



저작자표시-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위논문

COTS기반의 V-C연동 향상 방안 연구



2013년

HANSUNG
UNIVERSITY

한성대학교 국방과학대학원

국방 M&S 학과

국방 M&S 전공

김 경 수

석사학위논문
지도교수 이동준

COTS기반의 V-C연동 향상 방안 연구

A Study on COTS Based Interoperability Architecture
for Virtual and Constructive Simulation

2012년 12월 일

HANSUNG
UNIVERSITY

한성대학교 국방과학대학원

국방 M&S 학과

국방 M&S 전공

김 경 수

석사학위논문
지도교수 이동준

COTS기반의 V-C연동 향상 방안 연구

A Study on COTS Based Interoperability Architecture
for Virtual and Constructive Simulation

위 논문을 국방 M&S학 석사학위 논문으로 제출함

2012년 12월 일

한성대학교 국방과학대학원

국방 M&S 학과

국방 M&S 전공

김 경 수

김경수의 국방M&S학 석사학위논문을 인준함

2012년 12월 일



심사위원장 _____인

심 사 위 원 _____인

심 사 위 원 _____인

국 문 초 록

COTS기반의 V-C연동 향상 방안 연구

한성대학교 국방과학대학원

국방 M&S 학과

국방 M&S 전공

김 경 수

국방 M&S를 통해 가상 및 위게임 시뮬레이션을 활용한 다양한 군사훈련을 가능하게 했고, 최근에는 이에 한걸음 더 나아가 기존의 개별적으로 수행되던 야전, 가상 및 위게임 훈련을 통합하는 LVC 통합훈련 환경으로 발전하고 있는 추세이다.

그러나 국내의 LVC 체계 연동은 선진국 기술에 대비해 많은 취약점을 가지고 있다. 따라서 선진국의 COTS를 기반으로 국내 LVC체계 구축에 활용하면 우수한 기술을 빠르게 접목하면서 우리의 기술을 발전시킬 수 있을 것이다.

이에 본 논문은 LVC 체계 연동 중 VC의 연동방안에 대해 COTS와 HLA Evolved를 적용한 방법론을 제시한다. 또한 COTS를 활용한 구축 사례를 소개함으로써 VC연동에 관한 이론적인 방법론의 제시뿐만 아니라 이에 따른 실제 적용에 관한 실효성을 나타낸다.

COTS를 활용하면 V-C연동을 위한 인원 및 기간의 감소 효과가 기대되며, V-C연동체계의 안정성과 신뢰성의 향상을 가져 올 것으로 기대한다. 본문에서 제시한 V-C 연동 아키텍처를 구축 시 데이터의 재사용성, 확장성, 안전성 및 신뢰성의 증진을 가져 올 것으로 예상된다.

【주요어】 LVC연동, V-C연동, HLA, HLA Evolved, COTS

목 차

제 1 장 서 론	1
제 1 절 연구의 목적	1
제 2 절 연구의 범위와 방법	2
제 2 장 현 V-C 연동의 이론적 고찰	4
제 1 절 V-C 연동의 필요성	4
제 2 절 국내외 구축현황	6
1. 신궁	7
2. 나토의 EFW	8
3. 영국의 MTDS	9
제 3 절 V-C 연동의 문제점	11
제 3 장 HLA Evolved와 COTS 기반의 V-C 연동	14
제 1 절 현재 V-C 연동 아키텍처	14
1. 제어 모듈	16
2. 이기종간의 아키텍처 연동	18
3. 공급자가 서로 다른 RTI 연동	20
4. 갱신 주기율 동기화	21
제 2 절 향상된 V-C 연동 아키텍처 구성	23
1. Font-End	24
2. Back-End	36
제 3 절 현 V-C 체계와 비교분석	43

제 4 절 COTS기반 연동의 한계점	45
----------------------------	----

제 4 장 COTS 기반의 V-C 연동구축 사례	46
---	-----------

제 1 절 V-C 연동개요	46
----------------------	----

제 2 절 V-C 연동구축	47
----------------------	----

1. Constructive 모델	48
--------------------------	----

2. Virtual 모델	50
---------------------	----

3. 연동 모델	51
----------------	----

4. Server	53
-----------------	----

제 3 절 V-C 구축결론	54
----------------------	----

제 5 장 결 론	56
------------------------	-----------

【참고문헌】	58
---------------------	-----------

【 표 목 차 】

[표 1] LVC 특징	2
[표 2] V-C 연동 시 나타나는 문제점	12
[표 3] COTS와 HLA Evolved 활용	26
[표 4] 새로운 HLA의 기능별 카타고리	28
[표 5] HLA Evolved의 기능	29
[표 6] 현 V-C체계와 비교분석	44



【 그림 목 차 】

<그림 1> LVC 훈련체계 활용사례	5
<그림 2> 신궁 구성도	7
<그림 3> MTDS의 구성 체계	10
<그림 4> 현재 V-C 연동 아키텍처	15
<그림 5> 연동 Adaptor의 구조	16
<그림 6> 제어 모듈	17
<그림 7> 이기종간 V-C 연동의 아키텍처	18
<그림 8> 이기종간 연동을 위한 Adaptor 구조	19
<그림 9> 공급자가 서로 다른 RTI의 Adaptor	20
<그림 10> 공급자가 다른 RTI 연동 모듈	21
<그림 11> 갱신 주기율 모듈	22
<그림 12> 향상된 V-C 연동 방안	24
<그림 13> Front-End 단 아키텍처	25
<그림 14> VR-Exchange 아키텍처	27
<그림 15> Translator의 예	27
<그림 16> FOM 모듈 예시	31
<그림 17> WSDL의 코드 생성	32
<그림 18> WSPRC를 통한 RTI 접속	32
<그림 19> Fault Tolerance의 의미	33
<그림 20> SURR example	34
<그림 21> Evolved Dynamic Link Compatible	35
<그림 22> Back-End 아키텍처	36
<그림 23> VR-TheWorld	39
<그림 24> VR-TheWorld를 활용한 예	39
<그림 25> Entity Editor	41
<그림 26> 객체의 표시의 예시	42
<그림 27> V-C 연동 아키텍처	47

<그림 28> VR-Force 상황도	48
<그림 29> Plan과 Task 부여	49
<그림 30> IR 맵핑 처리 된 DB	50
<그림 31> Virtual 모델의 운용	51
<그림 32> MAK RTI	52
<그림 33> VR-Exchange	53
<그림 34> VR-TheWorld Server	54



제 1 장 서 론

제 1 절 연구의 목적

현 국방 체계에서는 무기의 첨단화와 무기의 다양성 전술의 발달로 인해 실제 훈련체계에서 이를 반영하기에 한계점이 있다. 이러한 한계점을 극복하기 위해서 국방 M&S(Modeling & Simulation)가 대두 되었고 이는 첨단 기술의 발전으로 인해 많은 발전을 이루었다. 국방 M&S를 통해 가상 및 위게임 시뮬레이션을 활용한 다양한 군사훈련을 가능하게 했고, 최근에는 이에 한걸음 더 나아가 기존의 개별적으로 수행되던 야전, 가상 및 위게임 훈련을 통합하는 LVC 통합훈련 환경으로 발전하고 있는 추세이다.

LVC 통합 모의 기술은 해외 여러 국가에서 다양한 형태의 LVC 통합 환경을 구축하여 훈련에 활용중이며 이는 모의 모델이나 기술들을 조립하여 다양한 모의가 가능하도록 발전중이다. 일반적으로 모의 합성전장을 구성하는 체계는 Constructive, Virtual, Live 시뮬레이션으로 나뉜다.

Constructive 시뮬레이션은 가상 상황 하에서 가상의 병력으로 시뮬레이션 하는 것이고, Virtual 시뮬레이션은 실제 장비 내부와 동일한 모의 환경을 구성하고 운용자가 가상 상황 하에서 운용을 할 수 있도록 하는 시뮬레이션이다. Live 시뮬레이션은 실 훈련으로써 실 병력이 실제지형 하에서 직접 기동 하면서 시뮬레이션 하는 것을 말한다.¹⁾

선진국에 비해 현 우리 LVC체계 연동은 많은 취약점을 가지고 있다. 이에 앞으로 많은 연구가 필요하고, 이를 활용해 국내 LVC체계 구축이 필요 하다. 이번 논문을 현재 국내 VC연동에 관한 현 주소와 문제점을 나타내고 국외선진 사례를 소개한다.

본 논문은 선진 COTS와 HLA Evolved를 활용한 V-C 체계연동방안을 구축 제시함으로써, COTS와 HLA Evolved를 통해 V-C 연동의 향상 방안의 가능성을 제시 하는데 있다.

1) 윤근호 외(2012), “Battle lab에서의 상호 운용성을 위한 LVC 연동방안”, 『한국시뮬레이션학회논문지』, 한국시뮬레이션학회, p.82

제 2 절 연구의 범위와 방법

과학화훈련을 체계적으로 구축하고 있는 선진국 미군에서는 과거 STOW (Synthetic Theater of War : 인공 가상합성전장체계)의 개념을 지속적으로 발전 시킨 LVC체계 Live, Virtual, Constructive 체계 중 2개 이상의 체계를 연동하여 훈련하는 체계를 부분적으로 완성시켜가고 있으며 이러한 LVC 체계를 군 과학화훈련의 최종적인 모습으로 개념을 세우고 추진 중에 있다.²⁾

LVC 체계의 특징은 [표1]과 같다.

[표1] LVC의 특징³⁾

구분	구성(Constructive) 시뮬레이션	가상(Virtual) 시뮬레이션	실 기동(Live) 시뮬레이션	
특징	• 실험을 위한 다양한 시나리오 구성이 용이 • 다양한 모의 수준에서 다양한 성능/효과 지수의 신속한 측정 • 전장 제 기능(협동전, 합동전) 실전적 모의(대부대 훈련가능) • 개략, 상세모사를 병행 모의하여 다양한 제대를 동시에 모의 • 운용 인원 최소화 가능/비용 저감 • 실전체험 제한(전술훈련 곤란)	• 실제장비와 유사한 물리적 환경에서 장비 조작/숙달 훈련 • 운용자의 육체적/정신적 능력 반영 • 안전한 장비 조작 훈련 • 자원제약 없이 주요 전투장비에 대한 승무원 및 팀 전술 훈련 실시(대부대 훈련 제한) • 운용비용 절감/개발비용 소모 • 장소 제약 없이 반복훈련 가능	• 실시간 실행으로 인한 시나리오 반복 및 재연 불가 • 실험환경의 시간적/공간적 제약 및 안전성/경제성 문제(시간, 비용 과다) • 실 전투와 유사 • 연습 운용 시 장소 제약/비용 과다 소모/환경파괴 영향/대내외적 압박(대부대 훈련 제한)	
적용 방법	전장	모의 전장	모의 전장	실 전장
	병력	모의 병력	실 병력	실 병력
	장비	모의 장비	모의 장비	실 장비
활용 방안	훈련용	다양한 모의 수준에서 훈련	장비/시스템 숙달	실 훈련
	전투실험/분석용	• 전력구조(기술 인프라, 전술, 구조, 교리) 제안 및 분석 • LV(Live-Virtual) 시뮬레이션용 시나리오 발생/모의실험	• 운용자의 실시간 의사결정 능력 • 장비의 기능 및 성능 측정	• VC 시뮬레이션에서 사용되는 변수에 대한 원천 자료 제공 • VC 시뮬레이션 모델 실증(Validation)

2) 김성호(2010), “Virtual-Constructive 시뮬레이션 연동체계 구축 방안에 관한 연구”, 배재대학교 석사 논문, p.1.

3) 김숙영 외(2010) 『Virtual-Constructive 시뮬레이션 연동 기술에 관한 연구』 한국 군사과학기술학회 종합학술대회, pp.2022-2025.

Constructive 시뮬레이션은 다양한 상황묘사를 통한 시뮬레이션이 가능하나 현실감이 떨어진다는 단점이 있으며 Virtual 시뮬레이션은 실기동이 어려운 장비를 적은 비용과 제한된 공간에서도 시뮬레이션을 실시할 수 있다는 장점이 있으나 내부에 내장된 시나리오만이 반복되어 다양한 숙달이 어렵다는 단점이 있다. Live 시뮬레이션은 실 상황과 같은 지형과 환경에서 실 병력이 연동하여 현실감을 높일 수 있으나 시간 및 비용이 많이 들어 여러 번 운용을 반복하는 것이 제한적이라는 단점이 있다. 이 세 가지 시뮬레이션을 통합하여 합성 전장 환경에서 분산 모의운용을 하게 된다면 서로의 장단점이 보완되어 네트워크 중심 동시/통합 전투 수행능력이 확충되고 실 전장 환경 하에서 저 비용 / 고효율의 시뮬레이션이 가능할 것이다. 또한 사이버 공간을 활용한 운용 여건을 보장하여 공간의 제약을 벗어날 수 있다. 이러한 장점 때문에 군에서는 LVC 연동 시뮬레이션을 추진하고 있는 것이다.⁴⁾

본 논문에서의 범위는 이중 Live를 제외한 Virtual-Constructive의 연동에 초점을 맞추었으며, 전 세계적으로 널리 이용되는 선진 COTS를 기반으로 실제 V-C연동 체계를 구축 한다.

본 논문은 서론에서는 LVC체계의 소개와 필요 이유를 기술 한다. 2장에서는 V-C체계의 사례의 소개와 국내 V-C체계의 현 주소 및 연동의 문제점에 대해 기술 한다. 3장에서는 이러한 문제점을 바탕으로 RPR-FOM과 HLA Evolved를 이용한 아키텍처를 제시 한다. 4장에서는 아키텍처를 기반으로 COTS를 활용하여 실제 구축을 한다. 활용 될 COTS는 각각 교전급의 Constructive모델, 연동 모델, Virtual 모델, 지형 DB, 통신 모델, 분석 모델로 구성되며, 이를 구축 후 현 V-C 연동 체계와 비교 분석을 한다. 이를 토대로 5장에서 결론을 내리는 순으로 기술 한다.

4) 윤근호 외 전계논문, p.83.

제 2 장 현 V-C 연동의 이론적 고찰

V-C 훈련 체계란, 물리적으로 서로 이격된 군 과학화 훈련의 구성요소인 실 기동(Live), 가상 시뮬레이션(Virtual Simulation), 구성 시뮬레이션(Constructive) 중 가상 시뮬레이션과 구성 시뮬레이션을 연동하는 훈련 체계를 말한다.

예를 들어, 국내에서 개발된 전구급 위 게임 모델인 창조21(Constructive)와 현재 각 군이 보유하는 또는 개발 중인 시뮬레이터(Virtual)들의 연동을 V-C 연동이라 한다. 여기서 연동이란, 서로 다른 정보 체계 간에 발생하는 자료를 정해진 규약을 기준으로 상호 공유하는 것을 의미한다.

이번 장에서는 V-C 연동의 사례와 그 연동에 따른 문제점을 연구하여 기술 한다.

제 1 절 V-C의 연동 필요성

현재 우리 군 뿐만 아니라 다른 여러 나라의 군 훈련체계는 많은 부분에 대해 한계점을 보이고 있다. 이런 한계점은 크게 환경적 요인, 경제성, 위험성으로 나눈다.

환경적 요인으로는 실제 각 군의 대규모 훈련 시 그에 따른 훈련장 확보가 어렵다는 것이다. 각 군의 대규모 훈련장을 확보하기도 어려운 상황에서 실제 육. 해. 공군의 통합 훈련의 감행 시 훈련장 확보는 한계점이 있다는 것이 첫 번째 이유이다. 두 번째는 군의 다양한 전술 증가로 인한 작전체계가 상당히 복잡해 졌다는 것이다. 이러한 증가된 전술을 실제 수행 한다면 앞서 언급한 장소의 문제뿐만 아니라 전술 수행에 따른 시간이 상당히 소요 되고 다양하고 복잡한 전술을 실제 전부 수행하기란 한계점이 있다. 세 번째는 무기체계의 발달이다. 과거 무기체계에 비해 현대 무기체계는 파괴력이 높고, 미치는 범위가 현저하게 넓다. 이러한 무기체계

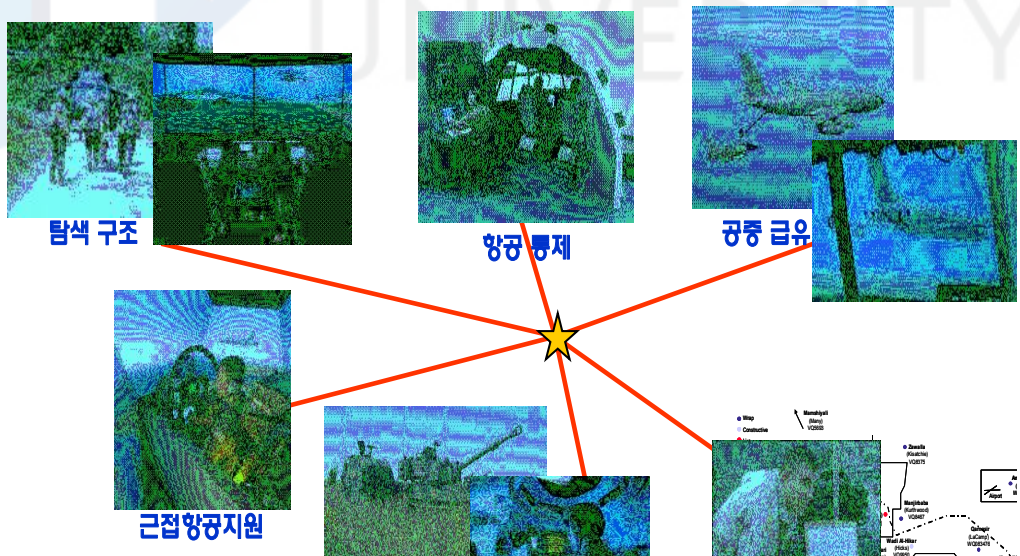
를 실제로 사용하여 훈련에 사용한다는 것은 한계가 있다. 이에 따라 현재 군은 이런 무기체계들을 이용한 훈련을 제한적으로 행하고 있다. 이는 실제 전쟁 시 전투수행능력의 저하를 야기 시킨다.

