

High Level Architecture - 시뮬레이션의 새 지평

윤 석 준*

〈 목 차 〉

I. 배경

II. 미 DoD의 HLA 정책

III. HLA의 기술적 내용

요 약

HLA(High Level Architecture)라는 생소한 전문 용어가 시뮬레이션 사회에 처음 소개된 시점은 1996년에 접어들어서 였다. 본 기고에서는 미국의 국방성(DoD: Department of Defense)이 주도하고 있는 HLA가 탄생된 배경과 지금까지의 진화 과정, 그리고 HLA의 기술적 내용과 DoD의 HLA에 대한 정책에 관하여 비교적 상세히 살펴보고자 한다.

I. 배 경

가. 정책적 배경

미 DoD가 HLA를 주도하게 된 정책적 배경을 요약하면 다음의 3가지로 정리된다.

* 세종대학교 항공우주공학과

HLA가 현실적으로 강제된 첫번째 동기는 미 국방 예산의 지속적인 삭감이다. 미군의 규모는 삭감되는 국방 예산에 따라 축소될 수 밖에 없고, 그럼에도 불구하고 세계의 유일한 경찰국가임을 자임하는 미국은 전세계에 여전히 많은 군대를 배치하고 있다. 또한, 미 군사력의 전시준비성을 유지하기 위한 가장 효율적인 수단인 야전 훈련은 소요되는 막대한 비용 때문에 필요한 만큼 자주 수행되기가 어렵다. 즉, 미 DoD는 예전처럼 국방 예산을 더 이상 원하는 만큼 자유롭게 집행하기가 어렵게 되었고, 자원의 투자를 결정한 경우에도 그 당위성과 효율성을 꼼꼼히 챙겨야 할 만큼 심각하게 예산의 압박을 받고 있다는 것이다. 