시지가 모두 송신이 되고 중앙의 통제 페더레이트에서도 메시지 수신이 끝나면 로그 데이터를 저장하고 Test 실행을 마친다.

3) Test 결과 분석

수신된 로그의 분석을 통해 다음과 같은 정보를 확인할 수 있다.

- ① 시뮬레이션 데이터 갱신 주기 내에 모두 메시지를 수신했는지 여부
- ② 갱신 주기 내 최대로 받을 수 있는 메시지 수량 확인
- ③ 각 메시지 수신간격 확인
- ④ 각 메시지 데이터 정보내용 확인
- ⑤ 각 메시지 수신 순서 확인
- ⑥ 누락메시지 존재 여부 확인 등

4.2 HLA/RTI 기반 Test bed 적용결과

시험을 수행할 Test bed Simulator를 실행하는 시스템에서 앞으로 HLA/RTI기반 분산시뮬레이션 M&S도구를 운용한다고 가정하였을 때, 실제 M&S도구를 개발하기 전 본 시스템의 네트워크 성능한계가 어느 정도인지 미리 확인할 수 있도록 Test bed Simulator를 통해 성능을 시험하고 그 결과를 분석하였다.

실험을 진행하기에 앞서, 먼저 N개의 페더레이트가 연합하는 시뮬레이션의 경우 수신되는 메시지의 양은 시뮬레이션 엔진이 처리할 수 있는 단위시간당 처리량보다 항상 많다고 가정한다. 다시 말해 Test bed의 메시지 수신모듈은 쉬지 않고 언제나 100%의 성능을 발휘한다고 가정한다. 실험환경은 구성 여건 상 1대의 PC에서 두 개의 페더레이트를 운용하는 페더레이션으로 구성하였다.

수행시험은 Test bed Simulator의 기능을 활용하여 수행할 수 있는 시험으로, 대량 객체 속성정보 동시 송수신, 1개 객체 속성정보 대량 송수신, 주기별 객체정보 반복 송수신 시험이다. 위 시험을 실시한 후 결과분석을 통해 본 시스템의 시뮬레이션 메시지 송수신 능력을 확인한다.

4.2.1 대량 객체 속성정보 동시 송수신

1) 시험 목표

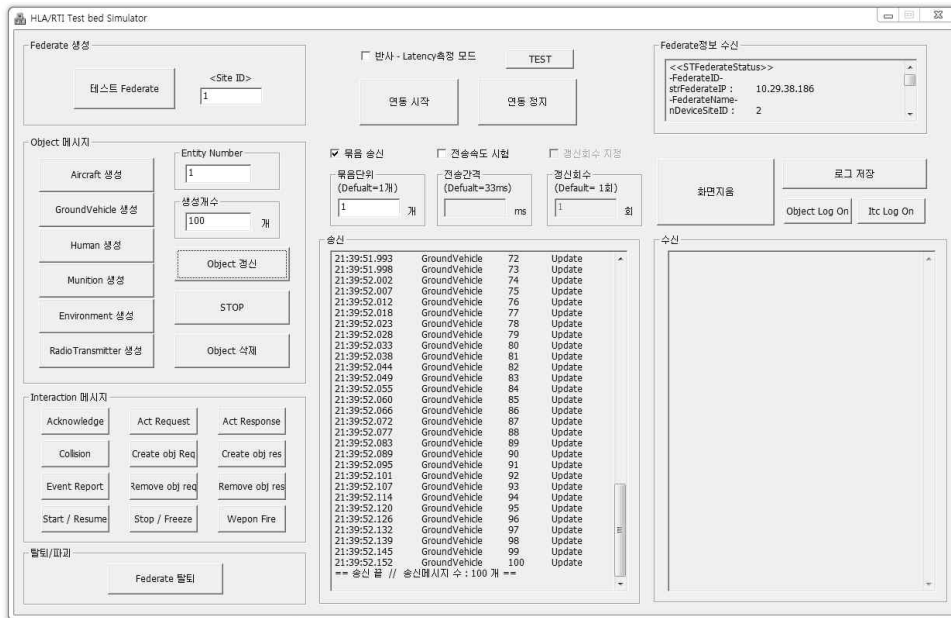
첫 번째 시험은 대량의 객체를 생성하여 객체 속성정보를 송수신 하는 시험이다. 다수의 객체를 모의하는 시뮬레이션의 상황을 가정한 것으로 100여 개의 객체를 생성하여 모든 객체의 정보를 1회 갱신하는데 걸리는 시간을 측정하는 것이 시험의 목표이다.