비용적인 측면에서는 각 군에 따른 실제 훈련 시 발생하는 부대비용은 많은 예산이 소요 된다. 또한 이런 부대비용뿐만 아니라 실제 무기체계를 사용하고 하는 예산 또한 만만치 않은 것이 사실이다. 이는 우리 군에 예산의 많은 부담을 줄 것이며, 앞으로 무기체계나 전술의 연구 및 발전에도 비용적인 측면에서 많은 부담을 안긴다.

실제 훈련을 감행 시에 이에 따른 많은 위험요소가 나타난다. 실제 훈련 시 인사사고 및 오발사고, 자연 파괴 등 곳곳에 위험을 초래 한다.

이러한 한계점을 극복하고 다양한 전술 훈련의 효과를 증대시키기 위해 선진국에서는 인공가상 합성체계개념을 지속적으로 발전 시켜왔고 이에 따라 LVC 체계가 대두되었다.

LVC 체계란 앞서 밝히 바 있듯이 Live, Virtual, Constructive 체계 중 2개 이상의 체계를 연동하는 훈련 체계이다.



<그림 1> 훈련체계 활용사례

앞장의 <그림 1>은 LVC체계를 도식화 하여 나타낸 것이다.

본 논문에서는 V-C 연동에 관한 발전 발전방향을 제시하는 논문으로 V-C 연동의 필요성에 대해 기술 한다.

V-C 연동의 필요성을 살펴보면, 첫째, V-C 훈련체계는 네트워크 중심전(NCW)으로 대변되는 현대전에서 Sensor-to-Shooter(정밀타격)에 이르는 반응시간을 최소화하기 위해, 인적요소가 포함된 세부적인 전투수행 절차훈련과 전구차원의 역동적 훈련 상황 조성을 위한 경제적·효과적인 수단이다. 둘째, V-C 훈련체계는 훈련여건의 변화에 능동적으로 대처하기 위한 현실적 훈련수단이다. 여기서 훈련여건 변화에 관련된 요소로는 소음/환경오염에 대한 문제, 훈련공간의 제한, 임무복잡도 및 훈련경비의 증가, 훈련수단 다양화 등이 있다. 셋째, V-C 훈련체계는 V, C 훈련자산 각각이 갖는 단점을 상호 보완한다. 즉, Virtual의 단점인 훈련의 규모 제한(개인/편대 위주) 및 항시 예측 가능한 위협상황에서 반복훈련이 이루어진다는 점은 Constructive를 통하여 보완할 수 있으며, 반면 Constructive의 단점인 전투원의 인적요소 반영 곤란, 임무절차 수행에 대한 전투원의 훈련 제한, 그리고 모의결과가 그 모델에서 정의한 가정 사항 및 모의논리에 의존한다는 점은 Virtual의 역할로 보완할 수 있다. 넷째, 최근 들어 시뮬레이션/IT 기술의 급격한 발전으로 이제는 실기동 훈련(Live)을 대체할 V-C 훈련체계 구축에 필요한 기술적 기반이 구축되어 있다.⁵⁾

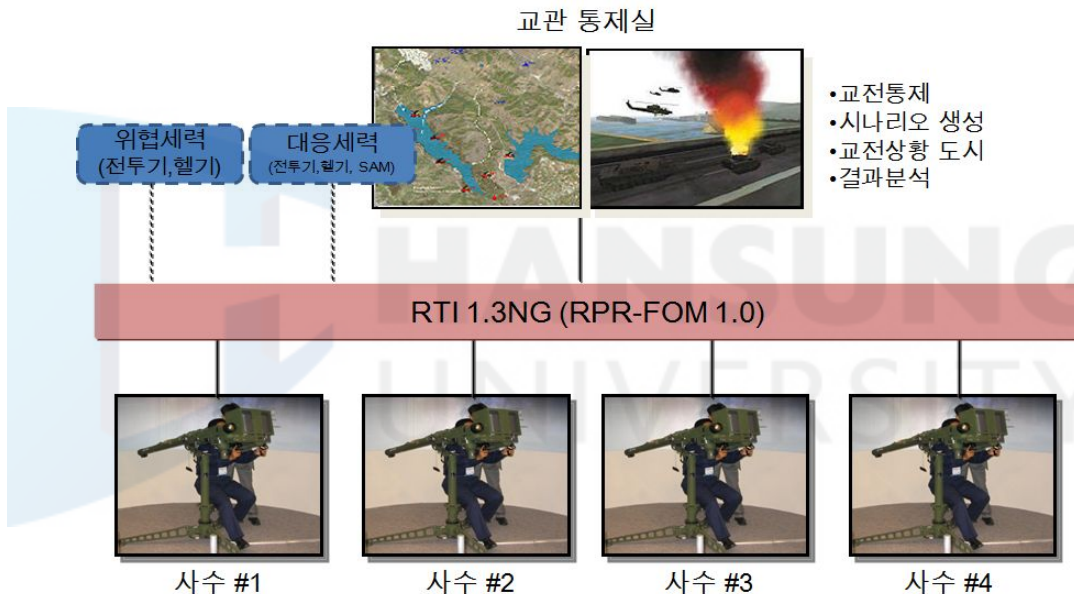
제 2 절 국내외 구축 현황

국내 V-C연동 사례는 선진국에 비해 찾아보기 쉽지 않은 게 사실이다. 이는 국내 분산 환경 시뮬레이션의 시작이 얼마 안 되었다는 것을 단적으로 보여 준다. 이 절에서는 몇 가지 국내외 V-C 연동 사례와 해외 선진 V-C연동 사례에 대해 소개한다.

5) 한국국방연구원(2010), 『공군 LVC체계 연동기술 개념연구』, 한국국방연구원 p.27.

1. 신궁

과거 10년 전에 개발되어 현재까지 양산형태로 이어지는 신궁은 V-C 연동의 좋은 국내 사례 중 하나이다. 신궁은 분산 환경 시뮬레이션에 적합한 HLA 아키텍처와 RPR-FOM 1.0을 사용함으로써 시뮬레이션의 확장성을 높이고 HLA기반의 Constructive 모델과의 연동이 용이하도록 개발되어졌다. 현재 교전급 Constructive의 COTS모델인 Presagis의 STAGE와 연동되어 합성 전장 환경 하에서 운용한다. 신궁의 구성 개념도는 <그림 2>과 같다.



<그림 2> 신궁 구성도

신궁은 최대 4명의 사수가 연동하여 훈련 할 수 있는 HLA기반의 분산 시뮬레이션으로 실제 전장 환경과 유사한 조건의 표적과 지형 DB구축, 사수의 훈련동작에 따른 영상을 실시간으로 처리 할 수 있으며, 실제 한반도 지형에서의 적기의 침투 경로에 대한 모의 시나리오를 구축 한다. 신궁의 특징은 다음과 같다.

Constructive 모델은 검증된 교전급 COTS를 사용하였으며, 최초 구축

시 HLA기반으로 구축하여 분산 시뮬레이션에 적합하도록 구축되었다. 객체 에디터를 포함하여 Virtual과 Constructive모델에 동일한 객체를 전시하였으며, 각각의 모델에 지원 가능한 포맷으로 DB가 제작되었다. Virtual 모델은 별도의 영상처리작업으로 환경 및 기후 변화가 적용된 영상을 투시하였다. 또한 RPR-FOM 1.0을 이용함으로써 FOM의 표준화를 이루었다.

신공은 2000년 개발을 시작으로 현재까지 양산 단계에 있으며 실제 훈련에 활용되어 지고 있는 국내 V-C 연동 사례이다.

2. 나토의 EFW

EFW(Exercise First Warfighter Alliance in a Virtual Environment)는 2004년 11월 가상환경에서 시행된 최초의 대규모 나토의 MTDS 이벤트이다. 본 훈련의 목표는 가상의 분산합성 환경 하에서 COMAO(COMposite Air Operations : 복합항공작전)의 훈련 및 효과를 평가하기 위한 것으로서, 구체적으로는 적의 전자전 환경 하에서 전투기, 공중급유기, 대공제압기, 정찰기 및 공중조기경보기 운용, 방어 작전 절차(전투기 책임구역 및 지점방어 임무) 수행, 공격/방어 작전지원 위한 전자전 자원 운용, 분산된 장소들로부터 구성된 편대군들에게 임무계획, 브리핑, 임무수행, 디브리핑의 전체 프로세스를 수행하였다.

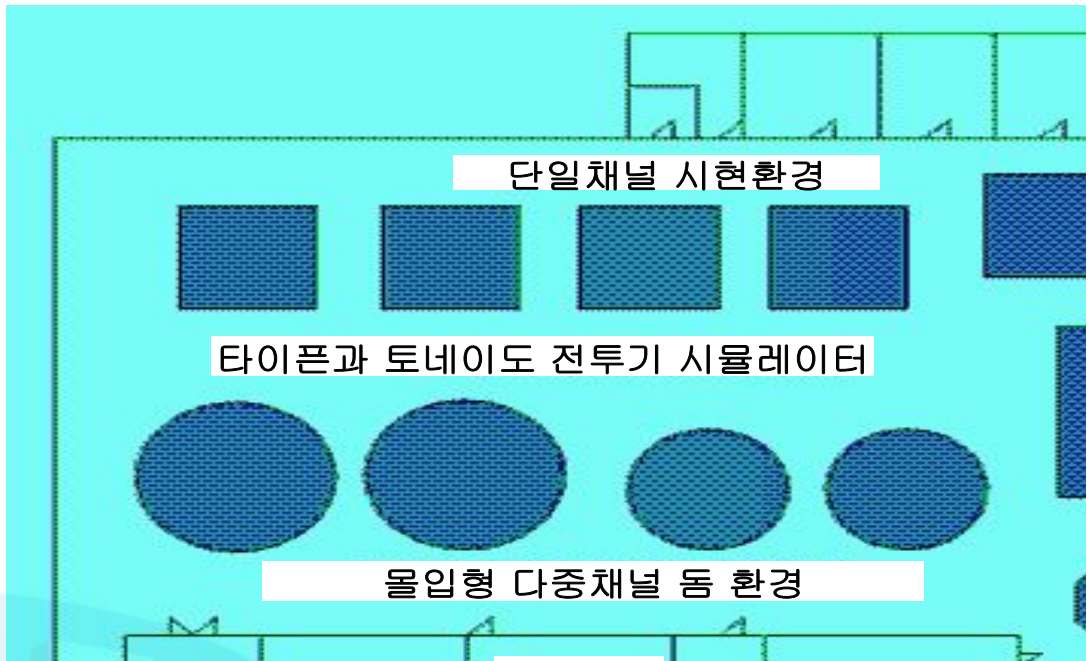
본 훈련에서는 HLA 기반 아키텍처로 선정, 레거시 시뮬레이터들이 실시간으로 참여하였고, 상용 CGF를 활용하며, 네트워크 구성을 위해 7개국의 15개 사이트(유인시뮬레이터 위치)와 미국의 올란도에 위치한 NOC(Network Operation Center : 모의훈련운용센터)가 참여하였다. 훈련 종료 후에는 NATO MTDS 실행/반복 가능성, 훈련가치 등에 대한 평가와 도전과제들이 식별되었다. 본 훈련의 특이 사항은 시뮬레이터/네트워크의 기술적 제약에도 불구하고 훈련참여자들은 훈련효과를 긍정적으로 평가하였다는 점이다.

3. 영국의 MTDS-DSALT

영국 공군의 MTDS(Mission Training via Distributed Simulation :분산 환경하에 임무수행 훈련) 훈련은 영국 공군의 집단임무훈련용 LVC 체계 개발을 위한 개념 시범기로서, 이의 시범연동을 통한 사용자 요구사항을 개발하고, 개념과 능력을 구체화하여 개발 위험을 최소화하고자 하였다. MTDS의 합성환경에서는 타이폰과, 토네이도 및 AWACS 시뮬레이터들로 구성된 2개 이상의 팀이 상호작용 훈련을 실시할 수 있으며, 각 팀은 상이한 군사적 역할을 수행할 수 있도록 되어 있다. 본 체계인 DSALT(Distributed Synthetic Air-Land Training : 분산환경의 공지 훈련) 개발 사업은 현재 활발히 추진 중에 있으며, 이 훈련을 활용함으로써 미 공군 DMO 네트워크와의 연동이 달성되고, 지상전력 합동훈련을 위한 신규소요가 도출되는 등의 성과가 창출되었다.

윗 장에 나타나는 <그림 3>에서 보는 바와 같이 MTDS는 다음과 같은 요소들로 구성한다. 전투기 시뮬레이터 8대는 타이폰 4대, GR-4 4대이며, AWACS 시뮬레이터는 E-3C 5대 랩탑 콘솔 2대, 연습관리 및 통제실는 보잉 BigTac(CGF) 시스템 보잉 InSight 시스템으로 구성되며, 네트워크 관찰 및 기술지원 콘솔, 계획수립-브리핑-디브리핑실, 네트워크(WAN) 연결실로 구분되며, 화력계획반-화력지원팀 운용은 FPC(FST 협조/화력할당), FST(포/항공기 지시)으로 구성된다.

MTDS 연습관리 통제실에서는 2가지 보잉 제품들을 사용한다. 보잉의 BigTac 시스템은 컴퓨터생성부대(CGF : Computer Generated Forces) 기능을 통해 풍부한 작전환경을 제공한다. 또한 보잉의 InSight(시스템의 모니터링 및 제어 프로그램)는 연습통제관용 상황을 시현하고 BigTag CGF를 통제한다. 계획수립-브리핑-디브리핑실의 승무원 계획 수립실은 VTC, 대화형 화이트보드, 기타 분산협업도구를 구비하여 훈련을 지원하고, 브리핑 / 디 브리핑 실은 2D와 3D InSight AAR 도구를 보유하여 브리핑과 디 브리핑 시에 활용한다.



<그림 3> MTDS의 구성 체계

네트워크 연결을 담당하는 WAN 연결실은 암호화장비, 전환기, 필터링, 방화벽 등을 제공하여 외부 체계와의 네트워크 연결 기능을 제공한다. 국내의 구축 사례인 신궁과 나토의 EFW, 영국의 MTDS의 국내외 구축 사례를 소개하였다. 앞서 언급을 했지만 다시 한 번 V-C 훈련체계의 필요성에 대해 얘기 하자면, 첫째, V-C 훈련체계는 네트워크 중심전(NCW)으로 대변되는 현대전에서 Sensor-to-Shooter(정밀타격)에 이르는 반응시간을 최소화하기 위해, 인적요소가 포함된 세부적인 전투수행 절차훈련(Virtual)과 역동적 훈련 상황 조성(Constructive)을 위한 경제적·효과적인 수단이다. 둘째, V-C 훈련체계는 훈련여건의 변화에 능동적으로 대처하기 위한 현실적 훈련수단이다. 여기서 훈련여건 변화에 관련된 요소로는 소음/환경오염에 대한 문제, 훈련공간의 제한, 임무복잡도 및 훈련경비의 증가, 훈련수단 다양화 등이 있다. 셋째, V-C 훈련체계는 V, C 훈련자산 각각이 갖는 단점을 상호 보완한다. 즉, V의 단점인 훈련의 규모 제한(개인/편대 위주) 및 항상 예측 가능한 위협상황의 반복훈련이 이루어진다는 점은 Constructive

를 통하여 보완할 수 있으며, 반면 Constructive의 단점인 전투원의 인적요소 반영 곤란, 임무절차 수행에 대한 전투원의 훈련 제한, 그리고 모의결과가 그 모델에서 정의한 가정 사항 및 모의논리에 의존한다는 점은 Virtual의 역할로 보완한다.

이러한 V-C 연동의 중요성이 두각 되면서 V-C 연동에 있어 연동의 문제점도 함께 두각 되고 있다.

제 3 절 V-C 연동의 문제점

V-C연동의 문제점은 이미 기 개발된 Constructive와 Virtual 간의 연동에서 가장 큰 문제가 나타난다. 이에 따른 이슈를 크게 보면 다음과 같다. 첫째, 동일 전투모의 공간의 공유가 필요하다. 소부대 전술 훈련의 예로 기체 화보병대대는 훈련제대의 전투특성 면에서 교전간 적과 혼재되면서 전투영역이 중첩되기도 하고, 편조에 의해 인접부대 간의 상호작용이 활발히 이루어진다. 이런 상황에서 전투모의 공간이 분할되어 부대별 각기 다른 영역을 가진다면 모델 간 (V, C) 상호작용 및 교전효과가 나타나지 않을 수가 있으며, 특성에 맞지 않는 모의가 이루어지게 된다. 둘째, 전투모의 공간을 공유하기 위해서는 V, C 모델 내 개체 간 직접 연동이 필요하므로 해상도의 일치가 필요하다. 현재 군이 보유하고 있는 훈련용 위게임 모델은 대부분 Aggregated Level 모델로 개체별 모의가 아니라 부대별 모의가 이루어진다. 예를 들면 전차 1대, 병사 1명이 정의되어 있지 않고 전차 4대의 집합인 소대, 병사 10명이 모인 분대 등과 같이 부대 단위로 모의되어 있어 Disaggregated Level인 Virtual 모델과 연동이 제한된다. 모델 간 연동을 하기 위해서는 이러한 해상도 불일치를 해결해야 한다. 셋째, 소부대 전술 훈련 시, 작전 운용 간 지형의 영향을 많이 받으며 소대, 중대급 부대가 전투의 중심이므로 국지적으로 이루어지는 소규모 전투가 대대전체에 미치는 영향이 크다. 현재 위 게임 모델이 보유하는 환경 DB와 전차 시뮬레이터가 가진 환경 DB내용이 상이하기 때문에 연동 시에 공정하지 못한 판정이 발생한다. 예를 들면 위 게임 내 환경 DB 특정 공간에 장애물 데이터가 없다면 위 게임의 전차 개체 A는 전차 시

시뮬레이터의 개체 B를 볼 수 있고 사격을 실시한다. 하지만 전차 시뮬레이터 내 환경 DB 동일공간에 장애물 데이터가 있다면 전차 시뮬레이터의 개체 B는 위 게임의 전차 개체 A를 볼 수 없고 사격을 실시할 수 없다. 넷째, 위 게임 모델과 전차 시뮬레이터가 연동을 하여 훈련을 하게 되면 전차 시뮬레이터의 전차장과 위 게임 모델 훈련부대의 지휘관 및 직속상관 그리고 인접 시뮬레이터 전차장 사이에 명령을 주고받을 수 있는 편제 전술통신이 필요하게 된다. 다섯 번째는 기 개발된 위 게임 모델과 Virtual 연동 시 기반이 된 아키텍처가 다를 수 있다. 예를 들어 위 게임은 모델은 DIS 아키텍처를 기반으로 개발되었고, Virtual 시뮬레이터는 HLA기반으로 만들어 졌다면 이기 종간의 연동 기술이 필요하게 된다.⁶⁾ 본 논문에서는 전구급 Constructive 모델이 아닌 교전급 Constructive의 기준을 두었으며, 해상도 일치부분을 배제한 연동 기술을 기술한다.