이에 대한 가장 효과적인 해결책으로 제시된 것이 HLA였고, HLA의 응용 개념은 시뮬레이션 기술을 무기체계의 획득에서 운영 그리고 후속지원, 연구개발 전단계에 적용하여 자원상의, 또한 비용상의 낭비적인 요소를 최소화하겠다는 것이다. 물론, 이러한 예산상의 동기가 HLA로 바로 이어지지는 않았으며, 전래의 시뮬레이션에서부터 HLA로의 진화 과정은 2장에서 상술하고자 한다.

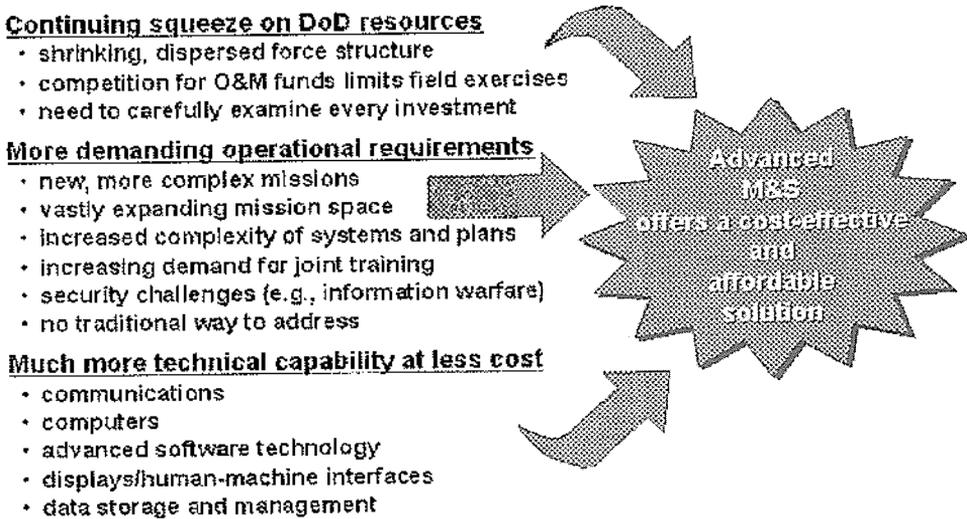
HLA가 요구된 두 번째 동기는 더욱 까다로워진 군사 작전상의 요구 사항들이다. 미·소 양극체계가 붕괴되면서 미국의 군사력이 수행하여야 하는 작전 임무들은 절대적인 적이 없이 그때 그때의 정치적인 상황에 따라 국지적으로 전개될 수밖에 없다. 즉, 미군이 수행하여야 하는 작전 임무들은 더욱 복잡 미묘해지고 전에 없던 새로운 국면을 맞게 되었다는 것이다. 또한, 세계의 유일한 경찰국가가 된 미국은 자국의 이해에 따라 자국의 군사력을 전세계 곳곳에 파견하고 있다. 그만큼 작전 지역이 광범위하게 확대되었다는 의미이다. 까다로워진 작전 임무 조항은 이에 그치지 않는다. 군사 작전에 투입되는 체계와 계획은 전에 없이 복잡해졌고, 합동 훈련에 대한 요구도 더욱 요구되고 있다. 여기에 보안상의 문제들도 이에 상응하여 난제로 대두되었다. 이러한 작전 임무 상의 까다로워진 문제들은 전통적인 훈련 방식으로는 해결될 수가 없다. 실제 합동 군사 훈련을 통해서도, 고립된 시뮬레이션 또는 시뮬레이터 훈련으로도 불가능하다. 그 해법으로 제안된 것이 바로 HLA인 것이다.