2) 시험 방법

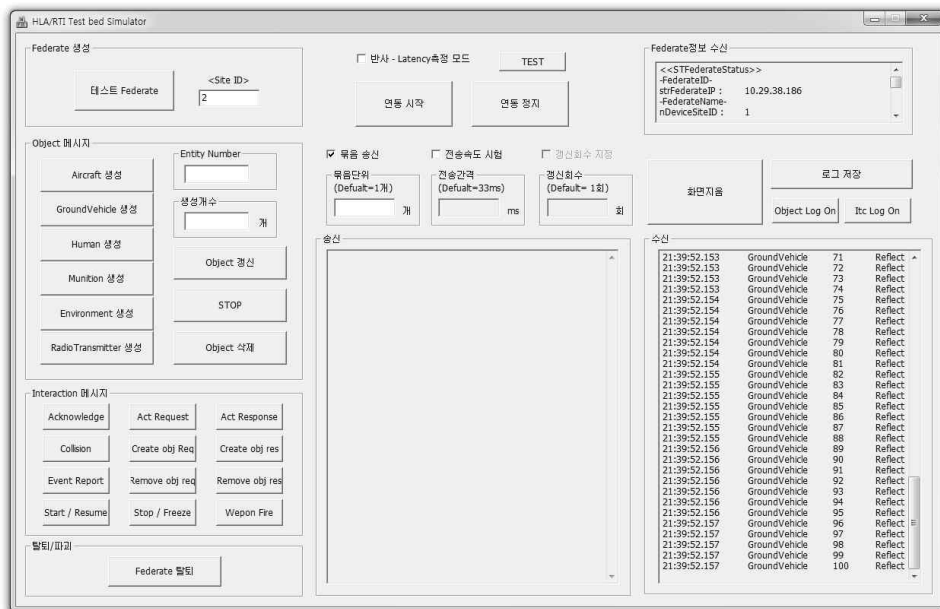
데이터 송신측 Test bed Simulator에서 Ground Vehicle 객체 100개를 생성한 후 이 객체들의 속성정보를 1회 Update 한다. 수신측 Test bed Simulator의 로그 창에서 데이터가 수신되는 것을 확인한다.

3) 시험 결과 및 분석

송신측에서는 404ms만에 송신을 완료하였고 수신측에서는 407ms 만에 수신을 완료하였다. 즉 객체 1개의 1회 상태정보 갱신에 약 4ms 의 시간이 걸린 것을 의미한다. 송신완료와 수신완료에 걸리는 시간의 차이가 1%내외인 것은 거의 차이가 없다는 것인데 이는 현실과는 조금 거리가 있는 시험결과이다. 보통은 메시지 디코딩 속도가 훨씬 느리기 때문에 모델이 시뮬레이터 엔진에게 데이터를 전달하는 속도가 충분히 빠르다면 데이터 수신을 완료하는데 걸리는 시간이 더 길다. 또한 현재 데이터를 갱신하는 속도가 실제 경험했던 사례보다 보다 많이 느리게 나오는데 그 이유는 사실 위 시험을 송신측 페더레이트와 수신측 페더레이트를 여건상 1대의 PC에서 수행하여 시스템에 부하가 가해졌기 때문이다.