이러한 문제점은 크게 4가지로 구분되어 질 수 있다. 첫 번째가 이기종간의 아키텍처 연동/통합, 두 번째는 공동 합성전장 환경 구현, 세 번째는 네트워크, 마지막으로 연동/훈련 관리 도구이다. 이 4가지 외에도 보안의 문제, 비밀등급별 보안대책, 승인된 암호장비, 제반 정보보호의 대책이 필요하지만 이 번 논문에서는 배제하기로 한다. 다음 [표2]는 V-C연동에 나타나는 일반적인 문제점을 나타낸 표이다.

[표 2] V-C 연동 시 나타는 문제점

구 분	상 세 내 용
아키텍처 연동/통합	이기종 아키텍처 간 게이트웨이/브리징 기술, 필터링, 포털 기능 공급자가 다른 RTI 적용시 연동 기술
공동합성 전장 구현	공동의 개념모델 정의 개념모델의 상세화 수준 합의 세부구현(인터페이스, 기술/성능, 절차)에 대한 약정

6) 김문수 외(2012), “소부대 전술 훈련을 위한 개체기반 위게임 모델과 전차시뮬레이터 연동에 관한 연구”, 『한국군사과학기술학회지』, 한국군사과학기술학회 pp.36-37.

	모의 해상도 동기화(시간 동기화, 모의수준 동기화) 공동의 임무데이터 제공 서비스(표준 레파지토리, 소유권/접근성 등) 합성환경 보강 위한 구성모의(CGF) 방안 합의
네트워크	지연시간 - 대역폭
연동/훈련 관리도구	모의지원 도구(모델 개발/관리, 페더레이션 실행 검증/관리/관찰/제어) 훈련단계별 지원도구

[표2]에서 나타나듯이 현재 V-C체계의 연동에는 많은 문제점이 제시되었다. 현재까지 이런 한계점의 극복방안 연구와 기술의 발달로 인해 이러한 문제점이 많이 극복되어 졌지만, 아직까지는 완벽한 V-C체계 연동에는 어려움 점이 남아있다.

본 논문에서는 위에 나타난 문제점 중 아키텍처 연동/통합과 공동 합성 전장 환경에 대해 COTS와 HLA Evolved를 사용하여 향상된 방향을 제시하고자 한다.

본 논문의 3장에서는 RPR-FOM의 적용과 HLA Evolved, 공동DB의 적용을 통한 V-C 체계 연동의 향상된 아키텍처를 제시하며, 4장에서는 제시된 아키텍처를 기반으로 검증된 COTS를 사용하여 실제 구축을 한다.

제 3 장 HLA Evolved와 COTS 기반의 V-C 연동 아키텍처

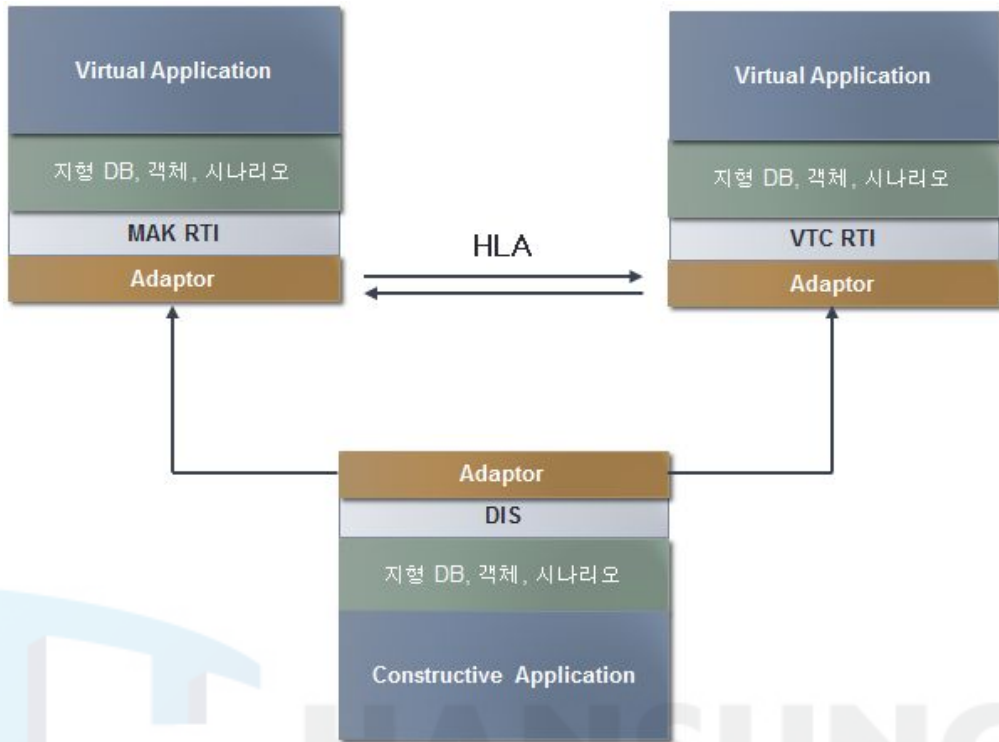
COTS(Commercial, Off-the-shelf : 상용제작도구)란, 상용화 된 S/W를 말한다. 이는 검증을 마치고 제품화 되어 출시되는 S/W이다. COTS의 가장 큰 장점으로서는 이미 검증이 되었다는 부분과 다양한 적용 사례가 있다는 것이다. 이는 시뮬레이션의 안정화를 보장하는데 큰 기여를 하는 것뿐만 아니라, 시뮬레이션의 개발기간을 단축시키는 장점 또한 있다. 반면, 개발업체에 의지를 해야 한다는 점과 도입 비용에 대한 부담이 있다는 것이 단점이다.

본 논문의 이번 장에서는 제시하는 V-C 연동에 사용 될 COTS의 선정과 소개를 다루며, 이를 바탕으로 HLA Evolved의 적용으로 인해 향상된 V-C 연동의 아키텍처를 제시 하고 이를 기존에 구현된 아키텍처와 비교 분석 한다.

제 1 절 현재 V-C 연동 아키텍처

현재 V-C 연동을 위해서는 연동을 위한 Adaptor를 사용함으로써 연동이 이루어진다. 이 Adaptor의 기능은 이기종 아키텍처간의 연동과 공급자가 다른 RTI의 연동, 각 시뮬레이션의 갱신주기 동기화를 지원한다. 이러한 Adaptor는 기존의 많은 연구와 제안이 있었고, 현재 이 Adaptor를 연동에 이용하고 있다. 다음 장에 나타내는 <그림 4>은 현재 V-C연동의 아키텍처를 나타낸 것으로서 각각의 시뮬레이션과 시뮬레이터는 연동을 위한 Adaptor를 포함하고 있다.

본 논문의 이번 장에서는 이러한 연동을 위한 Adaptor를 대체 하여 COTS와 HLA Evolved를 사용하는 아키텍처를 제시 한다.



<그림 4> 현재 V-C 연동 아키텍처

<그림 4>에서 보여 지는 봐야 같이 각각의 연동이 필요한 Adaptor를 포함하고 있다.

다음 장에 나타내는<그림 5>에서 보여 지는 봐와 같은 Adaptor의 구조를 가지고 있으며, 이를 통해 V-C는 연동 하게 된다.

이기중간의 연동 모듈은, 이기종 즉 HLA, DIS, TENA등 서로 다른 아키텍처를 기반으로 하는 시뮬레이션 또는 시뮬레이터의 연동을 위한 모듈이고, 내부통신 모듈은 각 시뮬레이션 또는 시뮬레이터의 연동 시 예기치 못한 종료 및 오류 발생 시 다른 시뮬레이터 또는 시뮬레이션에 영향을 끼치지 않도록 처리하는 역할과 연동 시스템을 전체적으로 모니터링 하는 모듈이다. 공급자가 다른 RTI 모듈은 상용 제품의 RTI를 이용 시 공급자 즉, 만든 회사가 다른 경우의 연동을 위한 모듈이다. 갱신을 동기화 모듈은 시뮬레이션 또는 시뮬레이터의 갱신율이 서로 다른 갱신율을 갖고 있

을 때 사용되며, 이 모듈을 사용하여 서로 같은 갱신율을 동기화 시켜 줌으로서 연동이 가능하도록 한다.



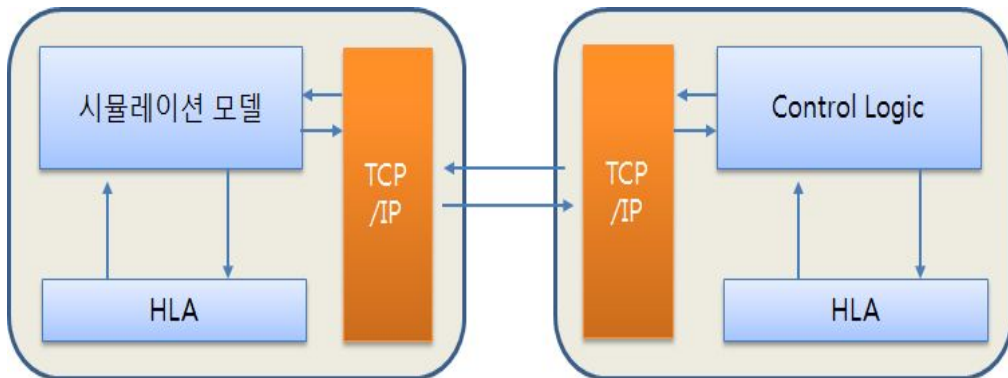
<그림 5> 연동 Adaptor의 구조

이러한 연동 Adaptor의 구조에 대해 각각의 모듈 별로 살펴보도록 한다.

1. 제어 모듈

제어 모듈이란 전체적인 연동 모델을 모니터링 하는 역할 등을 수행할 뿐만 아니라 각기 HLA기반에서 연동되는 시뮬레이션 또는 시뮬레이터의 애기치 못한 종료 또는 문제가 발생 했을 시, 문제가 발생된 시뮬레이션 또는 시뮬레이터를 강제 종료하는 것을 말한다. 이는 문제가 생긴 시뮬레이션 또는 시뮬레이터를 계속 연동 상태로 두게 되면 이 문제가 다른 시뮬레이션 또는 시뮬레이터에

영향을 미쳐 전체적인 연동에 문제가 나타날 수 있기 때문이다.



<그림 6> 제어 모듈

<그림 6>는 제어 모듈의 구조를 나타 낸 것으로 시물레이션모델과 제어를 담당하는 Control Logic은 TCP / IP로 연결되어 있다. 시물레이션 모델에서 예기치 않은 오류나 종료 시 Control logic에서 TCP / IP 프로토콜을 통해 강제 종료명령을 줌으로서 문제가 있는 연동 모델은 강제 종료가 된다. 제어 모듈에서 수행하는 역할 중 가장 큰 역할은 현재 운용중인 모든 시스템의 상태에 대한 모니터링 기능으로 시스템의 비정상 작동 시 강제 종료뿐만 아니라 시물레이션 연동 모델이 정상복구 되었을 시 이를 다시 HLA 아키텍처에 JOIN시키는 역할도 수행하게 된다.

이러한 시스템 모니터링 기능은 HLA를 이용하지 않고 TCP/IP를 통해 주기적 호출을 수행해 시스템의 상태를 감시하고 비정상 작동 시 이를 사용자에게 알려 사용자가 편리하게 시스템의 상태를 모니터링하고 관리하게 한다. 예를 들어 Control logic에서 이상을 감지한 신호를 받아 시물레이션 모델에 강제 종료 신호를 주었을 시 신호를 받아 강제 종료를 수행하고 복구 후 Control logic에서 Join 명령을 전송 시 시물레이션 모델은 이를 수신 후 Federation에 Join한다.

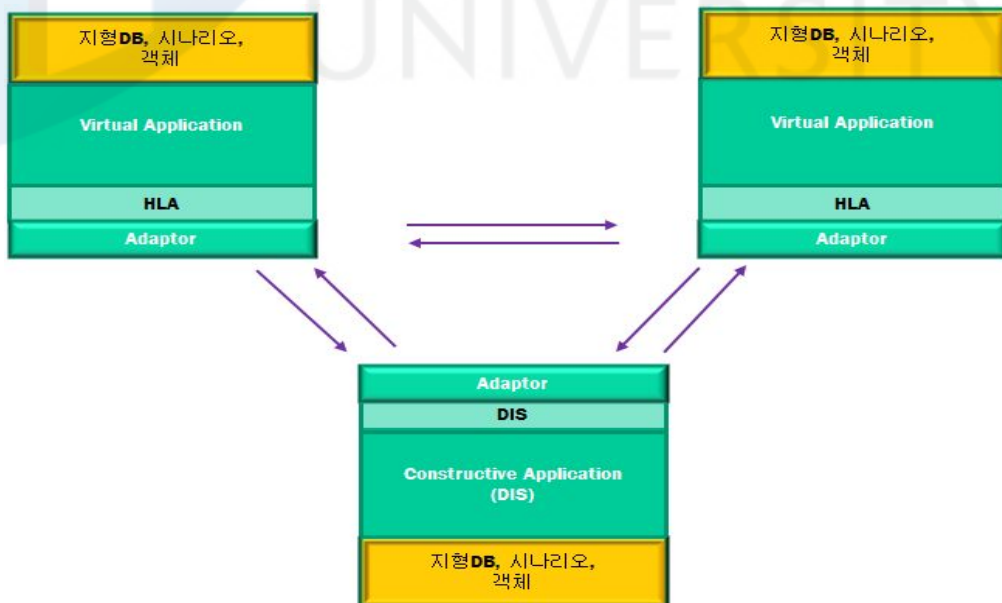
RTI Service에서도 이와 유사한 서비스를 제공하지만 이는 RTI를 위한 서비스에 대한 모니터링 서비스 일뿐 시스템 자체에 대한 모니터링 서비스는 제공하지 않는다. 따라서 이러한 서비스를 구축하기 위해 TCP/IP를

이용한 Control Logic과 같은 시스템 모듈을 개발한다. 또한 모든 시스템이 RTI와 연동하여 사용하기 쉽도록 인터페이스가 되지 않는 경우가 있다. 이러한 대표적 공학용 프로그램으로는 Matlab등이 있으며 이렇게 인터페이스가 쉽지 않는 프로그램과의 인터페이스를 사용하는 경우에는 표준적인 통신 프로토콜인 TCP/IP를 이용하여 기존의 응용 프로그램과 인터페이스가 쉽도록 개발한다.

2. 이기종 아키텍처간의 연동

<그림 7>은 현재 V-C 연동의 아키텍처 구조를 나타낸 것으로서 각각의 Virtual 어플리케이션과 Constructive 어플리케이션은 서로 다른 프로토콜의 기반으로 사용하여 구축되어 있는 모습을 볼 수 있다. 이러한 이기종간의 아키텍처를 연동하기 위해서는 연동을 위한 모듈이 필요로 하게 된다.

<그림 7>이런 이기종간의 아키텍처의 연동을 위한 아키텍처를 나타낸 것으로서 연동 모듈을 연동을 위한 Adaptor로서 표현한 아키텍처이다.



<그림 7> 이기종간 V-C 연동의 아키텍처

앞장의 <그림 7>에서 보여 지는 V-C 연동 아키텍처의 문제점은 이기종 아키텍처간의 연동 문제이다. 맨 밑쪽에 자리한 Constructive 모델은 DIS기반의 시뮬레이션이고, 위 양쪽에 위치한 것은 HLA기반의 Virtual 모델들이다. Virtual 모델들은 모두 HLA기반이기 때문에 서로의 연동에 있어서는 문제 될 것은 없다. 하지만 아키텍처가 다른 Constructive모델과의 연동에 있어서는 별도의 Adaptor가 필요하게 된다.



<그림 8> 이기종간 연동을 위한 Adaptor 구조

<그림 8>의 왼쪽 변환자는 HLA의 RPR-FOM의 Object Class를 기준으로 DIS PDU의 데이터로 변환시킨다. 반대로 오른쪽의 변환자는 DIS PDU의 데이터를 HLA의 데이터로 변환시키는 작업을 수행하게 된다. 이렇게 변환이 거친 후 전송을 시작 하게 된다. DIS와 HLA는 서로 다른 전송 메커니즘을 이용하고 있다. 그 방식의 차이에 따라 단순히 데이터 패킷을 매칭 하는 역할 뿐만 아니라 구조적인 처리도 같이 해 주는 역할을 전송 메커니즘에서 수행한다. 이런 두 가지 단계를 거쳐 HLA와 DIS는 연동을 하게 된다.

위와 같은 실제 모듈을 개발하기 위해서는 인원, 개발준비, 개발, 검증의 과정을 거쳐야 하기 때문에 실제 개발, 적용 하는데 있어서 많은 시간과 노력 그에 따른 많은 비용이 발생하게 된다.

3. 공급자 다른 RTI의 연동

HLA 기반으로 연동 시 문제가 되는 것 중 하나는 어떤 RTI를 사용하여 연동을 하고자 하는 것이다. 이는 사용자가 사용 할 수 있는 RTI의 종류가 여러 개가 된다는 것을 의미한다. 현재 분산 환경의 시뮬레이션에서는 거의 대부분이 상용 RTI를 사용하고 있다. 공급자가 다른 RTI라는 것은 이러한 상용 RTI를 만드는 회사가 각기 다를 것을 뜻한다. 현재 대표적으로 사용되는 상용 RTI는 MAK RTI, VTC RTI, Pitch RTI가 대표적이다. 이러한 상용 RTI는 혼합되어 사용되어지지 않도록 되어 있다.