HLA가 탄생될 수 있었던 세 번째 배경인 자신감은 관련 기술의 진보로부터 비롯된다. 통신 기술과 컴퓨터의 발달은 구태여 언급할 필요가 없을 만큼 지난 10여년간 비약적으로 발전하여 왔다. 그 기간 중 과거에는 상상할 수도 없을 고급 컴퓨터의 성능이 매우 경제적으로 이용 가능케 되었고, 이에 따라 시뮬레이션의 영역과 응용 범위는 엄청나게 확대되었다. 발달된 통신 기술은 독립적인 시뮬레

이션 자원들을 실시간 또는 비실시간으로 묶는데 이미 성공하였으며, HLA는 여기에 만족하지 않고 향후 다가올 초고속 통신 시대를 염두에 두고 다양한 시뮬레이션들이 통합된 거대한 가상 공간을 계획하고 있다.

이상과 같이 HLA가 탄생된 배경을 까다로워진 미 국방 예산의 압박, 까다로워진 군사 작전상의 요구 사항, 기술적 자신감 등으로 요약하여 보았다. 이에 대한 이해를 돕기 위하여 <그림 1>에 HLA 탄생 배경을 도시하였다.

<그림 1> HLA 탄생 배경도



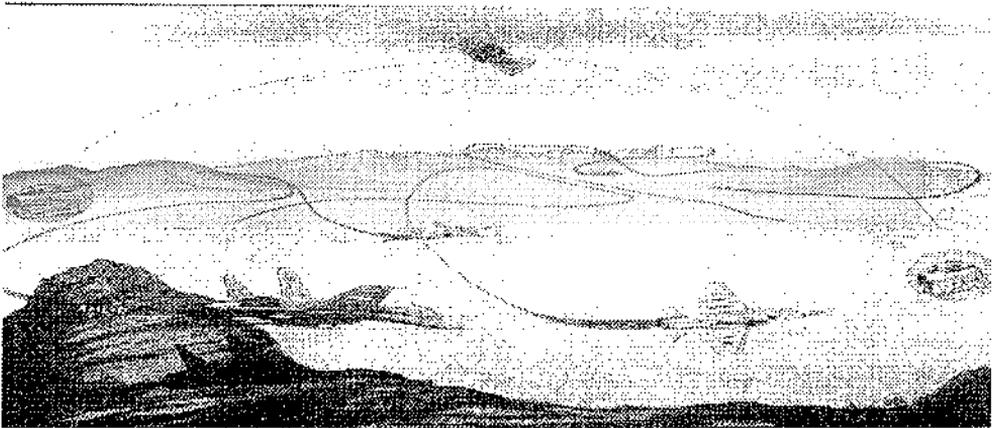
나. 기술적 배경

HLA의 기본 골격은 이미 1983년경에 미국의 국방성(DoD: Department of Defence)이 SIMNET(SIMulator NETworking)이라는 프로그램을 시작하며 도입되었다. SIMNET이 시작된 1983년까지도 미 DoD는 훈련용 시뮬레이터를 비행이나 사격 또는 조준과 같은 기술을 개인 차원에서 훈련시키기 위한 독립적인 고 정밀도의 장치로 인식하여 왔다. 이러한 고급 장비들은 지금까지도 개별적인 훈련에 매우 긍정적인 역할을 담당하고 있으며, 실제 장비의 훈련을 상당 부분 대체하는 효과를 가져왔다. 하지만, 이러한 고가의 장비들은 시간당 훈련비용 면에서 낭비적인 요소를 여전히 안고 있고, 보다 중요한 전략과 작전과 같은 팀 차

원의 훈련을 담당하지 못한다는 한계를 갖는다. 이러한 목적으로 활용되는 훈련 방식으로는 War Game과 실제 작전 훈련이 있는데, War Game의 경우는 지휘관에 초점이 맞추어져 있고 현실감이 떨어지는 비실시간 시뮬레이션이고, 실제 작전 훈련은 병력의 수송, 훈련 계획, 비용, 안전, 보안, 후속 지원, 환경 등의 면에서 난제가 산적해 있다. 즉, 개인이 아닌 팀 훈련을 위하여 War Game 보다 효과적이고, 실제 작전 훈련 보다 현실적인 훈련 방식의 필요성이 미 DoD에 대두되었고, 이 필요성은 SIMNET 프로그램을 탄생시키게 된 계기를 만들었다.

SIMNET 프로그램은 고급분산시뮬레이션(ADS: Advanced Distributed Simulation)의 공식적인 첫 단계라 할 수 있으며, 그 개념과 경험은 후에 DIS(Distributed Interactive Simulation)와 ALSP(Aggregate Level Simulation Protocol)로 맥을 이었으며, 계속 진화하여 HLA로까지 이어진다. 본 장에서는 HLA의 본질을 보다 명확하게 이해하기 위하여 그 진화 과정을 소개한다.

〈그림 2〉 고급 분산 시뮬레이션의 개념도



SIMNET 프로젝트

미 DoD 산하의 DARPA(Defense Advanced Research Projects Agency)가 주도하여 1983년에 시작하여 1989년에 막을 내린 SIMNET프로젝트는 시뮬레이션 계통에서나 미 DoD에서 수행한 여러 핵심 사업 중에서도 매우 중요하고 획기적인 프로그램이었다. 동일한 컴퓨터 플랫폼을 기반으로 한 수백 대의 탱크