〈그림 4-3〉 100개의 객체 상태갱신정보 송신화면



〈그림 4-4〉 100개의 객체 상태갱신정보 수신화면

4.2.2 1개 객체 속성정보 대량 송수신

1) 시험 목표

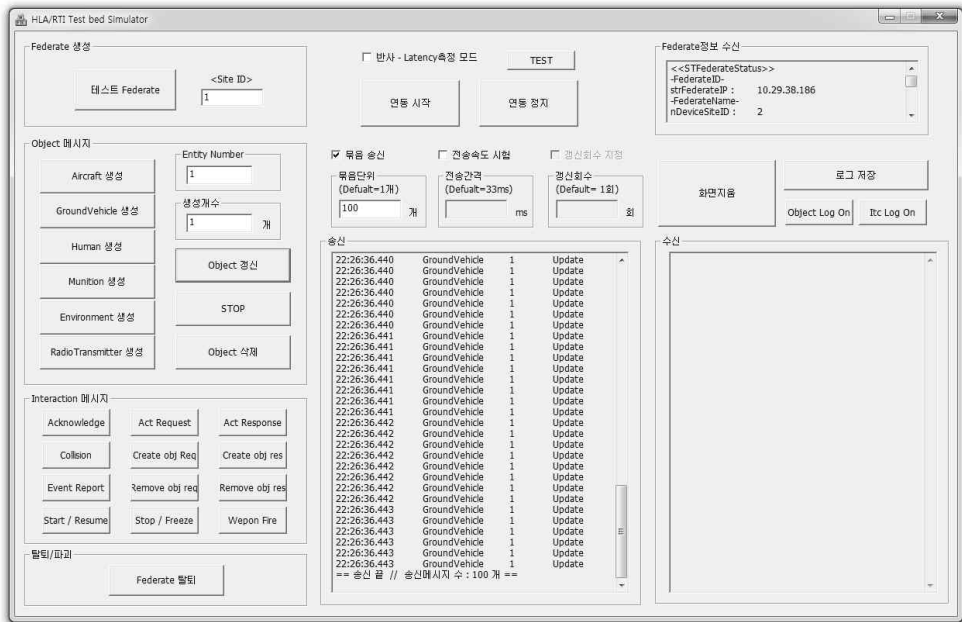
두 번째 시험은 1개의 객체를 생성하여 100회 송수신 하는 시험이다. 좀 더 현실적인 메시지 수신환경을 만들기 위하여 최대한 메시지 송신 속도를 증가시킬 수 있도록 객체 핸들링에 대한 부하를 줄이고 빠르게 반복갱신하기 위해 위와 같은 조건으로 시험을 실시한다.

2) 시험 방법

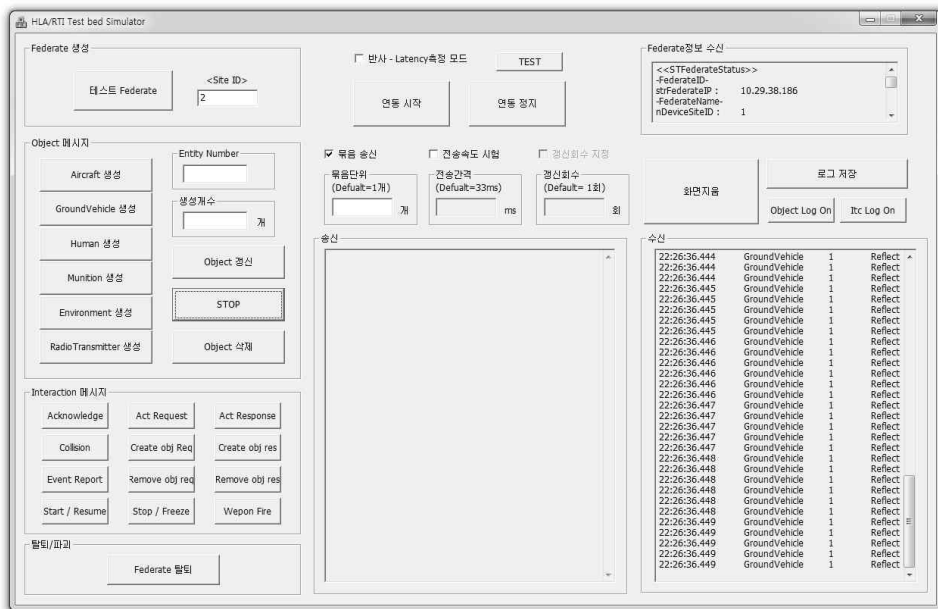
데이터 송신측 Test bed Simulator에서 Ground Vehicle 객체 1개를 생성한 후 묶음송신 기능에 100개를 입력하고 설정한다. Object 갱신을 시작하여 수신측 Test bed Simulator의 로그 창에서 1개의 객체 정보가 100회 수신되는 것을 확인한다.