<그림 9> 공급자가 서로 다른 RTI의 연동

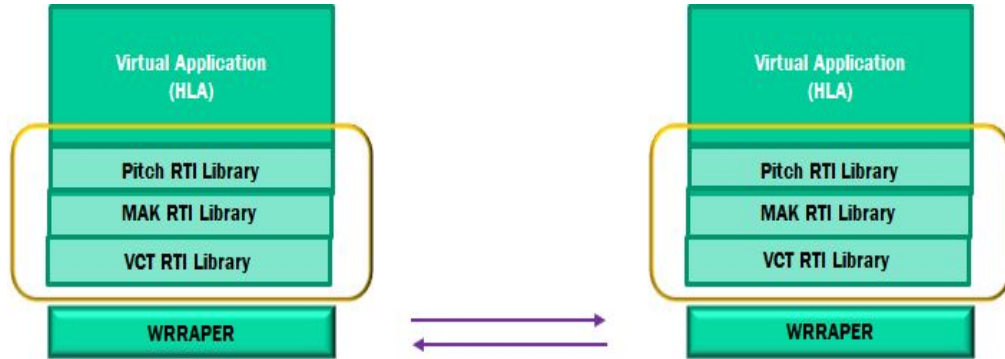
<그림 9>에서 보여 지는 좌와 같이 왼쪽의 Virtual 어플리케이션은 HLA기반의 MAK RTI를 이용한 분산 환경의 시뮬레이션이고 오른쪽의 Virtual 어플리케이션은 같은 HLA기반 이지만 MAK RTI가 아닌 VTC RTI를 사용 하였다.

이렇게 서로 다른 이기종간의 RTI를 이용 시 (MAK RTI, VTC RTI) 별도의 연동을 위한 모듈이 필요로 하게 된다.

이 이기종간의 RTI 연동을 위한 모듈은 WRAPPER⁷⁾를 이용하여 공급자가 서로 다른 RTI의 Library(MAK RTI Library, VTC RTI Library, Pitch RTI Library)를 WRAPPER가 포함하고 있음으로서 WRAPPER를 통해 연동 되어

7) WRAPPER: 사용 RTI의 Library를 포함하고 있는 모듈의 통칭이다.

지려는 시뮬레이션 또는 시뮬레이터의 사용되어지는 공급자의 RTI Library를 통해 서로 연동을 가능하게 한다. <그림 10>은 공급자가 다른 RTI 연동 모듈의 구조를 나타낸 것이다.



<그림 10> 공급자가 다른 RTI 연동 모듈

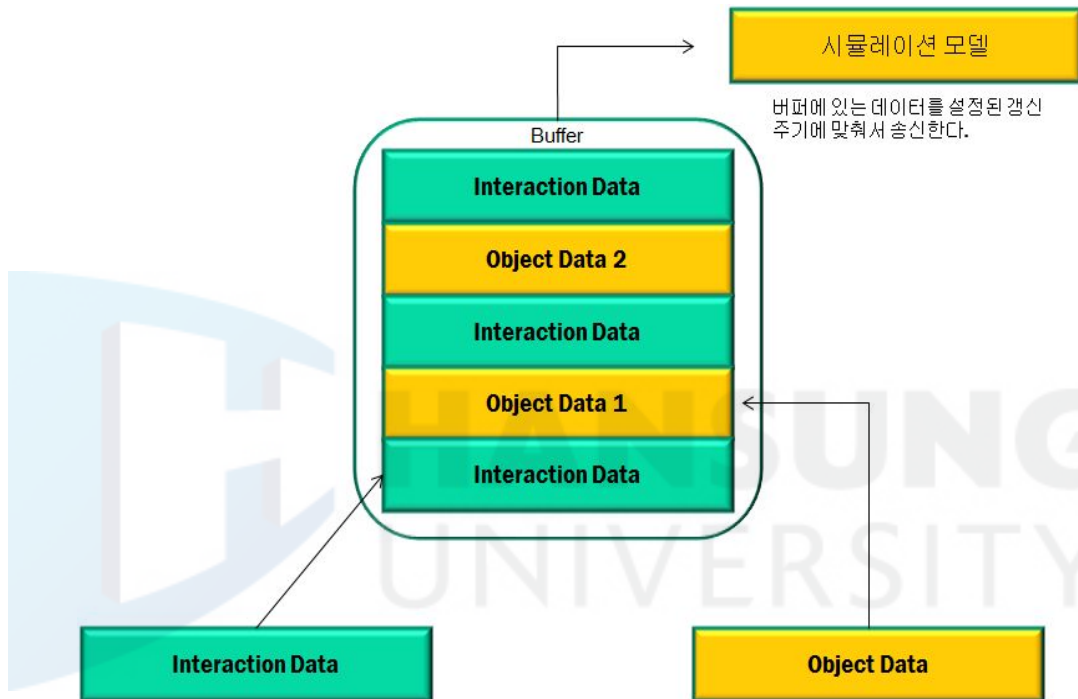
<그림 10>에서 보여 지는 바와 같이 양쪽의 Virtual 어플리케이션은 공급자가 다른 RTI를 사용하였다. WRAPPER는 그 밑단에서 공급자가 다른 RTI들의 라이브러리를 갖고 있음으로서 변환자의 역할을 하는 구조이다.

이는 공급자가 서로 다른 RTI사용 하더라도 동일한 FOM, SOM을 사용하면 주고받는 데이터를 이기종 RTI 연동 모듈을 통해 상대방과 동일한 공급자의 Library를 사용하여 상대방에게 데이터를 전송하면 RTI간의 연동에 따른 문제는 발생하지 않는다. 이는 서로 HLA / RTI 아키텍처를 따르기 때문에 공급자간의 사용하는 Library는 다르더라도 최종적으로 전달되는 데이터는 동일하기 때문이다.

4. 갱신 주기율 동기화

HLA를 통해 연동 시 각각의 시뮬레이션 모델들은 서로 다른 갱신 주기율을 갖게 된다. 이는 각각의 시뮬레이션 모델이 서로 다른 환경과 개발 목적으로 개발 되어 졌기 때문이다. 이들을 연동을 하기 위해서는 서로 다른 각각의 갱신 주기율을 맞춰줄 필요가 있음을 뜻한다.

갱신 주기가 서로 다른 시뮬레이션 모델을 연동하였을 때 데이터 갱신주기에 따른 페널티(Penalty), 혹은 어디밴티지(advantage)가 발생하게 된다. 이러한 불일치를 해소하기 위해 참여한 시뮬레이션 중에 갱신주기가 제일 낮은 시뮬레이터의 갱신 주기를 다른 모든 시뮬레이터에 맞추어 주는데, 이때 갱신 주기를 낮게 제어하는 것이 갱신 주기율 모듈의 역할 중의 하나이다.



<그림 11> 갱신 주기율 모듈

<그림 11>는 갱신 주기율 모듈의 내부 구조를 나타낸 것으로 갱신 주기율 모듈의 네트워크 데이터 갱신주기 제어 방식은 다음의 두 단계를 거친다. 첫 번째 단계는 내부적인 버퍼 공간을 형성하여 전달되는 모든 네트워크 데이터를 시간순서에 맞게 버퍼에 쌓아 놓는다. 이때 Object 데이터는 Interaction 데이터와 다르게 버퍼에 쌓을 때 이미 전달된 동일한 Object 데이터가 있을 때는 내용만 업데이트 하게 된다.

두 번째 단계는 갱신주기에 따라 현재 버퍼에 쌓여 있는 모든 네트워크 데

이터를 HLA / RTI에 전달한다.

이 두 단계를 거침으로서 연동되는 시뮬레이션 모델들은 갱신주기가 가장 낮은 시뮬레이션 모델의 갱신주기율을 맞추어 연동하게 된다.

제 2 절 향상된 V-C 연동 아키텍처 구성

앞서 제 3 장 1절을 통해 현재 V-C연동의 아키텍처와 연동에 필요한 연동 Adaptor 모듈에 대해 기술했다.

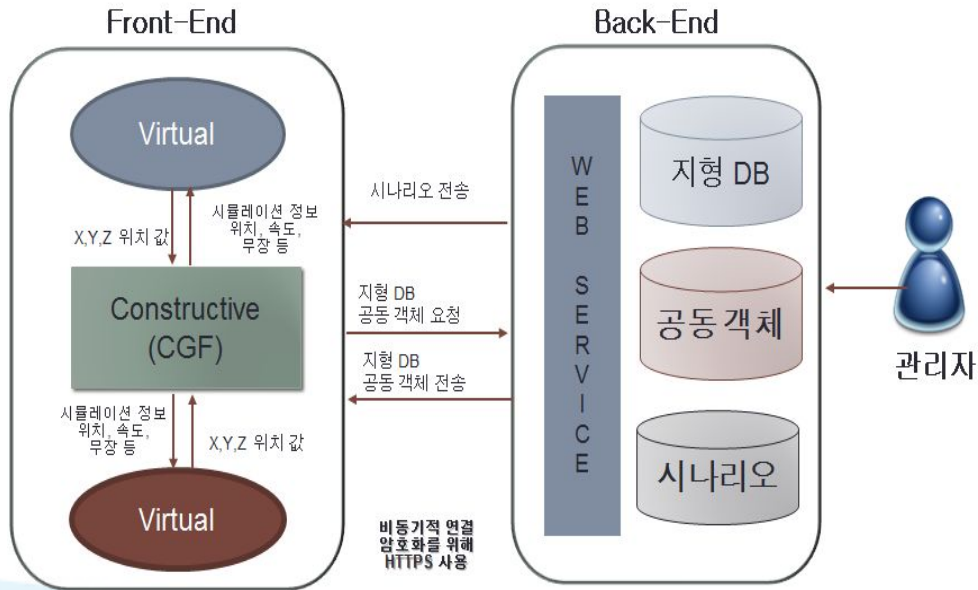
앞서 언급한 바와 같이 연동 Adaptor를 크게 4가지 모듈로서 나누었는데, 그 첫 번째가 제어 모듈이다. 이는 연동 모델의 예기치 않은 오류가 발생했을 시 제어 모듈을 통해 문제가 있는 모델을 강제종료 시키고, 복구 되었을 시 이를 다시 JOIN시키는 역할을 한다.

두 번째는 이기종 아키텍처간의 연동을 위한 모듈이다. 이는 서로 다른 프로토콜을 기반으로 갖고 있을시 서로 연동을 하기 위한 모듈로 변환자를 이용하여 데이터를 변환하여 전송 메커니즘을 통해 연동 되어지는 방식의 모듈이다. 이 모듈은 실제 구현이 어렵고, 장시간의 개발이 필요하다는 문제점이 야기 됐다.

세 번째는 이기종 RTI간의 연동 문제이다. 기본적으로 공급자가 다른 RTI를 이용 시에는 연동이 되지 않는다. 이런 문제를 해결하기 위해 WRAPPER를 이용하여 각기 다른 공급자의 Library를 포함하고 있음으로서 연동을 시키는 방식의 모듈이다.

네 번째는 갱신 주기의 동기화이다. 각각 시뮬레이션 또는 시뮬레이터 모델들은 서로 다른 갱신 주기율을 갖고 있기 때문에 이를 연동 했을 시 문제점이 발생한다. 따라서 갱신 주기율 모듈을 통해 이 각기 다른 갱신 주기율을 가장 낮은 쪽의 갱신 주기율을 기준으로 맞춤으로서 연동 했을 시 갱신 주기율에 대해 생기는 문제점을 해결 했다.

제 2 절에서는 기존의 3가지 모듈을 포함한 Adaptor를 활용하지 않는 대신 COTS와 HLA Evolved의 프로토콜을 이용하여 기존 Adaptor의 역할을 수행하는 향상된 V-C 연동에 대해 논한다.

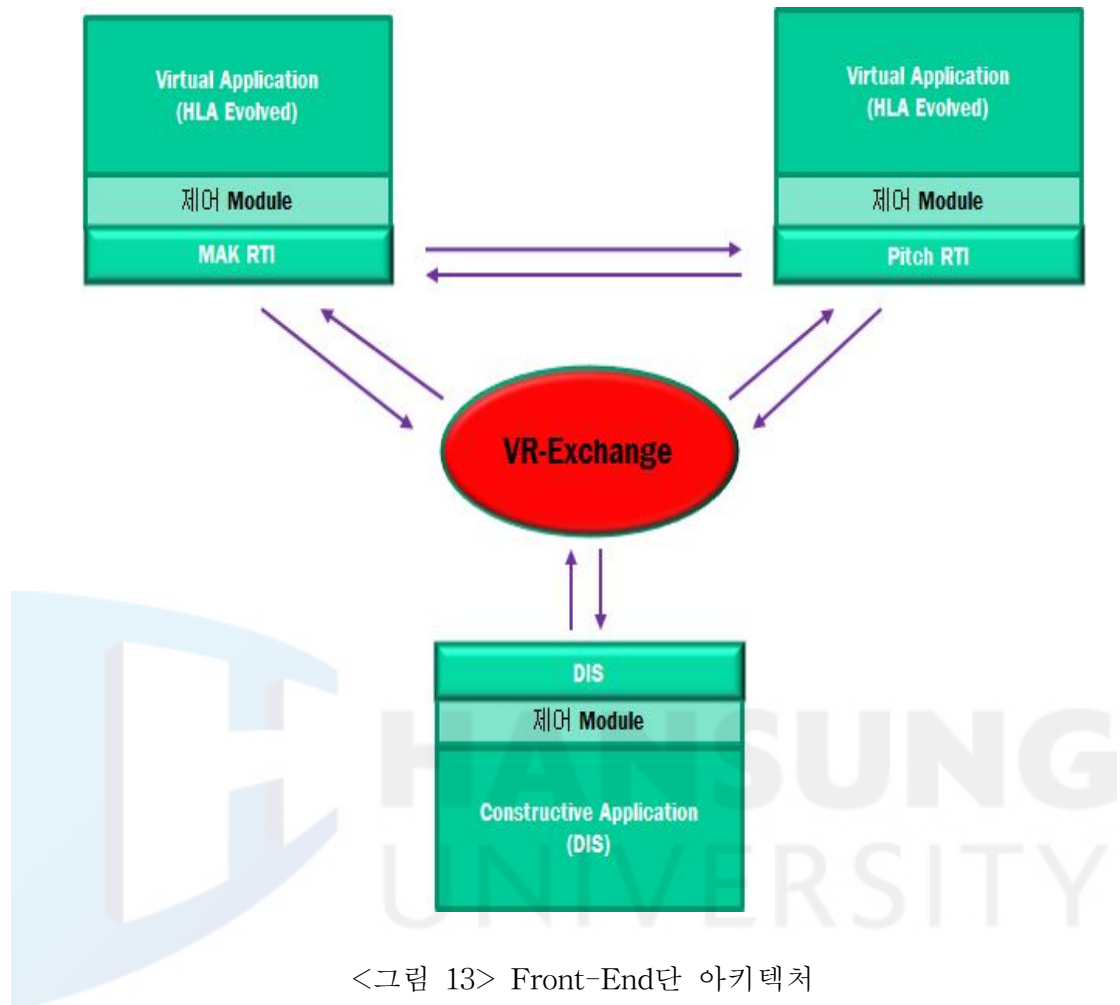


<그림 12> 향상된 V-C 연동 방안

<그림 12>은 이 논문에서 주장하는 V-C의 연동 아키텍처이다. 이 아키텍처는 크게 두 가지로 나눌 수 있다. 앞단의 V-C 연동 부분인 Front-End단과 뒤에서 서버 역할을 하는 Back-End단 이다.

1. Front-End

Front-End단인 V-C연동 부분을 살펴보면 기존과 달라진 점은 V와 C가 각각 갖고 있던 지형 DB, 객체, 시나리오 등의 요소가 사라졌다. 이들은 그러한 요소들은 Back-End단에서 공통으로 관리하여 각각의 V와 C에 전송된다. 이렇게 구성이 되면 Front-End단에서는 서로의 연동부분만이 이루어지면 되게 된다. V-C 연동이 이루어지기 위해서는 앞서 1절에서 말한 부분들이 해결 되어야 한다. 이는 COTS와 HLA Evolved를 통해 해결 한다.



<그림 13> Front-End단 아키텍처

<그림 13>은 Front-End단의 아키텍처를 나타 낸 것으로 기존의 V-C 아키텍처에 비해 제어 모듈을 제외한 다른 모듈들은 필요 없음을 나타내고 있다. 제어 모듈은 아직까지 COTS와 HLA Evolved에서의 지원을 하지 않기 때문에 그대로 사용 한다. COTS와 HLA Evolved를 사용하여 3가지 모듈의 역할을 대신하게 되는데 정리 하면 다음 장의 [표3]과 같다.

[표 3] COTS와 HLA Evolved의 활용

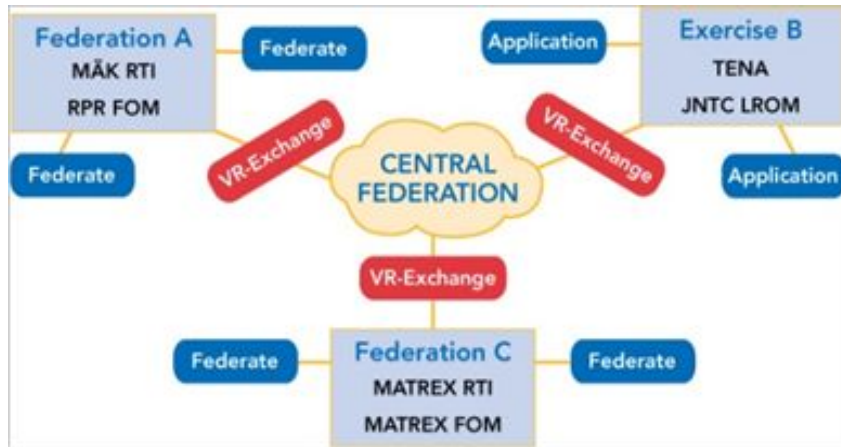
연동 문제점	대처 방안	내용
이기종 프로토콜간의 연동	COTS 활용	- TENA, DIS, HLA 연동을 위한 COTS, 각각의 Broker를 이용하여 변환 및 전송
공급자가 다른 RTI 연동	COTS 활용 또는 HLA Evolved 활용	- 공급자가 다른 RTI의 연동을 위한 COTS, 사용자의 별도의 모듈 없이 자동으로 연동 - HLA Evolved에서 지원하는 API를 활용하여 연동
갱신 주기율	HLA Evolved 활용	- HLA Evolved에서 지원하는 API를 활용하여 해결

COTS와 HLA Evolved를 활용하면 앞장의 <그림 13>과 같은 아키텍처의 모습이 그려지게 되고 이는 기존의 V-C연동에 비해 심플해지는 것을 볼 수 있다. 이는 연동을 위한 여러 가지 모듈이 필요 없음을 뜻한다. 이러한 모듈이 사라지게 되면 개발자 또는 사용자는 쉽게 연동을 구현 할 수 있고, 이는 시간의 절약과 비용발생을 최소화한다.