시뮬레이터들이 동일한 장소에서 지역통신망(Local Area Network: LAN)으로 연결되어 가상의 공간을 창출함으로써 팀 차원의 전술 및 작전 훈련을 치밀한 사전 준비와 계획하에 성공적으로 수행할 수 있었다. 대당 약 \$250,000 정도의 저급 시뮬레이터들은 운동판(Motion Platform)을 장착하지 않았을 뿐 아니라, 영상장치도 320×128 pixel의 저해상도에 화면 갱신율이 15Hz에 불과하였고, 조종장치도 약식으로 처리하였으며 조종실(hull) 내장도 fiberglass로 단순히 처리하였다. 결코, 실제 탱크의 특성과 운용 환경을 충실하게 구현하였다고 할 수 없는 비교적 조악한 시뮬레이터들로 구성된 SIMNET 프로젝트가 이후의 시뮬레이션 기술의 물결에 던진 화두는 무엇이였을까? 그것은 이후의 DIS(Distributed Interactive Simulation), ALSP(Aggregate Level Simulation Protocol), HLA(High Level Architecture)로 이어지는 고급분산시뮬레이션(ADS: Advanced Distributed Simulation)의 가능성이다. Virtual, Constructive, Live 등 다른 종류의 시뮬레이션들과 다양한 컴퓨터 플랫폼, 통신 매체들로 구성된 대규모의 유연한 합성 환경(Synthetic Environment)을 창출하여 팀의 합동 훈련은 물론 무기체계 전반의 획득, 운용, 후속 지원, 연구 개발의 전 과정을 일관성 있고 효과적으로 관리하고자 하는 고급분산시뮬레이션에 대한 미국 DoD의 노력과 자신감은 바로 이 SIMNET 프로젝트의 성공을 바탕으로 하는 것이다.

DIS (Distributed Interactive Simulation)

SIMNET 프로젝트가 종료된 해인 1989년 5월에는 SIMNET에서 배운 경험을 토대로 사람이 조종하는 단순 탱크 시뮬레이터 간의 통합 환경하에서 수행하는 팀의 훈련 차원을 벗어나, 그 개념을 어떻게 진일보 시킬 수 있을 것인가 하는 주제로 심포지움이 개최되었다. 그 해 후반에는 미국 Florida의 Orlando에서 국방 시뮬레이션 간의 상호 운용성(Interoperability)을 지원하기 위한 표준을 탐구 하기 위하여 워크숍이 개최되었고, 이 워크숍은 1년에 2회 개최되는 일련의 DIS(Distributed Interactive Simulation) 워크숍으로 96년까지 이어졌다. 초기의 워크숍은 SIMNET을 주도하였던 DARPA가 후원하였으나, 곧 미 육군의 STRICOM(Simulation, Training, Instrumentation Command)으로 후원 기관이 대체되었고, University of Central Florida의 부설 기관인 IST(Institute for Simulation and Training)가 워크숍의 기술과 행정을 실질적으로 지원하게 되었다. 그 결과 1990년 6월에는 시뮬레이션들 간의 주고 받는 데이터(PDU:

Protocol Data Unit)에 대한 형식과 내용이 포함된 DIS 표준의 청사진이 발표되었다. 이 초안은 이후 업계, 정부, 학계 등의 참여를 통한 무수한 토론과 개정을 거쳐 1993년에 IEEE 1278-1993으로 공인되었다.

IEEE 1278-1993이 발표되었을 시는 이미 표준에 포함되어 있는 많은 정보가 시대에 뒤져 있었고, 무선통신, 전자파, 시뮬레이션 관리 등을 새로이 포함시키고자 하는 PDU의 Version 2가 개발되고 있었으며, 환경 데이터와 Live Simulation을 지원하기 위한 기능들에 대한 PDU도 연구되었다. 한편으로는, DIS 표준규격을 반영한 DARPA의 War Breaker 프로그램과 같은 DIS 시연 행사들이 다수 시행되었고, 이 행사들을 통한 경험들은 DIS 표준의 문제점들을 노출시켜 개선안들을 제공하였다. 이러한 노력들로 1995년에는 DIS의 PDU규격 Version 2로 IEEE1278.1-1995가 탄생되었고, PDU에 의해 요구되는 통신 서비스에 대한 표준으로 IEEE1278.2-1995가, DIS시행에 따른 관리와 feedback에 대한 규격으로 IEEE1278.3이, 평가와 인증에 대한 표준으로 IEEE1278.4가 차례로 공표되었고, 마지막으로 DIS에 사용되는 충실도의 분류 기준으로 IEEE1278.5가 1997년 8월중에 심사되었다.