3) 시험 결과 및 분석

시험 결과, 100여회의 객체상태정보 갱신 메시지 송신하는데 걸린 시간은 약 12ms였고 100여회의 객체상태정보 갱신 메시지를 수신하는데 걸린 시간은 약 18ms였다. 즉 객체정보 1개 수신 당 0.18ms 가 걸린 것이다. 데이터를 수신을 마치는데 걸리는 시간이 송신을 완료하기 걸린 시간보다 약 50% 더 오래 걸렸고 개당 메시지 수신시간은 실제 경험적 사례와 비슷하거나 조금 더 짧게 나온 것으로 보아 시험은 정상적으로 수행된 것으로 판단하였다. 또한 위 실험결과를 통하여 100개의 객체를 1회 갱신하는 것 보다 1개의 객체를 100회 갱신하는 데 걸리는 시간이 더 오래 걸린다는 사실도 알 수 있다.



〈그림 4-5〉 1개 객체 상태갱신정보 100회 송신화면



〈그림 4-6〉 1개 객체 상태갱신정보 100회 수신화면

4.2.3 주기별 객체정보 반복 송수신

1) 시험 목표

세 번째 시험은 주기별로 객체정보를 반복 송수신 하는 시험이다. 시뮬레이션에서는 일정 시간마다 시간을 끊어서 객체정보를 처리하는 경우가 있다. 이러한 경우, 정해진 시간 주기 내에 정보가 수신되어야 시뮬레이터에서 시간지연을 시키지 않고 실시간으로 데이터를 처리할 수 있기 때문에 정해진 시간 주기 내 모든 데이터 송수신 여부는 중요하다. 시험 조건은 매 33ms 마다 30여회의 객체상태갱신 메시지를 3회 송신하여 송신/수신 양측 모두 정상적으로 데이터를 주고받는지 확인한다.

2) 시험 방법

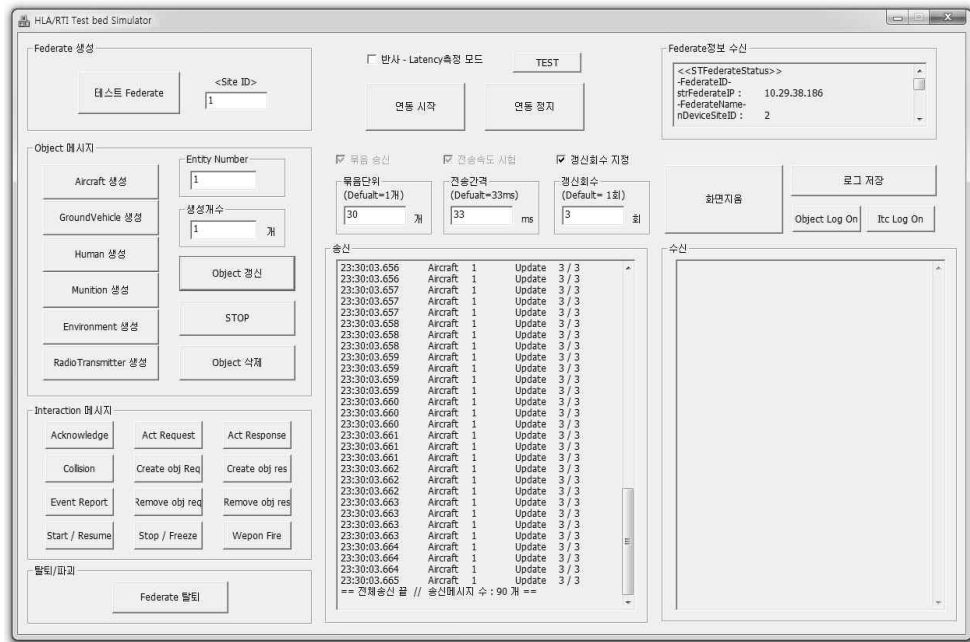
데이터 송신측 Test bed Simulator에서 Ground Vehicle 객체 30개를 생성한 후 묶음송신 기능에 30개, 전송간격에 33ms, 갱신회수에 3회를 입력하고 설정한다. Object 갱신을 시작하여 수신측 Test bed Simulator의 로그 창에서 30개의 객체 정보가 33ms 주기로 3회 수신되는 것을 확인한다.