연동을 위해 사용 되어지는 COTS는 MAK사의 VR-Exchange라는 COTS로서 현재 연동을 위해 검증을 받은 COTS중 하나이다.

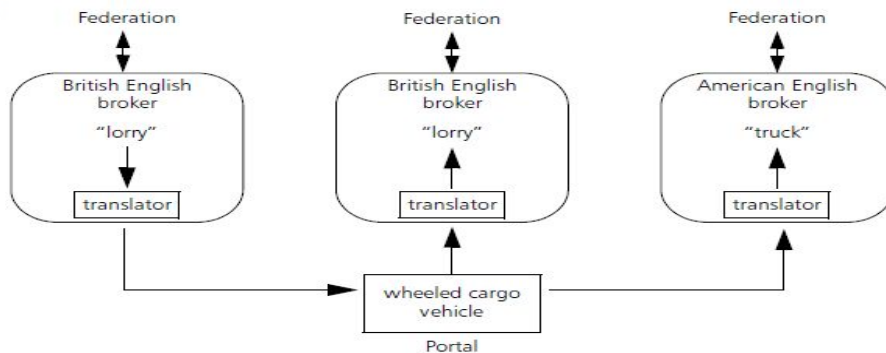
가. VR-Exchange

VR-Exchange는 서로 다른 연동 아키텍처, 이기종 간의 연동 시 사용된다. 예를 들어 DIS-HLA, TENA-HLA등 아키텍처가 다를 경우 중간 게이트 역할을 해서 연동을 가능케 한다. 또한 공급자가 다른 RTI, Federation과 Federation의 연동 경우에도 가능케 하는 역할을 한다. VR-Exchange의 아키텍처는 다음 장의 <그림 14>와 같다.



<그림 14> VR-Exchange 아키텍처⁸⁾

<그림 14>와 같이 VR-Exchange를 사용하여 이기종간의 아키텍처의 연동을 가능케 할뿐 아니라, 공급자가 다른 RTI, 연동과 Federation과 Federation간의 연동 또한 가능하게 한다. 이기종 아키텍처간의 연동 시 VR-Exchange는 각기 다른 FOM을 읽어 들이는 broker의 개념을 이용한다. 따라서 각기 다른 아키텍처의 broker가 필요하다. 예, DIS broker, TENA broker, HLA broker 등 이런 broker를 활용하여 언어를 불러들려 변환자를 이용하여 연동에 맞는 언어로 번역하는 방식으로 이루어져 있다.



<그림 15> Translator의 예⁹⁾

8) MAK Technology(2009), 『VR-Exchange userguide』, U.S.A, p.13

9) *ibid*, p.15

앞장의 <그림 15>은 이해를 돕기 위한 VR-Exchange의 변환자의 예이다. 동일한 데이터를 변환자를 통해 각자의 언어로 전송함으로써 이들이 쉽게 연동 될 수 있음을 나타내고 있다.

나. HLA Evolved

HLA는 1990년대 초 U.S. DoD M&S Vision에 목적을 바탕으로 분산 시물레이션의 재사용성과 상호운용성의 개선을 위해 U.S. Department of Defense (DoD)에 의해 개발 되었다. 분산 시물레이션의 연동 기술은 과거에서부터 현재 계속 발전 중이고 이에 따른 새로운 요구사항이 생겨나고 있다. 이에 따라 HLA는 1998년 표준 형식인 HLA 1.3을 발표 하였고, 2000년에는 개선된 DDM(Data Distribution Management)영역과 개체 모델링 위한 XML 포맷을 도입한 IEEE1516을 발표 하였다. 또한 2010년에는 사용자와 운용자의 대표적인 요구사항을 받아드려 이를 기능화한 HLA Evolved가 발표 되었다. HLA Evolved는 현재의 HLA IEEE1516의 기능적인 한계와 문제를 개선하고자 한다.

[표 4] 새로운 HLA의 기능별 카타고리¹⁰⁾

	Develop	Deploy	Net-centric	Quality
FOM Modules	X	X		X
Extensible XML Format for OMT	X			
Schemas for OMT Compliance Testing	X			X
Federate Conformance in OMT	X			X
Encoding Helpers	X			X
Web Services (WSDL API)	X	X	X	
Dynamic Link Compatibility		X		
Standardized Time Types	X	X		X
Fault Tolerance		X		X
Smart Update Rate Reduction		X	X	
Improved Support for Data Logging	X	X		X
Evolved Transportation Types		X	X	
List Federation Executions		X		
Explicit Connect Statement		X	X	
Object names can be reused	X	X		
DoD Interpretations				X

10) Björn Möller, Katherine L Morse, Mike Lightner, Reed Little, Robert Lutz(2010) 『HLA Evolved A Summary of Major Technical Improvement』 U.S.A p.3

앞장의 [표4]는 HLA Evolved의 새로운 기능을 나타낸 것으로서 HLA Evolved의 새로운 기능은 Federation/Federate의 개발 시 빠르고 쉬운 환경을 제공하고, 오류발생의 요소들을 줄여줄 뿐만 아니라(Development features) Federation/Federate의 효율적인 배치환경을 제공한다.(Deployment features) Federation이 네트워크 중심의 어플리케이션에서 사용가능하게 되며, 이는 HLA의 Web서비스의 가능성을 뜻한다.(Net-centric features) 이러한 기능들은 개발자에게 표준안에서 어떠한 상황에서도 High-Quality Federation의 개발 환경을 제공하며 이는 High-Quality의 분산 환경시물레이션을 제공하며, 이는 전반적으로 HLA의 질의 성숙도의 향상으로 나타낸다.(Quality enhancement features)

HLA Evolved는 기존에 발표 된 HLA 1516에 비해 기능적인 측면에서 많은 변화가 생겼다. 그 중 대표적인 새로운 5가지 기능은 [표5]와 같다.

[표 5] HLA Evolved의 기능

기능	내용
Modular FOMs	Federation 개발자가 객체 모델(FOM)을 여러 개로 분할하여 사용 가능 Federate는 자신이 사용하는 FOM Modules만 필요
Web Service API	Web Service Description Language (WSDL) "binding" 지원 Web기반이라는 특징 외에는 C++, or Java binding과 유사
Fault Tolerance	Federation 실행 중에 특정 Federate가 네트워크 상에서 연결이 끊겼을 때, 참여 Federate들에게 이러한 상황을 통보
Smart Update Rate Reduction	RTI에게 지정한 update rate 이하로 데이터 전송 처리 요청 가능 Federate가 busy Federation에 안정적으로 참여 가능
Dynamic Link Compatibility	개발 시점에 사용된 RTI가 아닌 다른 벤더의 RTI를 사용하여 실행하고자 할 때 어플리케이션을 recompile 없이 실행 가능

1) Modular FOMs

FOM은 HLA1.3과 현재 HLA IEEE1516-2000에서도 HLA표준에 중요한 요소 중 하나이다. HLA Evolved는 이들을 기반으로 개발되어 졌기 때문에 HLA Evolved에서도 FOM은 여전히 중요한 요소이다. HLA Evolved에서는 새로운 FOM의 Concept을 제공하였는데 그 Concept이 바로 Modular FOM이다.

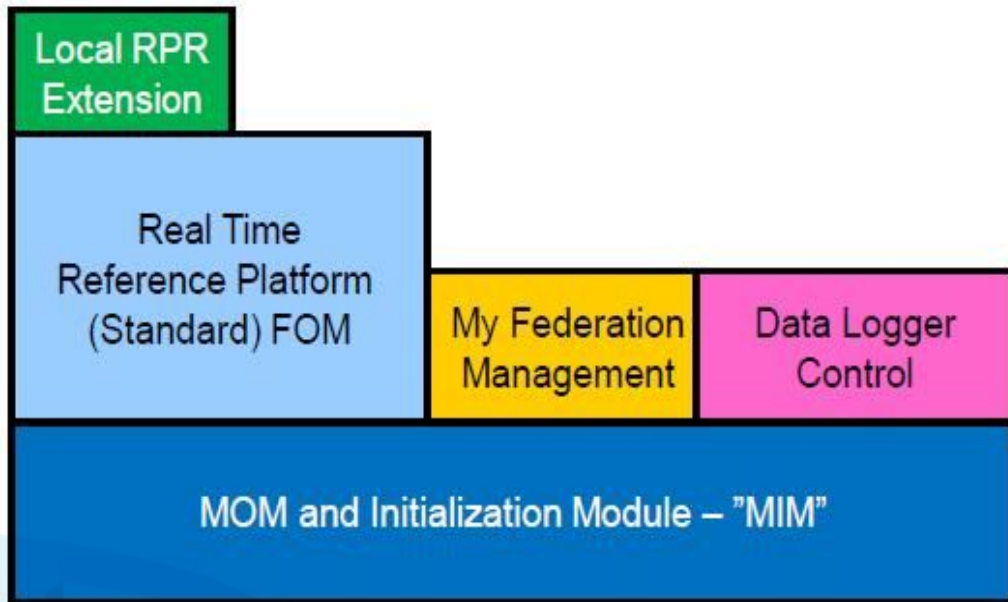
현재 IEEE 1516-2000은 FOM을 기술하기 위한 형식으로 XML구조를 채택하여 정보 표현의 표준화와 메타데이터 표준화를 제공하고 있으며, HLA Evolved 버전에서는 XML형식과 연동모델의 모듈화가 가능한 Modular FOM을 제공하였다. 모듈화는 기존 FOM의 Object Class나 Interaction Class의 계층구조를 각각 다른 FOM에 작성할 수 있으며, 최초 Federation 수행 시 적용된 FOM에 새롭게 가입하려는 Federate의 FOM 정보를 추가할 수 있다. Modular FOM은 FOM Modules, MOM Modules, SOM Modules와 같은 3가지의 구성요소를 갖는다.¹¹⁾

FOM Modules: FOM 정보를 포함하는 모듈을 정의하며, Federate의 FOM 모듈을 조립하여 Federation의 FOM을 생성할 수 있다. OMT 표준의 DIF(Data Interchange Formats) 스키마를 준수함으로써 사용자가 요구하는 기본 형식만을 기술할 수 있도록 지원한다. DIF 서식은 공통적인 데이터 모델로부터 도출된 형식으로 상호운용성과 활용성을 증진 시킬 수 있다.

MOM Modules: 기존의 1516-2000표준과 동일한 MOM(Management Object Model) 정보를 포함하는 모듈을 정의하며 확장 가능한 MOM 모듈을 제공한다.

SOM Modules: FOM 모듈과 동일하게 조립 기능과 표준 스키마를 제공한다.

11) 심준용 외(2009), “RTI기반 시뮬레이션의 상호운용성 향상을 위한 연동모델의 모듈화 방안에 대한 연구“, 『한국시뮬레이션학회지』, 한국시뮬레이션학회, p.141.



<그림 16> FOM 모듈 예시¹²⁾

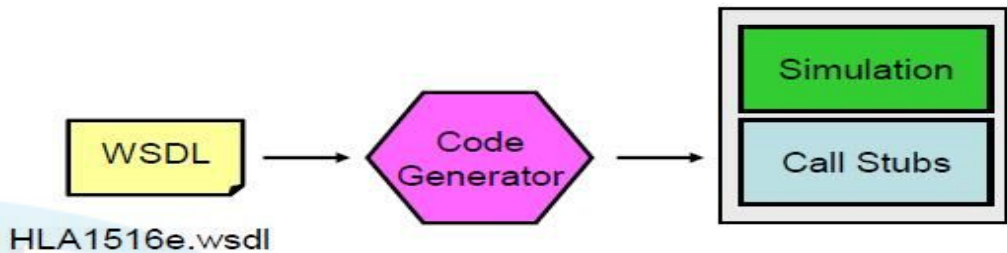
각 Federate의 FOM은 HLABase(MOM을 위한 표준모듈) 계층을 기반으로 하는 확장 RPR-FOM 모듈로 구성되어 있으며, 시뮬레이션 수행 시에도 Modular FOM에 제공하는 서비스를 통해서 지역 FOM을 추가할 수 있는 구조를 제공한다.

2) Web Service API

HLA Evolved는 HLA IEEE1516의 시뮬레이션 정보처리운용에 새로운 API를 제공한다. 이 새로운 API는 Web service기반을 제공 하는 것이다. 언뜻 보면 HLA와 Web service를 비슷하게 생각 할 수 있으나 두 가지 Architecture는 몇 가지 분명한 차이점을 보인다. Web Service는 LAN과 WAN을 통해 망을 형성하고 이를 통해 시스템들이 연결되는 메커니즘을 보인다. 반면 HLA는 시스템들 간 사이에 상호작용과 공유, 복잡한 데이터 통신

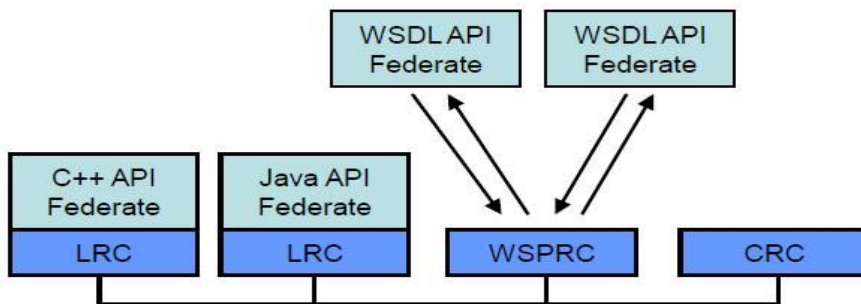
12) Björn Möller, Katherine L Morse, Mike Lightner, Reed Little, Robert Lutz(2010), 『HLA Evolved A Summary of Major Technical Improvement』, U.S.A p.16.

을 위한 고성능과 확장성을 제공한다. 또한 논리적 시스템 사이의 데이터 동기화의 기능을 제공하는 것도 특징이다. HLA Evolved의 Web Service API는 WSDL(Web Service Description Language) Language를 통해 구현된다. WSDL은 개발자가 실제 프로그래밍을 하는 것이 아니라 명세를 나타낸 것으로, WSDL API를 통해 개발자는 필요한 함수를 호출하고 코드를 생성 한다. 이는 WSDL가 제공하는 중요한 장점 중에 하나이다. 이러한 기능을 통해 개발자는 Wide한 플랫폼을 개발하고 배치를 한다.



<그림 17> WSDL의 코드생성¹³⁾

Federate는 각각의 LRC(Local Runtime Component)를 필수적으로 포함하고 있어야 이를 통해 RTI에 접속하도록 이루어 졌다. 하지만 WSDL은 LRC를 필수적으로 포함하고 있지 않더라도 LRC를 대신하여 WSPRC(Web Service Provider RTI Component)를 통해 RTI에 접속이 가능하다. 기존의 Federate는 각각 필수적으로 하나의 LRC를 포함하고 있는 것에 비해 WSPRC는 여러 Federate가 공유하여 사용 한다.



<그림 18> WSPRC를 통한 RTI 접속¹⁴⁾

13) Björn Möller, Katherine L Morse, Mike Lightner, Reed Little, Robert Lutz(2010) 『HLA Evolved A Summary of Major Technical Improvement』, U.S.A, p.18

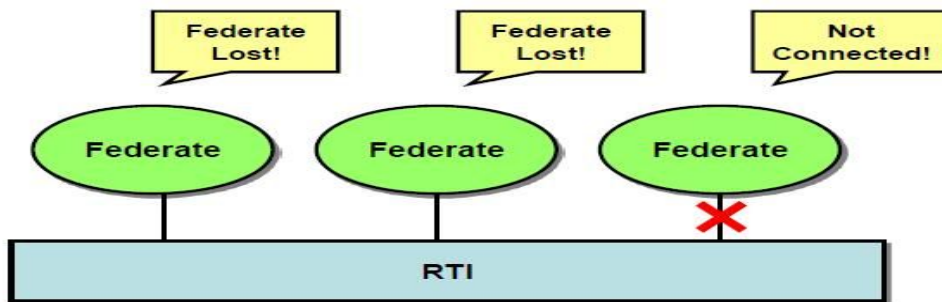
3) Fault Tolerance

Fault Tolerance는 원 뜻은 “시스템의 일부에 고장(fault)이 발생해도 요구되는 기능의 모든 것, 혹은 일부를 유지하는 기술 또는 방법”이다. 이러한 방식은 HLA에서도 요구되는 기능으로 일부 Federate가 connection오류 또는 Federate의 오류가 발생했을 시에도 Federation이 동작을 멈추지 않고 계속 유지 지속되는 것을 말한다. 현재 HLA기반에서는 Federation의 사이즈가 커짐에 따라 많은 오류가 발생하고 있다. 이러한 오류들을 발견하고 신호를 주어 판단하고 적절한 조치를 취해 완벽하지 않은 조건에서도 동작을 지속할 필요성이 있다.

Federate Lost: Federation내에서 예기치 않은 장애로 인해 Federate가 Lost될 때 발생하는 오류이다. 이 오류는 MOM과 상호작용을 통해 Lost 되지 않은 남은 Federate에게 “HLAreportFederateLost”의 신호를 준다. MOM interaction은 손실되기 전 정상적인 작동에 대한 timestamp정보를 제공한다.

Connection Lost: 이 오류는 C++ 및 Java로 FederateAmbassador를 호출하여, RTI의 Callback을 통해 오류의 신호를 준다. 이 Callback에의 문제의 Federate는 Not connection상태가 되고 RTI에 다시 연결을 하려고 시도하게 된다.

이 두 가지 신호에 의해 각각의 Federate는 <그림 19>와 같이 다른 Federate의 상태를 알 수 있고, 이를 통해 연결 상태를 점검하여 현재의 상황에 대처한다.