ALSP (Aggregate Level Simulation Protocol)

항공기 시뮬레이터와 같은 실시간 Virtual Simulation을 중심으로 하여 Live Simulation과 Constructive Simulation들을 접목하며 가상 공간의 영역을 확장해 가던 DIS와는 달리 Discrete Event Simulation을 위주로 집단 단위의 분산 위 게임(Distributed War Game)을 시도하는 시험이 DIS 워크숍들이 첫 단추를 끼우던 1989년 가을에 시행되었다. DARPA가 후원을 한 ACE-89라 명명된 이 시험은 집단 단위의 분산 위 게임의 가능성을 제시하였으나, 기술적인 단점들을 표출하였다. DARPA는 이러한 종류의 시뮬레이션들간의 상호 운용성 문제를 해결하는데, SIMNET의 기본 원칙들이 적용될 수 있는가를 검토하기 시작하였다.

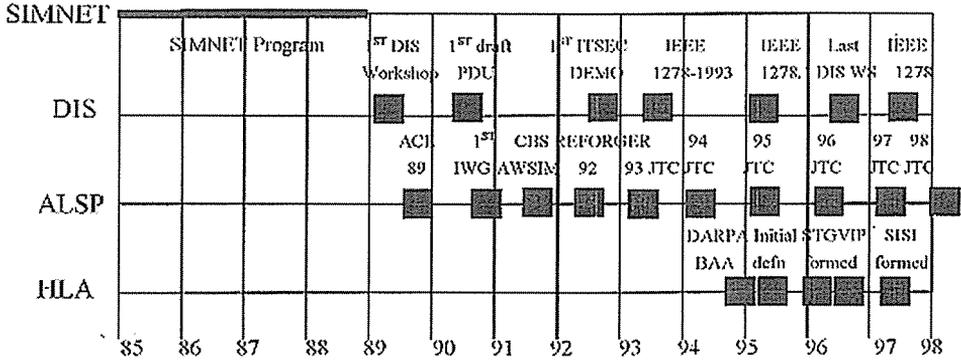
SIMNET의 분산 컴퓨터(No Central Node), 지리적 분산(Geographic Distribution), 오브젝트 소유권(Object Ownership), 메시지식 규약(Message-Based Protocol) 등 4가지 기저 원칙들은 Constructive Simulation에도 그대로 적용될 수 있음이 확인되었다. 하지만, 실시간 Virtual Simulation과의 기본기술상의 상이점 등 다양한 사유로 인하여 Constructive Simulation의 상호 운용성을 위해서는 시간 관리(Time Management), 데이터 관리(Data

Management), 구조(Architecture) 상의 요구 사항들이 추가로 적용되어야 함이 드러났다. 이러한 ALSP 개념을 진전시키기 위하여 1990년 12월에는 학계, 산업체, 정부의 시뮬레이션 전문인력들의 컨소시엄 형태로 구성된 Interface Working Group이 결성되었고, 1991년 1월에는 WPC의 지상 전투 시뮬레이션인 GRWSIM의 여러 copy로 구성된 연합체(Confederation)의 원형이 시연되었으며, 그 해 6월에는 GRWSIM과 미 공군의 공중전 시뮬레이션인 AWSIM 등 2가지 다른 시뮬레이션들을 묶는 시도가 있었다. 이외에도 ALSP개념을 적용시키고 발전시키기 위하여 다양한 Constructive Simulation 들의 연합 프로그램들이 시행되었는데, 이 중에는 미군과 한국군의 합동 작전 훈련인 1992년 8월의 울치 포커스(Ulchi Focus Lens)도 포함된다.

이러한 일련의 작전 훈련들의 성공은 ALSP를 Constructive War game들 간의 상호 운용성을 위한 골격으로 자리잡게 하였으며, 미군의 많은 주요 시뮬레이션 훈련 센터들에서 시뮬레이션 도구로 채택하고 있는 Joint Training Confederation(JTC)는 ALSP를 적용한 대표적인 시뮬레이션 연합체이다. 1997년에 시행된 JTC에는 미군의 육군, 해군, 공군, 해병 등에서 사용되는 총 12개의 War Game들이 연합되었으며, 후속 지원, 첩보, 전자전 등의 기능들이 또한 새로이 포함되었다.