3) 시험 결과 및 분석

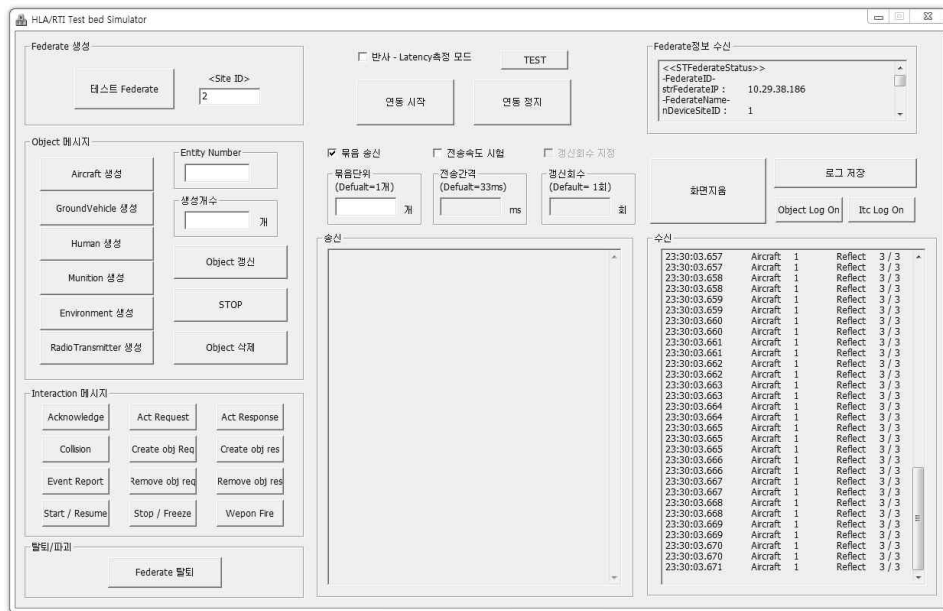
시험 결과, [표 4-3]과 같이 데이터가 송/수신 되었다. 송신측과 수신측 모두 데이터를 주고받는데 33ms 이내로 소요되었다. 데이터를 수신 처리하는데 걸리는 시간은 송신하는데 걸리는 시간보다 약 50% 정도 더 필요하다는 것을 알 수 있다.

[표 4-1] 33ms 주기로 객체정보 30회 갱신 3회 실시 결과

송신		수신	
회차	소요 시간	회차	소요 시간
1	9ms	1	14ms
2	9ms	2	14ms
3	9ms	3	14ms



〈그림 4-7〉 33ms 주기로 객체정보 30회 갱신 3회 송신화면



〈그림 4-8〉 33ms 주기로 객체정보 30회 갱신 3회 수신화면

4.2.4 시험결과 분석

시험환경의 제약조건 상 1대의 PC로 구성된 시뮬레이션 네트워크 부하량 테스트를 실시할 수밖에 없었으나 시뮬레이션의 구성환경에 가장 영향을 받지 않는 시험인 2번 시험을 기준으로 시험결과를 분석한다.

위 Test bed Simulator에 적용된 형태의 FOM을 사용하는 시뮬레이션인 경우에는 1개의 객체 정보를 수신하는데 대략 0.18ms의 시간이 필요한 것으로 확인되었다. 앞으로 개발될 M&S도구의 시뮬레이션의 시간 주기에 대한 요구사항이 1초일 경우, 시뮬레이션 시간의 지연 없이 최대 모사 가능한 객체의 수는 $5,555$ 개($1000 \div 0.18$)가 될 것이다. 여기에 10% 정도 성능의 마진을 둔다면 대략 5,000여개의 객체를 위 시스템 환경에서 동시에 표현 가능하다는 결론을 도출할 수 있다.