<그림 19> Fault Tolerance의 의미¹⁵⁾

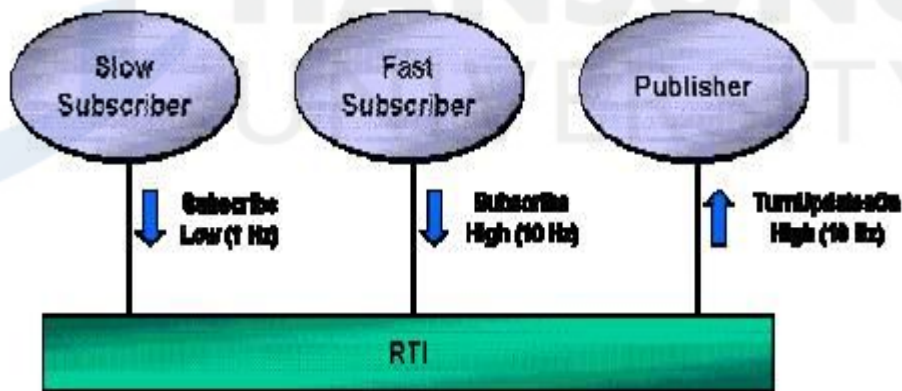
14) Björn Möller, Katherine L Morse, Mike Lightner, Reed Little, Robert Lutz(2010) 『HLA Evolved A Summary of Major Technical Improvement』,U.S.A, p.20.

15) Björn Möller, Katherine L Morse, Mike Lightner, Reed Little, Robert Lutz 2010 『HLA Evolved A Summary of Major Technical Improvement』, U.S.A, p.25.

4) Smart Update Rate Reduction

이전의 M&S의 상호운용 방식은 DIS와 같은 Broadcast방식을 사용했다. 이는 모든 시스템에 정보를 제공하는 것으로서, 원하던 원치 않던 모든 시스템이 정보를 제공받는 것이다. HLA 또한 이 Broadcast방식을 기초로 하는 공통 데이터 bus방식의 개념을 사용하였다. 이는 XML을 사용하여 exchange model의 정보를 공유함으로써 어떤 어플리케이션에서도 쉽게 적용이 가능하다는 장점이 있다. HLA에서는 정보처리의 과부하를 줄이기 위해 Declararation system과 데이터 분산 관리시스템 두 가지 시스템을 사용하였다.

이러한 방식은 update 주기율에 관한 몇 가지 문제점을 나타냈는데, HLA Evolved에서는 이를 보완하기 위해 SURR(Smart Update Rate Reduction)의 시스템을 제공하고 있다. SURR은 RTI에게 지정한 update rate 이하로 데이터 전송 처리 요청 가능하도록 하는 것을 뜻한다.



<그림 20> SURR 예시¹⁶⁾

<그림 20>은 SURR의 예시를 나타낸것으로서 낮은 갱신주기율을 동기화하는 하는 것은 다음 순서를 따른다.

16) Björn Möller, Katherine L Morse, Mike Lightner, Reed Little, Robert Lutz(2010), 『HLA Evolved A Summary of Major Technical Improvement』, U.S.A, p.27.

첫 번째, Slow subscriber는 attributes(best-effort)에 “LOW” update rate를 subscribe 한다.

두 번째, Publisher는 TurnUpdatesOnForObjectInstances를 callback하고 그 지시에 따라 “LOW” update rate를 update rate한다.

세 번째, RTI는 요청된 best-effort update를 넘어서는 maxiumrate에 대해서는 삭제를 한다. 이때 Publisher는 어떤 Rate를 통해 send update를 하는지는 영향을 미치지 않는다.

네 번째, Fast subscriber는 attributes(best-effort)에 “High” update rate를 subscribe 한다.

다섯 번째, Publisher는 TurnUpdatesOnForObjectInstances를 callback하고 그 지시에 따라 “High” update rate를 update rate한다.

여섯 번째, 이때 Slow subscriber는 “LOW” update rate 보다 높은 Rate를 update하지 않고, Fast subscriber는 “High”update rate 보다 높은 Rate를 update하지 않는다.

5) Dynamic Link

Evolved의 Dynamic Link Compatible는 SISO DLC의 표준을 기초로 만들어 졌다. 이는 사용자들이 서로 각기 다른 RTI를 이용하여 개발했을 시 서로 연동이 안 된다는 점을 개선하고자 제공된다. 개발자는 HLA Evolved에서 제공하는 EDLC(Evolved Dynamic Link Compatible)의 API 는 HLA Federates에 서로 다른 RTI가 호환이 가능하도록 하는 메커니즘을 제공한다.



<그림 21> Evolved Dynamic Link Compatible(one different RTI user)¹⁷⁾

17) Björn Möller, Katherine L Morse, Mike Lightner, Reed Little, Robert Lutz(2010), 『HLA Evolved A Summary of Major Technical Improvement』, U.S.A p.29.

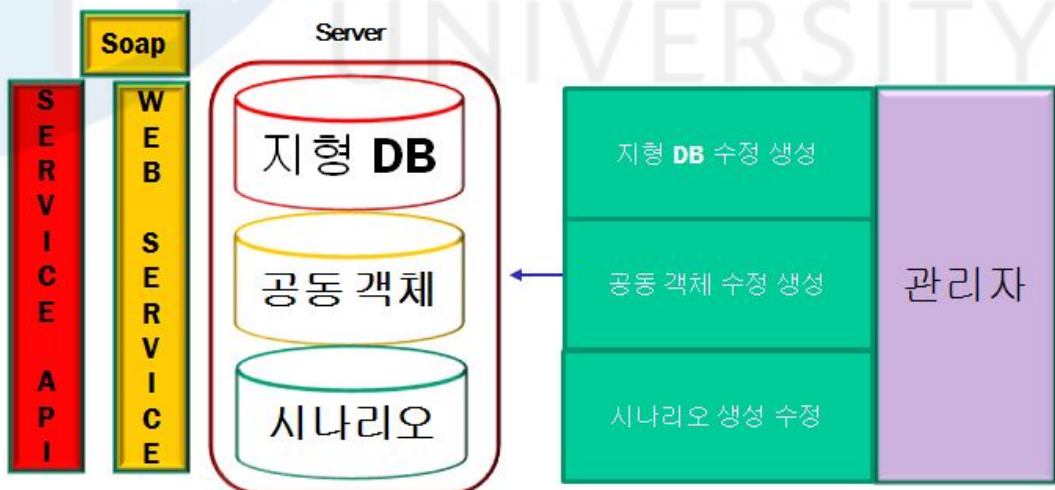
앞장의 <그림 21>은 Dynamic Link Compatible를 도식화 한 것으로서 공급자가 다른 RTI의 Library 만을 바꿈으로써 서로 다른 RTI를 쉽게 연동시킨다. 여기서 서로 다른 RTI란 제공자가 각기 다른 RTI를 의미한다. (예 MAK RTI, Pitch RTI)

EDLC는 Library 파일을 나타내며, C++과 Java language 모두 구현이 가능하고, 각기 다른 RTI가 설치 됐을 시 이중 한 가지를 선택을 하여 구현이 가능하다. 이는 개발자나 요구자 에게 좀 더 많은 유연성을 부여 한다.

2. Back-End

Back-End단은 Front-End단과 달리 연동에 중점이 맞춰져 있는 것이 아니라 서버의 개념이다. 즉 공통 지형DB 및 공통 객체, 공통 시나리오를 저장 하는 DB로서 V와 C에 이를 전송하는 역할을 한다. 서버를 구축하여 지형, 객체, 시나리오의 저장 및 수정, 로드 하는 역할을 말한다.

<그림 22>는 Back-End의 아키텍처를 나타낸 것 이다.



<그림 22> Back-End 아키텍처

과거 V-C 연동에서는 Virtual와 Constructive는 각자의 지형 DB 및 객체, 시나리오를 가지고 운용을 하였다. 이렇게 운용되던 Virtual 모델 또는

Constructive모델의 연동을 하게 되면 앞서 언급한 연동문제 뿐만 아니라 각자의 해상도의 차이 및 객체의 속성 등의 문제가 발생하게 된다. 즉 같은 위치 값을 가지고 있지만 서로의 지형 표현의 상세도 및 환경 묘사의 차이를 보인 다던가 같은 이름의 객체이지만 그 안의 속성이 값이 다르다는 문제가 나타난다. 즉 같은 객체의 속성값의 차이를 보이는 것이다. 기존에는 이러한 문제를 수정하기 위해 연동 시 지형 DB의 수정 및 객체의 동기화가 필수적으로 필요 했다. 하지만 본 논문에서 제안하는 Back-End 단을 구축 하게 되면 이러한 문제들은 더 이상 일어나지 않는다.

Back-End는 크게 세 가지 형태로 구분 할 수 있다. 첫 번째는 전송 프로토콜 방식, 두 번째는 구축 서버, 세 번째는 이를 관리하는 관리자 이다.

가. 프로토콜

첫 번째 전송 프로토콜 방식은 HTTPS¹⁸⁾를 이용한 Web service를 이용한다. Web서비스를 이용하면 로컬유저 뿐 만 아니라 Web service가 가능한 어떤 지역에서라도 손쉽게 사용 될 수 있다. 즉 국가와 국가간의 연동 시에도 같은 지형, 객체, 시나리오의 공유를 할 수 있다. 즉 예를 들어 한국과 미국의 V-C연동 및 C-C 연동, V-V의 연동 시 같은 Back-End를 이용하여 공통의 자원을 활용한 시뮬레이터 및 시뮬레이션의 구축이 가능하다는 것이다. 반면, Web service의 취약점은 바로 보안의 문제이다. 따라서 본 논문에서 제시하는 아키텍처는 HTTPS를 이용한다.

HTTPS는 하이퍼텍스트 전송 규약(HTTP) 계층 아래의 SSL¹⁹⁾ 서브 계층에서 사용자 페이지 요청을 암호화, 복호화하는 브라우저에 설치된 넷스케이프 웹 프로토콜. TCP/IP에서 HTTP 포트 80 대신에 포트 443을 사용하고, SSL은 RC4 스트림 암호 알고리즘용으로 40비트 키 크기를 사용한다. 넷스케이프 브라우저에서 https://URL로 페이지를 지정하면 HTTPS는 그것을 암호화하고, 도착된 https://URL은 HTTPS 서브 계층에서 복호화 된다. HTTPS와 SSL은 사용자의 송신자 인증을 위해 서버로부터 X.509 디지털 인증서 사

18) HTTPS: HTTP프로토콜과 SSL의 합성어로서 HTTP에 비해 보안부분이 강화되었다.

19) SSL: Secure Sockets Layer 암호화 기능을 통해 데이터를 송수신하는 프로토콜

용을 지원하는 방식이다²⁰⁾. HTTPS를 통하면 취약한 보안 문제는 보완 된다.

각 자원의 전송 시에는 비동기적 방식을 사용한다. 비동기적 방식이란 상대 방에서 필요로 하는 자원을 전송하여 주고 바로 연결을 끊는 방식이다. 이는 시스템의 안정화를 위한 방식으로 각각의 Virtual 또는 Constructive가 계속해서 Back-End와 연결이 되어 구동 시 Back-End에 예기치 않은 오류 및 종료가 일어났을 시 실행되는 Virtual 어플리케이션 또는 Constructive 어플리케이션의 영향을 주지 않기 위해서 이다. 예를 들어 Back-End가 비동기적 방식이 아닌 동기적 방식을 이용한다면, 지형 DB의 전송은 Streaming 방식이 될 것이고, 이는 객체의 움직임에 따라 계속해서 지형 DB를 올려 주는 방식이다. 이러한 방식으로 진행시 Back-End의 예기치 않은 문제가 발생 하게 되면 진행되는 시뮬레이션 및 시뮬레이터는 치명적인 문제를 일으킨다. 따라서 Back-End에서는 Virtual 또는 Constructive가 원하는 지형 DB 및 객체, 시나리오를 한번에 송신하고 연결을 끊어 버림으로서 이러한 문제점을 극복 한다.

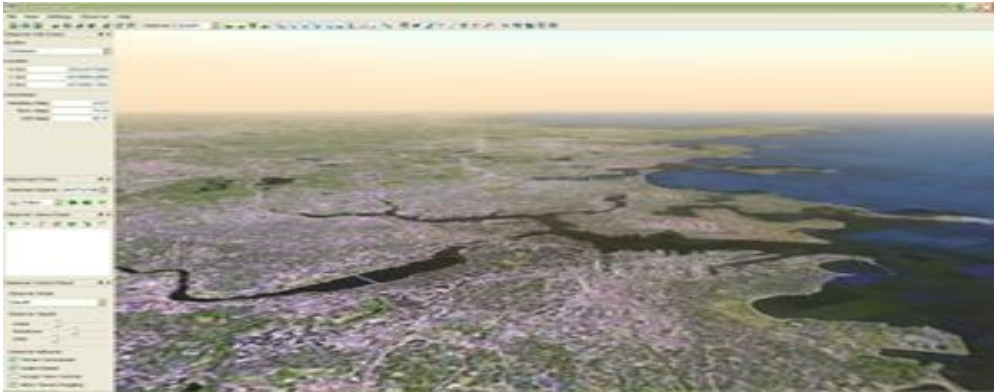
나. 서버 구축

구축서버는 세 가지의 자원요소를 갖게 된다. 첫 번째가 지형DB, 두 번째가 공통객체, 세 번째가 시나리오 이다. 이들을 개발 및 생성하여 공동 서버에 저장하여 관리한다는 것이 그 아이디어 이다. 공동 DB의 구축 방안에서 첫 번째가 되는 것은 기본적인 지형 Data Base를 구축하는 것이다. 이런 기본적인 지형 Data Base의 구축을 위해서는 COTS를 사용한다. 이런 기능을 제공하는 COTS 로는 MAK사의 VR-TheWorld와 Presagis의 CDB가 대표적이다. 이번 논문에서는 MAK사의 VR-TheWorld를 기반으로 기술 하고자 한다.

생성정차를 살펴보면 DB개발자는 필요한 위치를 지정하여 VR-TheWorld의 기본적인 지형 정보를 로드 한다. 그 후 필요한 해상도에 맞추어 DB를 수정하고 저장한다. 즉 예를 들어 탱크의 시뮬레이터를 구성하는 지형

DB가 필요로 한다면 상세한 지형 DB가 필요로 하게 되는데, 이때 기본적인 VR-TheWorld의 지형에 상세 지형을 맵핑하여 구현 한다.

20) 한국정보통신기술협회 용어정리



<그림 23> VR-TheWorld

즉 1m 고도 이미지, 1m 급 위성이미지를 활용 하여 맵핑을 한다는 것이다. 기타필요요소의 건물 및 도로 등의 지형 DB 요소는 모델링 Tool을 활용하여 구현 하도록 한다.

VR-The World는 기본적으로 전 세계의 지형 데이터를 가지고 있다. 전 세계 모든 지형에 대해서 90M급(DTED 1 Level)의 고도 값을 가지고 있으며, 15M급의 위성영상을 포함하고 있다. 이 DB를 기본적으로 활용하여 DB개발자는 사용자의 필요한 지형 DB를 생성 해낸다.



<그림 24> VR-TheWorld를 활용한 예

앞장의 <그림 24>는 VR-The World에서 기본적으로 제공되는 데이터를

기초로 하여 별도의 지형 수정, 편집이 가능한 도구를 이용해 사용자의 목적(전차 시뮬레이터)에 맞게 지형 DB를 수정, 보완한 지형 DB이다.

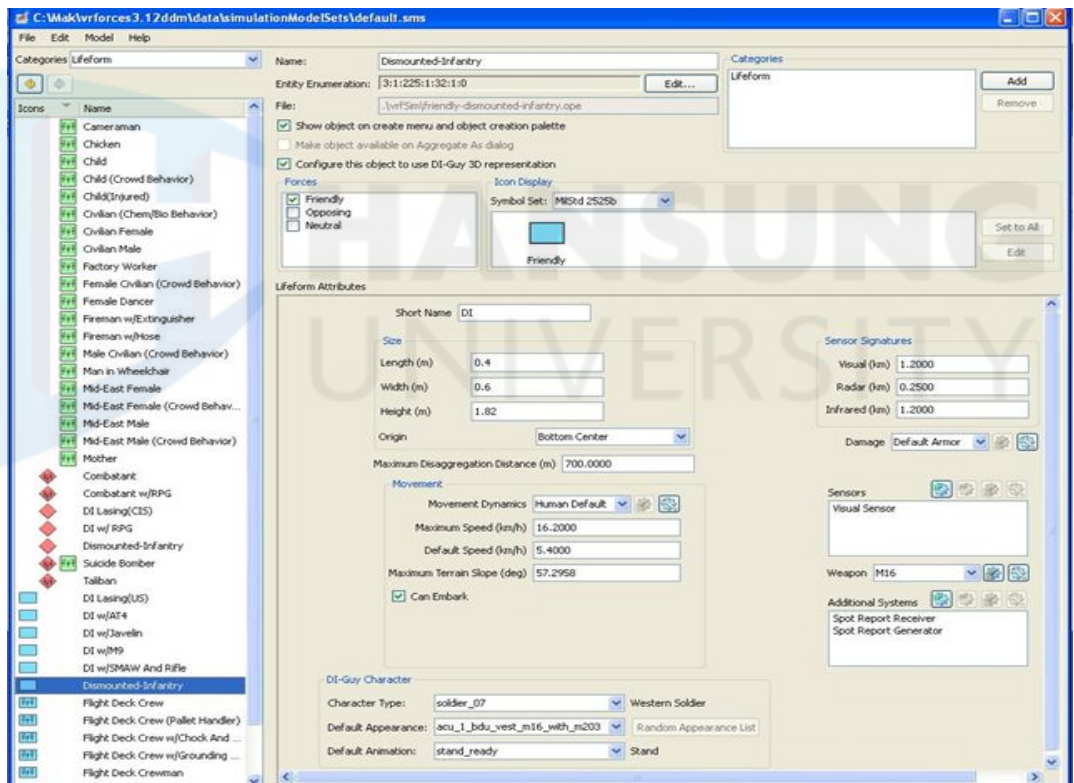
이렇게 수정 및 보완 작업을 거친 DB는 공통 서버에 저장 되게 된다. 이때 저장 시 지형 DB에 고유의 경로를 정의하여 저장 한다면, 이 DB가 어떤 어플리케이션에 사용되었고, 그 어플리케이션을 이용 시 어떤 지형 DB를 불러올릴지를 식별이 가능하게 된다.