1990년에 접어들며 미 DoD는 광범위한 영역에서 모델링과 시뮬레이션 기술을 적용할 필요성을 절감하게 되었다. 이 영역에는 단순 훈련을 넘어 연구 개발, 획득, 시험 평가, 개념 시연 등이 모두 포함된다. 이러한 활동을 조직적으로 지원하기 위하여 미 DoD는 DMSO(Defense Modeling and Simulation Office)을 같은 해에 결성하게 되었고, 이후 DMSO는 미 DoD가 모델링과 시뮬레이션 기술의 개발과 지원을 전세계적으로 선도하는데 중추적인 역할을 담당하게 된다. 미 DoD는 DIS와 ALSP를 통하여 배운 경험과 기술적 진보를 기반으로 모든 종류의 M&S(Modeling & Simulation)의 상호 운용성을 보장하기 위한 골격을 제공하는 통합된 고급 분산 시뮬레이션(ADS: Advanced Distributed Simulation)을 시도하게 되었고, 수많은 원형(Prototype) 제작 활동을 거쳐 마침내 High Level Architecture(HLA)가 1996년에 탄생되었다. SIMNET으로부터 DIS, ALSP, HLA로 이어지는 변천 과정이 <그림 3>에 도시 되어 있다.

〈그림 3〉 고급 분산 시뮬레이션의 진화



II. 미 DoD의 HLA 정책

미 국방성(DoD)의 HLA 관련 정책을 파악하기 위해서는 먼저 HLA보다 상위 개념인 모델링/시뮬레이션(M&S: Modeling & Simulation) 전반에 대하여 DoD가 어떠한 비전(Vision)을 갖고 있는지를 살펴볼 필요가 있다. DoD는 작전 개념상으로 유효한 어떠한 환경이라도 DoD의 어느 부서가 필요로 할 경우 항상 미리 준비가 되어 있도록 M&S 기술을 계속 발전시켜야 할 것으로 믿고 있다. 여기서 언급된 환경의 용도 또는 M&S의 적용 범위는 합동 훈련에서부터 교의와 전술의 개발, 작전 계획의 수립, 전시 상황의 평가, 기술 평가 지원, 체계 개량, 원형 및 실물 크기의 개발 등까지를 망라한다. DoD는 이미 M&S에 대한 종합기본계획을 수립하여 1995년 10월 17일자로 승인을 받았는데, 이 계획을 살펴보면 미 DoD가 계획하는 M&S의 적용 범위를 보다 구체적으로 감지할 수 있을 것이다(그림 4). 미 DoD는 이러한 환경을 창출하기 위한 방편으로 구체적인 세부 목표를 또한 수립하였다. HLA도 M&S의 궁극적인 비전을 성취하기 위한 하나의 단계이며, 중심 되는 기반임을 〈그림 5〉에 나타나 있는 DoD의 M&S 관련 종합기본계획을 통하여 확인할 수 있다.

이제 M&S에서 시각을 좁히어 DoD의 HLA에 대한 정책을 살펴보자. 1996년 9월 10일 미 DoD의 M&S 관련 최고 책임자인 Dr. Paul Kaminski는 다음과 같은 정책을 발표하였다: "Under the authority of DoD Directive 5000.59, and as prescribed by [the DoD Modeling and Simulation Master Plan], I desig-

Ⅲ. HLA의 기술적 내용

HLA는 미 DoD의 모든 시뮬레이션 응용에 관련된 기능적 요소(functional elements), 공유 영역(interfaces), 설계 규정(design rules) 등으로 구성되며, 특정 체계 구조물(system architecture)들이 정의될 수 있는 공유 골격을 제공한다. 즉, HLA는 DoD 시뮬레이션의 기술적 구조물(technical architecture)라 정의할 수 있다. <그림 6>은 HLA의 구성을 기능적 관점에서 표현한 것이며, 그 구성 요소들을 요약하면 다음과 같다.