이러한 사실은 Test bed를 구성하여 미리 시험하지 않고서는 M&S도구 개발을 시작하여 단위/통합시험이 가능해 지는 단계까지 진행되지 않는 한 알기가 사실상 어렵다. 반대로 말하면, Test bed를 이용하여 미리 시험을 수행하면 요구사항 결정 단계부터 먼저 파악하고 설계를 변경한다든가 다른 대안을 미리 모색할 수 있어서 M&S도구 개발의 위험을 줄이고 이에 따르는 시간과 비용의 절감 역시 기대 할 수 있게 된다.

1번 시험과 3번 시험의 Case에서도 확인하였듯이 HLA/RTI 기반 Test bed를 이용하면 실제 모델과 시뮬레이션 엔진, 다수의 시뮬레이터로 시뮬레이션 망을 구성하지 않고도 RTI 연동망을 구성, 원하는 객체를 원하는 수량만큼 생성하여 객체상태갱신 메시지를 원하는 수량과 주기, 갱신 회수로 네트워크에 부하를 임의로 가하면서 시험이 가능 한 것을 알 수 있다. 객체의 수, 갱신주기, 데이터양과 같은 S/W에 기반을 둔 변수 뿐 아니라, 물리적 네트워크의 상태, 시뮬레이터가 실행되는 서버 등 H/W에 따라서도 RTI의 연동성능이 달라지기 때문에 위와 같은 Test bed가 없이는 시뮬레이션 망이 어느 정도의 수준까지 모의를 지원해주는지 가늠하기 어렵다는 점에서 본 Test bed의 실용성에 대한 의미를 찾을 수 있다.

V. 결론 및 향후 연구방향

국방 M&S 도구는 과거 Stand Alone 형태에서 현재는 분산 컴퓨팅 기술을 활용한 분산시뮬레이션으로 발전되어왔으며, 시뮬레이션과 시뮬레이션 간의 연동은 HLA/RTI라는 표준기술구조가 존재한 덕분에 가능하게 되었다.

물리적으로 분리되어 있는 시스템 간의 정보 교환을 기반에 두고 있는 시뮬레이션의 특성 상, 오가는 데이터가 많아지면 많아질수록 시뮬레이션의 진행에 영향을 주는 것은 사실이다. 본래 시뮬레이션의 세계에서는 시뮬레이션 시간과 실제 시간과의 직접적인 연관성은 없지만, 사람이 실제 시뮬레이션에 참여하는 가상 시뮬레이션의 경우, 시뮬레이션 시간의 진행속도가 실제 시간속도와 같아야만 하기 때문에 시뮬레이션의 지연발생은 정상적인 시뮬레이션 수행에 큰 위험요소로 작용한다.

시뮬레이션 망의 네트워크 부하량은 기존의 경우 모델과 시뮬레이션 엔진이 어느 정도 개발이 되고난 후에야 시험이 가능하나, 만약의 경우 네트워크가 시뮬레이션의 데이터 트래픽을 감당하지 못하는 경우, 이를 해결하기 위해 시간과 비용이 많이 투입되게 된다. 이를 미리 검증해보고 확인하여 대처할 수 있는 대안으로 본 연구논문의 주제인 Test bed를 제안하였다.

Test bed 를 구성하는 Test bed Simulator는 RTI의 서비스를 이용하여 사용자가 간단하게 시뮬레이션 네트워크의 성능한계점을 확인해 볼 수 있도록 구현되었다. 하지만 짧은 시간 내에 설계/구현을 하기 위해서 포함하지 못한 기능들도 있었다.

HLA/RTI 존재의 중요가치 중 하나는 바로 ‘상호운용성’이다. HLA/RTI는 개발에 서로 전혀 관여하지 않은 두 시뮬레이션이 HLA에서 제시하는 규칙을 지키고 시뮬레이션에서 사용하는 객체/상호작용 정보만 일치한다면 서로 연동이 되는 것을 보장해 준다.