이런 방식의 공동 DB를 이용 시 재사용성 또한 높인다. 예를 들어 K1 전차의 시뮬레이터 구축 되어있고, K21 장갑차의 시뮬레이터의 구축이 필요 시 K21과 K1전차는 같은 지상무기체계이기 때문에 지형DB 해상도 부분에서는 크게 차이나 나는 것이 없을 것이다. 따라서 K21 시뮬레이터의 구축 시 K1와 같은 훈련장에서의 훈련이라면 별도의 지형DB 작업 없이 이미 구축 되어 있는 K1의 지형 DB를 이용 한다. 현재 VR-TheWorld는 기본적으로 Streaming Terrain²¹⁾ 방식이다. 그렇게 때문에 실제 제안하는 아키텍처의 구현을 위해서는 이 Streaming 방식을 비동식으로 바꿔주는 작업이 필요하다.

두 번째는 공동 객체의 구축이다. 이는 각기 다른 시뮬레이션 모델들이 각각의 객체를 별도로 보유하고 있을 시 객체의 표현이나 객체의 속성 값이 다를 때 연동을 할 때 그 객체들을 전부 수정해야하는 불편함을 없애기 위함이다. 공동 객체를 구성하게 되면 서버에서 각각의 객체를 보유하여 관리하기 때문에 객체의 표현이나 속성 값이 일치 하게 된다. 이는 연동 시 별도의 수정 작업 없이 객체를 사용 할 수 있음을 뜻한다. 객체의 공통화를 위해서는 각각의 객체의 표현을 구현 할 수 있는 프로그램과 객체의 속성을 입력할 수 있는 별도의 프로그램이 필요하다. 이 또한 별도의 개발 없이 COTS를 통해 활용이 가능하다. 본 논문에서는 MAK사의 교전급Constructive모델인 VR-Force에서 기본적으로 제공하는 Entity Editor를 이용한다. Entity Editor는 기본적으로 야군과 적군, 중립요소의 카타고리를 갖고 있으며, 트리 구조를 통해 Air Craft, Ground vehicle, Navy unit등의 카타고리를 갖고 있다. 이러한 트리 구조의 분류를 통해 사용자

21) Streaming Terrain: 객체가 이동하는 것을 예측하여 그에 상응하는 지형 데이터를 올려준다

또는 개발자가 쉽게 접근 할 수 있도록 하는 GUI를 제공한다. 이 GUI를 통해 사용자 또는 개발자는 각기 필요한 객체에 속성값을 입력함으로써 객체를 관리하게 된다. 이러한 공동 객체 제외하고 군사 기밀의 보안 객체이거나 보안이 필요한 특정 객체들은 사용자의 어플리케이션에 따로 저장하여 사용하거나 공동 DB의 특정 폴더를 생성하여 따로 관리 할 수 있다. 공동객체의 구축의 또 한 가지의 장점은 공동지형 DB와 마찬가지로 재사용성을 들 수 있다. 새로 개발이 필요한 시뮬레이터의 시나리오의 생성 시 별도의 객체생성을 위한 작업이 필요 없이 공동 객체 DB를 이용하여 객체를 구현하게 된다.



<그림 25> Entity Editor

<그림 25>은 Entity Editor를 나타낸 것으로 사용자는 객체의 기본적인 Parameters의 값을 입력하고, 무장의 선택 및 무장의 Parameters까지 입

력 할 수 있게 되어 있다.

객체의 표현은 주로 모델링 도구를 이용하여 객체를 표현하게 된다. 이 객체의 표현은 2D 뿐만 아니라 3D의 표현도 가능하다.



<그림 26> 객체표현의 예시

<그림 26>은 3D객체의 표현을 예로 나타낸 것으로 개발자 또는 사용자가 COTS를 이용하여 객체를 쉽게 제작 및 수정을 할 수 있다. 위에서 보이는 3D 뿐만 아니라 2D 객체의 표현, 단대호의 표현 또한 제작 및 수정이 가능하다.

세 번째는 공동 시나리오 구축이다. 공동으로 시나리오를 구축하여 서로 연동 시 시나리오 따른 예기치 않은 오류를 줄인다. 시나리오 파일은 파일 형태로 저장 되게 되고, Virtual 이나 Constructive의 연동으로 시 서버에 시나리오를 요청하게 되고 이에 공동 시나리오는 각기 다른 모델에 같은 시나리오를 전송 하게 된다. 비동기적 전송 방식을 이용하여 시나리오를 전송하면 시나리오를 받은 모델들은 바로 시나리오를 시작하고 수행 한다.

다. 관리자

관리자는 이렇게 구축된 공동 지형DB, 공동 객체, 공동 시나리오 서버를 관리하는 역할을 한다. 관리자는 필요에 따라 공동 지형DB, 공동 객체, 공동 시나리오 수정 및 저장할 수 있게 된다. 따라서 권한을 부여받은 개발자 및 사용자가 관리자가 될 수 있음 뜻한다. 이에 관리자는 DB에 Path를 활용하여 자신만의 최적화된 자원을 관리할 수 있다. 또한 통합적으로 관리하는 관리자는 지형 DB 및 공동 객체, 공동 시나리오의 추가 및 수정 보완할 수 있는 권한이 부여된다.

제 3 절 현 V-C 체계와 비교분석

본 논문의 3장의 1절에서는 현재 V-C연동의 기본적인 연동 방법을 기술하였다. 현재 V-C연동에 있어서 몇 가지 제약사항을 여러 모듈을 조합한 하나의 Adaptor를 구성함으로써 연동을 시키고 있다.

본 문에서 언급한 4가지 연동 한계점 외에도 연동에 관한 여러 문제점이 존재한다. 본 문에서 언급한 연동 문제점은 가장 일반적이고 보편화된 문제점을 제시한 것이고 그 문제에 대해 현재 어떻게 대처하고 있는가를 나타낸 것이다.

3장 2절에서는 앞서 언급한 문제점을 현재의 방법이 아닌 COTS와 HLA Evolved을 이용함으로써 보다 향상된 V-C 연동의 아키텍처를 제안하였다. Front-End와 Back-End방식으로 나누어 Front-End에서는 연동의 문제점의 해결방안을 제시하였고, Back-End 방식은 Web기반의 공통 서버를 구축하고 로서 V-C연동에 가장 큰 문제점 중 하나라고 할 수 있는 지형 DB 및 객체의 이질화에 대한 방안을 제시하였다.

본 3절에서는 앞서 언급한 현재 V-C연동 체계와 본 문에서 제안한 V-C연동의 차이점을 기술한다. 이는 앞선 1절과 2절의 요약하여 비교하여 기술하는 것이다.

[표 6]는 현재 V-C 연동 체계와 본 문서에서 제안한 연동의 차이점을 요약하여 나타낸 표이다.

[표 6] 현 V-C 체계와 비교분석

연동의 이슈	현재 V-C 체계	제안된 V-C체계
이기종간의 아키텍처 연동 문제	개발자가 각각의 아키텍처의 특성에 맞게 변환자를 개발	COTS인 VR-Exchange를 이용하여 자동적으로 연동
이기종 RTI 간의 연동	Wrapper를 구축하여 각각의 RTI library를 포함하도록 개발자가 모듈을 개발	COTS인 VR-Exchange를 이용하여 자동적으로 연동 HLA Evolved API를 이용하여 Library를 선택하여 연동
갱신을 동기화	버퍼를 이용하여 가장 낮은 갱신율에 동기화가 되게끔 개발자가 모듈을 개발	HLA Evolved의 API를 이용하여 자동적으로 갱신을 동기화 구현
공동 합성 전장 환경	개발자는 모의 해상도 모듈을 개발 연동되는 DB를 전체적으로 수정 연동되는 객체를 전체적으로 수정 복잡한 FOM의 수정 CGF의 합의	검증된 CGF기반의 COTS를 이용 Web기반의 공동지형DB, 공동 객체, 공동 시나리오 구축하여 통합관리자가 관리 HLA Evolved의 Modular FOM을 이용

위의 표에서 나타나는 것과 같이 현재 V-C 체계에서는 연동의 문제가 되는 것을 개발자들이 연구 분석하여 그에 따른 연동 모듈을 개발함으로

서 연동을 가능하게 하고 있다. 이런 노력과 시간, 비용을 줄이기 위해 이미 검증된 COTS를 이용과 HLA의 문제점을 개선하고자 나온 HLA Evolved의 기능을 활용하는 것이 효율적이고 경제적인 대안이다.

제 4 절 COTS기반 연동의 한계점

V-C연동에서 연동의 문제점을 해결하고자 별도의 Adaptor를 개발하는 것에 비해 COTS를 이용하게 되면, 개발 인원 및 기간을 단축시킨다. 또한 이미 검증된 COTS를 사용하기 때문에 안정성면에서도 Adaptor에 비해 높다 할 수 있다. 하지만 이러한 COTS의 이용 또한 문제점이 있다. COTS를 개발한 회사에 기술적으로 의존해야하는 점, 제품의 업그레이드와 오류 및 버그의 발생 시 개발회사를 통해 해결해야한다. 또한 초기 도입 비용이 부담으로 작용 할 수 있다. 제품의 유지보수 또한 비용부담을 초래 할 수 있다. 일반적으로 개발한 회사와 1년 단위 또는 분기단위별로 비용을 지불하는 형식의 유지보수를 계약을 하는 형태로 진행 된다.

종합적으로 보면 COTS의 이용은 기간의 단축 및 인원의 감축, 안정성면에서는 큰 장점으로 작용하나, 기술의 의존과 도입, 유지보수의 비용은 단점으로 작용한다.

제 4 장 COTS 기반의 V-C 연동구축 사례

제 1 절 V-C연동 개요

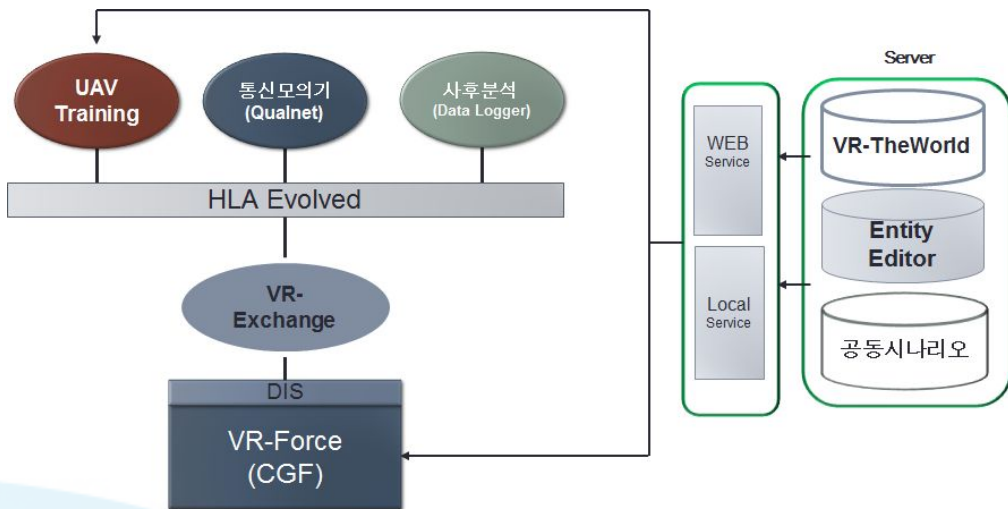
본 4장에서는 3장 2절에서 제안된 V-C 연동 체계의 아키텍처의 검증을 위해 실제 HLA Evolved 기반으로 COTS를 사용하여 구축한다.

이 구축되는 시스템은 Virtual 모델인 UAV Training과 CGF 기반의 Constructive의 연동, 객체간의 통신을 시뮬레이션 하는 통신 모의기, 시나리오 종료 후 이를 분석할 수 있는 사후 분석기, 공통 지형DB와 객체, 시나리오를 송신하는 서버의 구성으로 이루어진다. 이 시스템에서의 COTS의 사용은 합성전장 환경을 구현하는 Constructive 모델에 VR-Force, Virtual 모델을 구성하는 요소 중하나인 영상 처리작업을 하는 VR-Vantage, UAV HMI를 구현하는 GL-Studio, HLA기반의 연동에 필요한 미들웨어인 MAK RTI, 이기종 아키텍처 간의 연동 제약 사항을 해결 할 수 있는 VR-Exchange, 공동 DB를 제공하는 VR-TheWorld, 객체간의 통신 시뮬레이션을 구현하는 Qualnet, 사후분석에 쓰이는 DataLogger 로서 구성을 한다.

본 V-C연동에서 사용되는 각각의 COTS들은 이미 전 세계적으로 널리 사용되는 검증된 COTS이다. 앞서 말한 것처럼 이들을 사용하게 되면, 개발의 기간이 단축되고 시스템의 안정화, 유지보수의 편의성을 보장한다. 실제 본 V-C 구축은 제안된 아키텍처와는 현재 환경상의 제약 사항 때문에 제한적으로 구축한다. Front-End단의 연동에서는 이기종 프로토콜간의 연동을 포함하고, 이기종 RTI간의 연동은 배제한다. Back-End에서는 실제 Web기반의 구축제한으로 인해 Web Service가 아닌 로컬 프로토콜을 이용한다.

본 V-C연동의 개략적인 운용 시나리오는 UAV Training인 Virtual 모델을 통해 UAV에 탑재된 스캐닝 카메라를 운용자가 조작하여 전장상황을 스캐닝하고 스캐닝을 통해 적을 식별하여 커맨드센터에 정보를 보내고 이를 커맨드센터에서 판단하여 적의 파괴 명령을 내리는 시나리오이다.

제 2 절 V-C 연동의 구축



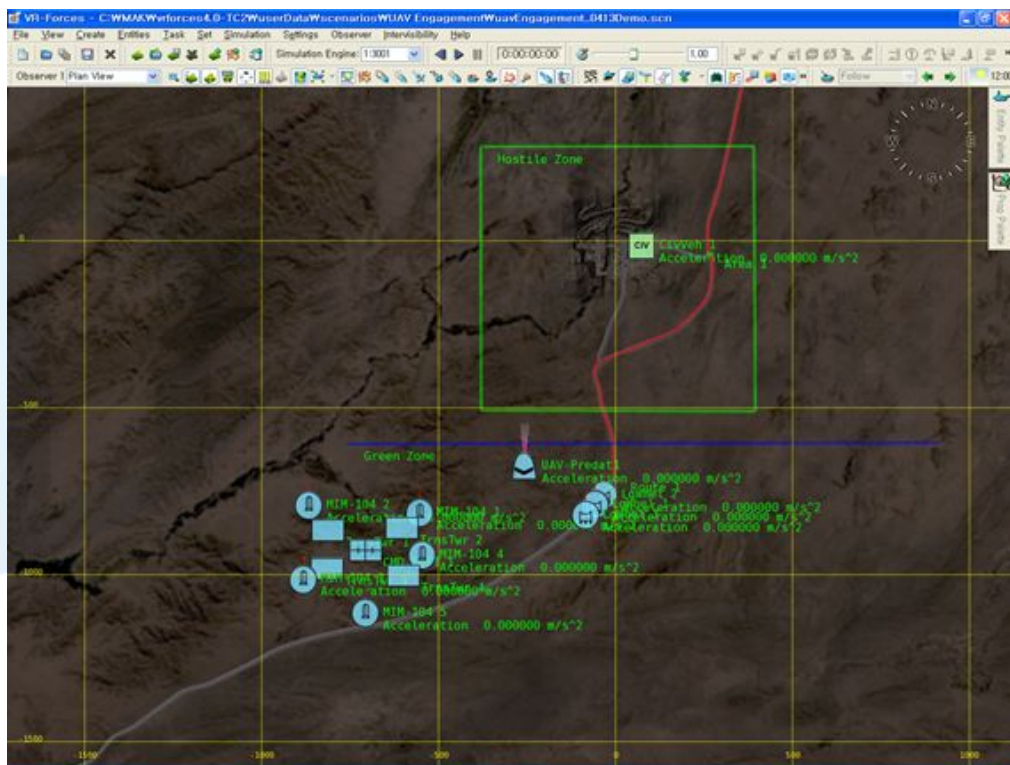
<그림 27> V-C 연동 아키텍처

<그림27>은 본 논문에서 구축하는 V-C연동 사례에 대한 아키텍처이다. Front-End단에서는 Virtual모델인 UAV Training과 통신모의기, 사후분석기가 HLA Evolved기반으로 구성이 되어 있고, Constructive모델인 VR-Force는 DIS기반으로 구성되어있다. HLA Evolved와 DIS는 서로 다른 아키텍처임으로 이들의 연동을 위해 VR-Exchange를 이용하였다. Back-End단에서는 공통DB의 구축을 위하여 VR-TheWorld를 이용하였다. VR-TheWorld에서 기본적으로 제공되는 지형 DB를 활용하여 본 시나리오에 필요한 전장상황을 구현하였다. 이 지형 DB는 시나리오 시작 전 각각의 모델에 Local Service를 통해 공통적으로 전송되어진다. 또한 공통 객체를 위해서는 Entity Editor를 이용하였으며, 지형 DB와 마찬가지로 시나리오 시작 전 Local Service를 통해 전송되어진다. 시나리오는 별도의 서버를 두었으며, 이는 Web Service를 통해 각각의 모델에 전송 된다.

1. Constructive 모델

Constructive의 모델은 교전급 모델인 MAK사의 VR-Force를 이용한다. VR-Force는 교전급의 Constructive 모델로서 CGF기반의 전장 시나리오를 제작, 실행하는 Tool로서 미국 MAK의 대표적인 COTS Tool 이다.

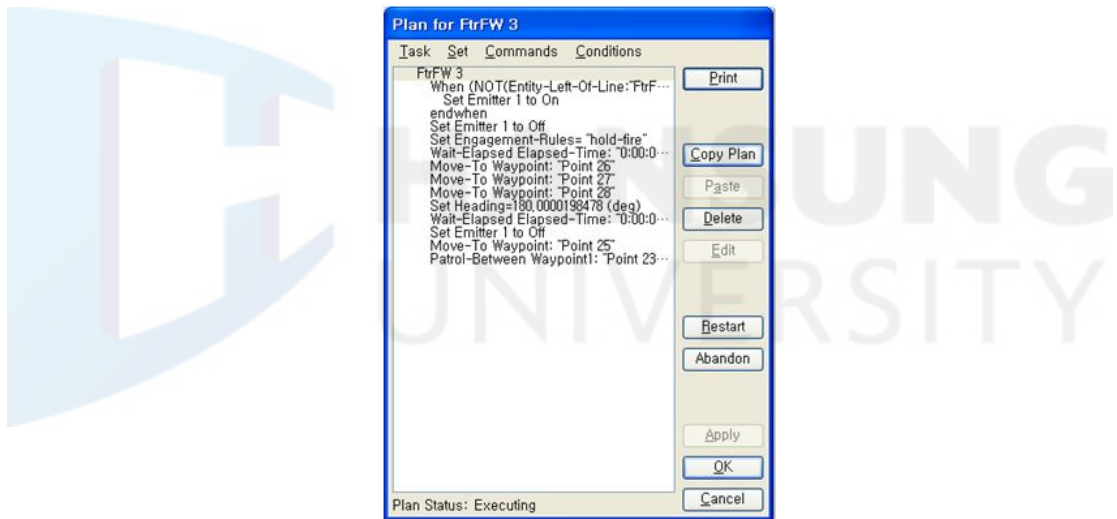
다양한 센서의 표시 및 객체 모델의 지원, 보기 쉬운 합성 전장 환경에 적합함을 지니고 있다. 2D/3D 지형과 객체의 표현이 모두 가능하다.