HLA Rules

Federation 내 시뮬레이션들 간의 적절한 상호작용을 성취하기 위하여 적용되는 일련의 규정들로서 HLA federations 안의 시뮬레이션들과 runtime infrastructure에 대한 책임들을 기술한다.

Interface Specification

Runtime infrastructure와 시뮬레이션들 간 공유 영역의 기능들을 정의한다.

Object Model Template

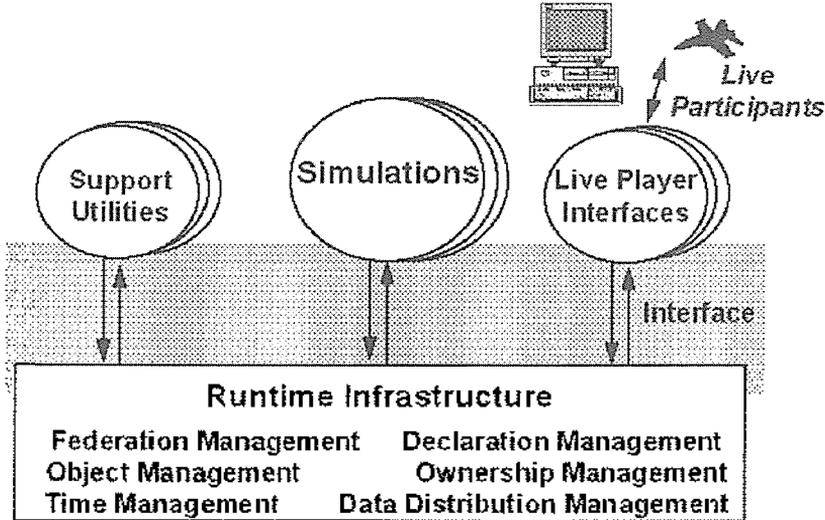
각 federation과 시뮬레이션에 대한 HLA Object Model에 포함될 정보들을 기록하기 위하여 규정된 공통 양식이다.

HLA의 구성 요소들을 보다 면밀히 살펴보기 이전에 HLA이 탄생하게 된 배경을 고급 분산 시뮬레이션의 진화 관점에서 조명한 앞서의 <그림 3>과 달리 HLA에 초점을 맞추어 살펴보자. <그림 7>은 HLA의 개발이 DIS, ALSP 등을 포함한 HLA 이전의 DoD 노력들이 근간이 되어 정부, 학계, 산업체 등의 참여로 수행되었음을 보여준다. 한편, 본 장에서 기술될 HLA의 기술적 내용들에 대한 이해를 돕기 위하여 HLA의 설명에 자주 사용되는 시뮬레이션 관련 주요 전문 용어들의 정의를 표1과 같이 정리하였다.