Test bed는 목적의 특성 상 불특정 다수의 시뮬레이션 망에 대한 시험지원

이 가능해야 하기 때문에 어떠한 형태의 FOM이 되어도 그것에 대한 시뮬레이션 망 테스트가 가능해야 하나 현재는 RPR-FOM 만 적용하고 있으며 그 외의 사용자 정의 FOM은 아직 지원하지 않고 있다. 추후에는 FDD문서만 입력시키면 이를 읽고 해당 시뮬레이션에서 사용하는 객체/상호작용 메시지로 테스트가 가능하게 해야 하겠다.

또한 Test bed Simulator의 송수신 로그 전시기능을 활용하여 개발 중에 있는 시뮬레이션들의 송수신 검증 테스트 도구로써도 활용발전방안을 찾을 수 있을 것이다.

모쪼록 본 연구와 Test bed Simulator가 좀 더 다듬어지고 발전하여 국방 M&S도구 개발에 도움이 될 수 있기를 희망한다.

참 고 문 헌

1. 국내문헌

- 공군대학. (2000). 『위게임 개론』. 대전: 공군대학.
- 국방기술품질원. (2011). 『국방과학기술용어사전』 서울: 국방기술품질원.
- 김성용 외. (2010). 『RTI와 전술데이터링크 간 상호연동 시뮬레이터 설계』. 한국통신학회.
- 김종만. (2006). 『국방 M&S 개론』. 서울: 국방기술품질원.
- 김탁곤. (2012). 『국방M&S이론 및 기술(II)』. 대전: KAIST.
- 김형현. (2009). 『국방M&S개론』. 서울: 경성문화사
- 김현철 외. (2010). 지상무기체계 효과도 분석을 위한 RTI기반의 연동인터페이스 설계 및 미들웨어의 적용. 『한국시뮬레이션학회』. 1-5.
- 민계료 외. (1993). 『OR의 군사적 응용』. 서울: 국방대학교.
- 배성한. (2006). 『무기체계 획득단계별 M&S 활용방안』. 서울: 국방대학교.
- 윤석준. (1997). 시뮬레이션 산업 현황과 나아갈 길. 『항공우주』.
- 장상철 외. (2001). 『차세대 시뮬레이션 연동체계 기술연구』. 서울: 국방연구원
- 정현욱. (2002). 『국방 모델링 및 시뮬레이션(M&S) 발전방안 연구』. 대전: 공군대학.
- 진정훈 외. (2007). 분산 시스템 환경을 위한 미들웨어 성능분석. 『지상무기 학술대회』, 15, 1-4.
- 최상영. (1999). 『국방모델링 및 시뮬레이션 I』. 서울: 국방대학교.
- 최상영. (2001). 『국방시뮬레이션 고수준 아키텍처 개론』. 서울: 국방대학교.
- 최상영. (2006). 『국방모델링 및 시뮬레이션(모델링 이론과 실제)』. 서울: 국방대학교.

2. 국외문헌

- IEEE (2000). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules* : IEEE.
- IEEE. (2000). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specification* : IEEE.
- IEEE. (2000). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification* : IEEE.
- Lalit K. Piplani. (1994). *Systems Acquisition Manager's Guide for the use of Models and Simulations* : Defense Systems management college press.
- SISO. (1999). *RPR-FOM Version 1.0 SISO-STD-001.1-1999*. : SISO.

ABSTRACT

An Empirical Study for the Implementation of Test bed based on HLA/RTI : Focused on Distributed Simulation Development and Implementation

Park, Hee-Jun

Major in National Defense Modeling & Simulation

Dept. of National Defense Modeling & Simulation

Graduate School of National Defense Science

Hansung University

U.S. Department of Defense's standard simulation architecture, HLA/RTI used in the world's military M&S for distributed simulation system. Defense M&S in Korea also HLA/RTI technology widely adopted to simulate the linkage between simulations.

The implementation and execution of distributed RTI simulation for one of a many of services provided to the user so that the system resources are used , and the message will be strong enough to transfer the load on the data transmission performance is significantly drop. Therefore, in order to reduce the risk of the development process before implementing the simulation, should be consider determining data transmission and network's capability.

In this research, identify key point when develop distributed simulation based on HLA/RTI, Example th case that development practices on the issues, suggest the Test bed based on HLA/RTI for proper simulation design.

【Keyword】 M&S, HLA, RTI, Test bed, Modeling, Simulation