<그림 28> VR-Force 상황도

<그림 28>은 VR-Force상에 나타나는 상황도를 나타낸것으로서 객체나 지형을 나타내고 있다. VR-Force는 부대와 단위 개체의 표현과 각각의 객체의 표현이 모두 가능하다. 이는 VR-Force를 Constructive모델로 사용시 Virtual과 생기는 해상도를 별도의 처리 없이 가능하게 한다.

기본적으로 GUI와 엔진이 나누어져 있어 객체의 처리 및 이벤트 처리의 퍼포먼스를 최적화 시킨다. GUI와 Engine이 떨어져 있다는 것은 GUI 한 개와 엔진을 여러 개 연동시켜 구성 할 수 있다는 것을 의미 하며 각각의 엔진별로 별도의 작업을 수행 할 수 있게끔 한다는 것을 의미한다. 객체의 관리는 VR-Force에서 기본적으로 제공되는 Entity Editor를 통해 객체의 속성이나 속성값을 입력함으로써 손쉽게 객체를 관리 한다. 시나리오 생성 시 객체의 Plan과 Task는 <그림 29>에서 나타는 것과 같이 객체별로 Plan과 Task를 주어 실행이 가능하다. 이러한 임무수행 능력은 곧 인공지능을 뜻하는데 이는 B-HAVE나 AI-IMPLANT를 VR-Force의 하나의 모듈처럼 사용하여 처리 가능하게 한다.



<그림 29> Plan과 Task 부여

기본적으로 분산 시뮬레이션에 최적화가 되어있기 때문에 DIS, HLA 1.3, HLA 1516, HLA Evolved를 기본적으로 지원하며, RPR-FOM을 표준으로 갖고 있다. 이와 같은 편리성으로 인해 Constructive모델로서 VR-Force를 선정 하게 되었다.

본 논문에서 구축한 V-C연동에서의 Constructive모델은 CGF기반으로 적군과 아군을 생성하고 객체마다 각각의 Plan과 Task를 부여하고, 전체

적인 전장 상황도를 디스플레이한다.

2. Virtual 모델

본 문에서 구현하는 Virtual 모델은 UAV Training이다. 이 Virtual 모델은 MAK사의 VR-Vantage IG를 통해 영상 처리작업을 수행 하였으며, 이 영상처리 작업을 통해 전장 환경의 기상변화나 타임 오브 데이등의 환경적 요소를 가시화 하였으며, IR맵핑이 이루어진 지형 DB와 객체에 IR / NVG의 효과를 부여하였다. 또한 DISTI사의 GL-Studio를 이용하여 손쉽게 1인칭 시점의 Cockpit을 구현하였다.



<그림 30> IR 맵핑 처리된 DB

<그림 30>은 IR 맵핑 처리된 DB를 UAV에 탑재된 카메라를 모의한 것으로 운용자가 바라보는 화면이 된다. 이때 운용자는 여러 가지 모드로서 카메라를 운용하게 된다. 위치와 시간에 따라 IR, NVG, TV모드를 사

용하여 전장상황을 스캐닝하게 되고, 필요에 따라서는 카메라의 Zoom/Out 기능을 이용한다.



<그림 31> Virtual 모델의 운용

<그림 31>은 운용자가 실제 조이스틱을 통해서 Virtual 모델을 운용하는 모습이다.

UAV Training은 3 DOF로 운용되며, 조이스틱을 통해 통제 된다. 사용자는 UAV 위치 통제뿐만 아니라 이 UAV에 탑재된 카메라 또한 통제가 가능하다. 이 사용자는 카메라를 이용하여 전장상황을 스캐닝 한다.

본 논문에서의 Virtual 모델인 UAV Training은 전장상황을 스캐닝 하고 스캐닝 된 정보를 지휘통제실에 전송한다. 전송 시 통신모의기와 연동되어 환경에 따른 신호 송신율/수신율을 시뮬레이션 한다. 이는 실제 전장상황에서 나타나는 주변 환경에 따른 통신상태 등의 모의가 가능하다.

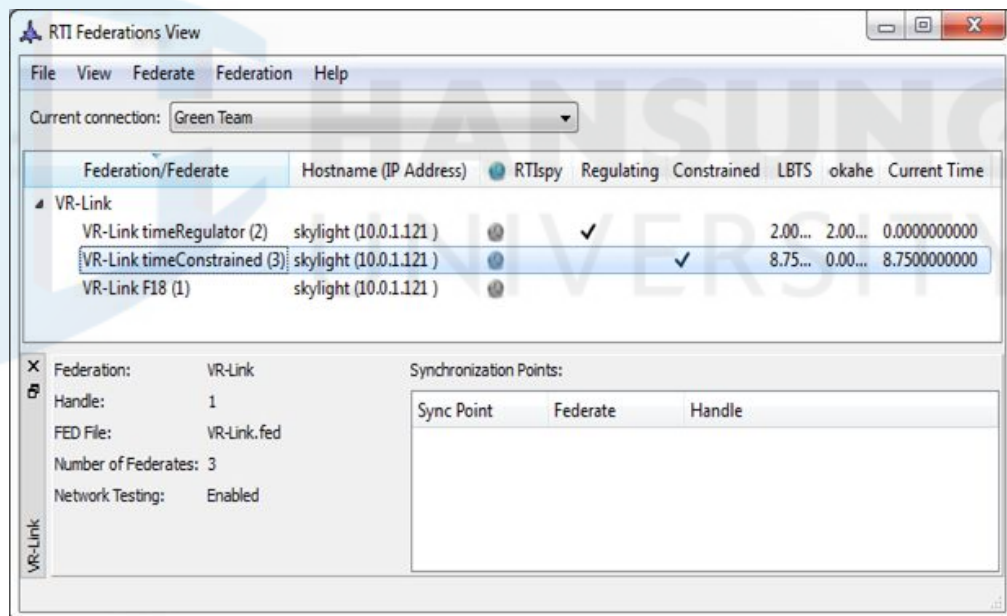
3. 연동 모델

V-C 체계의 연동모델은 MAK사의 RTI와 VR-Exchange를 사용한다. RTI는 HLA기반의 분산 시뮬레이션 환경에서 선택이 아닌 필수적으로 꼭

필요한 미들웨어 이다. HLA기반의 분산 시뮬레이션은 RTI를 통해 데이터 통신이 이루어지기 때문이다. 상용화된 RTI는 MAK RTI 뿐만 아니라 Pitch RTI, VTC RTI등이 있다. 이들, 서로 다른 공급자가 제공하는 RTI는 현재로서 별도의 모듈이 없는 한 서로 연동 되지 않는다. 이에 연동을 위해 VR-Exchange를 사용한다. VR-Exchange는 이기종 프로토콜간의 연동도 지원한다. 이러한 연동을 지원하는 COTS는 현재 VR-Exchange가 유일하다는 것이 선정의 이유이다.

MAK RTI의 선정 이유는 VR-Force와 공급사가 같아 연동이 보다 손쉬운 점도 있겠지만 현재 국내에서 가장 널리 이용되기 때문이다.

RTI S/W는 현재 국내에서도 개발에 성공 되었나, 아직 상용화 단계는 아니다. 따라서 이미 검증을 마치고, 상용화 된 MAK RTI를 사용한다.



<그림 32> MAK RTI

<그림 32> MAK RTI는 개발자나 사용자가 바라보는 GUI화면으로 각 각의 Federate의 상태를 모니터링 할 수 있다. 3장에서 언급 했듯이 VR-Exchange는 서로 다른 연동 아키텍처, 이기종 간의 연동 시 사용 된

본 논문에서는 이 두 가지 연동 모델을 사용 할 것이며, HLA Evolved 기반으로 구축을 한다.



본 논문에서 VR-Exchange는 HLA Evolved와 DIS 기반의 어플리케이션을 연동하는데 사용되었다. HLA Evolved기반으로 개발된 어플리케이션은 MAK RTI를 사용함으로써 구현되었다.

4. Server

본 논문에서 구축한 V-C 사례에서는 Virtual 모델과 Constructive모델은 모두 공통DB와 객체, 시나리오를 사용하였다. 이는 본 문에서 제시한 아키텍처의 Back-End단에 해당하는 것으로서 지형 DB는 VR-TheWorld를 사용하였고, 객체는 Entity Editor를 이용하였다. 시나리오는 Constructive모델인 VR-Force에서 생성한 시나리오를 공통적으로 사용하였다.

VR-TheWorld에서 기본적으로 제공되는 지형데이터를 이용하여 임의의 지형에 대해 Virtual에 필요한 지형 DB를 제작하였다.



<그림 34> VR-TheWorld Server

<그림 34>은 VR-TheWorld 서버의 실제 모습이며 이런 서버 형태로 제공 된다. 사용자 또는 개발자는 이 서버를 개발 PC 또는 운용PC에 연결함으로써 VR-TheWorld 서버가 제공하는 기본적인 지형 DB 데이터를 이용 할 수 있게 된다.

제 3 절 V-C 연동 구축 결론

COTS와 HLA Evolved를 사용하여 V-C연동을 구축을 해본결과 연동의 문제에 있어서는 개발자의 별도의 코딩과 Recompile없이 손쉽게 연동되는 모습을 확인하였다 또한 서버에서 지형 DB, 객체, 시나리오를 로드하는 작업도 손쉽게 구현되었다. 본 구축에서는 지형 DB와 객체와 같은 무거운 데이터들은 Local Service를 이용하였고, 시나리오는 Web Service를 통해 로드하는 프로토콜을 구축하였다. 추후 구축에서는 모든 데이터를 Web Service를 통한 프로토콜 구축이 이루어 져야 할 것이다.

COTS와 HLA Evolved의 통한 V-C 연동 구축은 구축기간이 짧고, 별도의 코딩이 필요 없는 장점이 있으나 기존에 개발된 어플리케이션과 연동문제 및 비용 문제가 한계점으로 나타났다. 개발된 어플리케이션과의 연동에서는 표준인 RPR-FOM의 사용이 아닌 경우 FOM의 전체적인 수정이 필요하며, 이는 상당한 시간을 소요하게 된다. 비용의 문제는 사용된 모든 어플리케이션이 COTS이기 때문에 이를 구입하여 개발하여하는데 이는 상당부분의 예산이 필요할 것으로 예상된다.

제 5 장 결 론

현재 국방M&S에서 가상훈련체계의 훈련방법으로서 제시되고 개발된 모델은 Live, Virtual, Constructive 로서, 이러한 각각의 개발된 훈련체계는 많은 한계점을 들어낸다. 이러한 한계점을 극복하기위한 하나의 수단은 이러한 각각의 훈련모델을 연동을 통해 한계점을 극복하는 것이다. 이는 바로 LVC 연동체계를 구축하는 것이다. 이러한 이유에서 우리에게 LVC 연동은 선택사항이 아닌 필수사항이다.

각각의 모델은 서로 다른 목적과 아키텍처로 구성되어 있다. 이러한 각각 다른 모델을 하나로 연동하는 것은 간단한 일 아니다. 이에 따라 연동에 관한 많은 연구가 있었고 그에 따른 많은 연동 방법론이 제시 되었다. 본 논문은 이러한 연동을 위한 하나의 방법론으로써 COTS와 HLA Evolved를 그 기반을 둔다.

COTS(Commercial, Off-the-shelf)란, 상용화된 S/W로 검증을 마치고 제품화 되어 출시되는 S/W이다. COTS의 가장 큰 장점으로서는 이미 검증이 되었다는 부분과 다양한 적용사례가 있다는 것이다. 이는 시뮬레이션의 안정화를 보장하는데 큰 기여를 한다. 뿐만 아니라, 시뮬레이션의 개발기간을 단축시키는 장점 또한 있다. 반면, 개발업체에 의지를 해야 한다는 점과 도입 비용에 대한 부담이 있다는 것이 단점이다.

본 논문에서 연동에 사용된 COTS인 VR-Exchange는 이미 검증을 마친 상용도구 중 하나로서 이기종 아키텍처 또는 공급자가 다른 RTI 간의 연동 가능성을 제시하여 본 논문의 V-C연동 구축에 적용하여 사용하였다.

HLA Evolved는 현재 분산 시뮬레이션환경에서 가장 많이 사용되고 있는 아키텍처중의 하나인 HLA의 최신 버전으로서 기존의 HLA의 한계점을 극복하고자 HLA에 비해 개선된 아키텍처이다.

본 연구를 통해 연동에 사용된 COTS를 바탕으로 RPR-FOM의 적용과 HLA Evolved의 적용으로 인해 향상된 V-C 연동의 아키텍처를 제시 하고 이를 기존에 구현된 아키텍처와 비교 분석하였다. 또한 COTS를 이용하여 DIS

와 HLA의 이기종 아키텍처간의 연동을 구축하는 사례를 보였다.

본 연구는 과거 Adaptor를 이용한 V-C연동에 비해 COTS를 활용한 V-C 연동은 개발 기간의 단축, 개발 인원의 감축에 대한 경제적인 효과 뿐 아니라 이미 검증된 COTS를 사용함으로써 전체적인 시스템의 안정화를 높여 주는 신뢰성 효과, 지형DB 및 객체, 시나리오를 공통으로 관리를 함으로서 재 사용성 증진에 대한 효과, Webservice를 통한 전송방식 사용함으로써 지역에 관계없는 확장성 등 여러 측면에서 기존 V-C 연동보다 향상된 방법론임을 확인했다.

LVC 연동은 대한 방안은 국내 연구개발이 필수적이며 보다 많은 연구와 개발 사례를 통해 보다 안정적이고 발전적인 개발방안을 정립하는 것이 필요하다. 이러한 측면에서 본 연구가 향후 국내 연동체계 발전에 기여하기를 기대한다.

향후 연구에서는 V-C의 연동 방안뿐만 아니라 LIVE까지 포함 된 L-V-C 구축에 관한 향상 된 연동방안의 제시가 필요할 것으로 예상 된다.

【참고문헌】

1. 국내문헌

[단행본]

김문수, 김대규, 권혁래, 이태억(2012), 『소부대 전술 훈련을 위한 개체기반 위
게임모델과 전차시뮬레이터 연동에 관한 연구』,
한국군사과학기술학회지

김숙영, 안정현, 성창호, 김탁곤(2010), 『Virtual-Constructive 시뮬레이션 연동
기술에 관한 연구』, 한국 군사과학기술학회 종합학술대회

심준용, 조원섭, 진정훈, 김세환(2009), 『RTI기반 시뮬레이션의 상호 운용성
향상을 위한 연동모델의 모듈화 방안에 대한 연구』, 시물
레이션 학회지

안정현, 홍정희, 성창호, 김탁곤(2008), “IEEE 1516 HLA/RTI 기반 연동
Adaptor의 설계 및 구현”, 『종합학술대회』, 한국과학기술원

윤근호, 심신우, 이동준(2012), “Battle lab에서의 상호 운용성을 위한 LVC 연
동방안”, 『한국시뮬레이션학회 논문지』, 한국시뮬레이션학회

최상영(1999), “시뮬레이션 고수준 아키텍처 HLA”, 『대한 산업 공학회 춘계학
술 발표 논문집』, 한국경영과학회

홍정희, 안정현, 김탁곤(2008), “IEEE 1516 HLA/RTI 표준을 만족하는 시간
관리서비스 모듈의 설계 및 구현”, 『시뮬레이션
학회지』, 한국시뮬레이션학회

[논 문]

김성호(2010), “Virtual-Constructive 시뮬레이션 연동체계 구축 방안에 관한
연구”, 배재대학교 석사학위논문

윤석준(1998), “항공기 시뮬레이션 기술의 진화”, 세종대학교 항공 산업연구소

2. 국외문헌

Jim Kogler(2010), 『*HLA_Evolved_An_Introduction*』, MAK Webinar ,U.S.A
Pitch Technology(2010), 『*HLA-Evolved Tutorial*』, U.S.A
U.S. Department of Defense(1998), 『*High-Level Architecture Rules
Version 1.3*』, IEEE P1516/D1, April 1998s U.S.A
MAK Technology(2009), 『*VR-Exchange User guide*』, U.S.A
Björn Möller, Katherine L Morse, Mike Lightner, Reed Little, Robert Lutz
(2010), 『*HLA Evolved A Summary of Major Technical Improvement*』, U.S.A



ABSTRACT

A Study on COTS Based Interoperability Architecture for Virtual and Constructive Simulation

Kim Kyung Soo

Major in National Defense Modeling&Simulation

Dept. of National Defense Modeling&Simulation

Graduate School of National Defense Science

Hansung University

In the Defense Modeling and Simulation (DM&S), Many models have been developed with various L-V-C model and Adaptor for simulation interoperability.

However, the derived results with the limited interoperability simulation are limited in the simulation in Korea. To overcome the limitation, we propose the COTS based interoperability architecture for virtual and constructive simulation. In this paper, the proposed framework is enabled to apply the exist models. The models to apply in this paper are the virtual and constructive models, the models were developed at different RTI and different purpose. Through the proposed interoperability architecture based COTS, we perform the simulation interoperability of two models.

By using the COTS, the reduction effect of the personnel and duration for the V-C Interoperability is expected and it is also expected to bring about the improvement of the stability and reliability of the V-C Interoperability system. In addition, it is expected to bring about the enhancement of the reusability,

expandability, stability, and reliability of the data when building the V-C interface architecture presented in the body.

【Keyword】 L-V-C, COTS, HLA, HLA Evolved