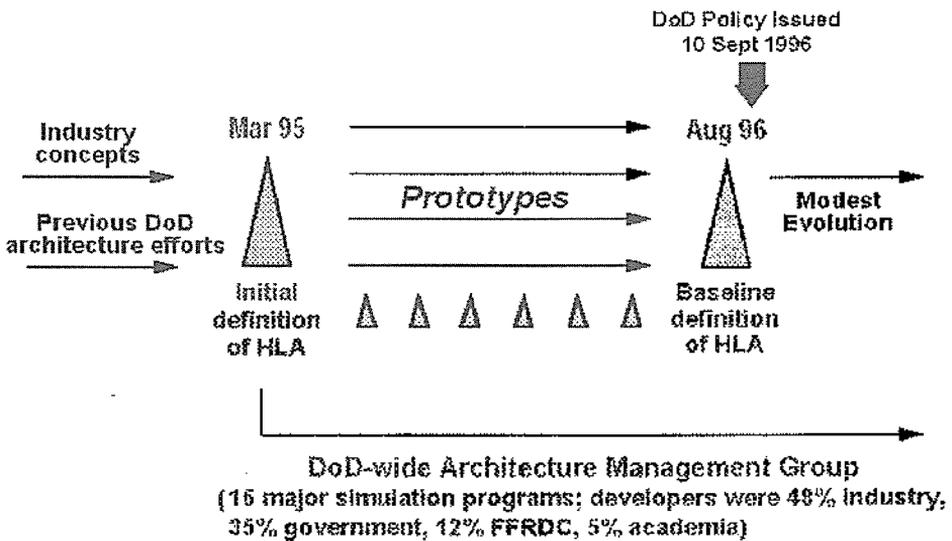
가. HLA Rules

HLA federation 구성 요소들 간의 책무와 관계들은 federates와 관련하여 5개, federation과 연계하여 5개 등 총 10개의 기본 규정들로 정의된다. 다음은 각 규정에 대한 요약이다.

<그림 6> HLA의 구성에 대한 기능적 관점



<그림 7> HLA 개발 과정



Rule 1

Federation은 HLA Object Model Template(OMT)에 준하여 문서화된 하나의 HLA Federation Object Model(FOM)을 갖는다.

〈표 1〉 HLA 관련 주요 전문용어

Architecture	컴퓨터를 기반으로 하는 시뮬레이션 시스템에서 시스템을 구성하는 주요 기능 요소(functional components)와 이들의 설계기준(design rules), 그리고 기능요소 간의 공유영역(interface) 등.
Model	시스템/개체/현상/공정의 물리적/수학적/논리적 표현.
Simulation	시간 축 상에서 모델을 시행하기 위한 하나의 방법으로 시험, 분석, 또는 훈련을 위한 기술. 시뮬레이션에는 실제 세계의 시스템들이 포함되기도 하며, 실제 또는 개념적 세계가 모델에 의하여 표현될 수도 있음.
Real-time	물리적 공정이 발생하는 실제의 시간.
Constrained simulation	시간 진행이 실제의 벽시계와 일정한 관계를 유지하는 시뮬레이션. Real-time simulation과 동일한 의미임.
Federate	HLA federation의 구성원. Federation에 참여하는 모든 applications는 federates라 일컬어짐.
Federation	어느 특정 목표를 성취하기 위한 하나의 집합으로서 상호간에 작용을 하는 federate들과 하나의 공통된 federation object model 그리고 상호 작용을 보좌하는 RTI로 구성됨.
Object	실제 세계를 반영하는 federate에 대한 개념적 표현을 구성하는 하나의 기본 요소로서 federate 간의 상호 작용성이 보장될 수 있는 추상적 개념과 해상도 수준의 단위. 주어진 시간에서 object의 상태는 object의 모든 속성값들로 정의됨.
Object Model	주어진 체계에 내재한 objects들의 사양으로서 objects의 특성(attributes)과 objects 사이에 존재하는 정적 및 동적 관계에 대한 묘사를 포함.
Attribute	object의 상태를 결정짓는 명목적인 부분.
Class	유사한 특성, 공통된 거동, 공통된 관계, 공통된 의의학(semantics) 등을 갖는 objects 집합의 분류.
Event	object attribute의 변화, objects간의 상호 작용, 새로운 object의 생성, 또는 object의 소멸을 정의하며, federation 시간 축 상의 한 점과 연계됨.
Message	federates 간에 송수신되는 데이터의 단위로 최대 하나의 event를 포함. 하나의 message는 전형적으로 하나의 event와 관련한 정보들을 담고 있으며, 이 message는 다른 federates에 event가 발생하였음을 통지하는데 사용됨.
FOM	Federation Object Model: 특정 federation을 묘사하는데 필수적인 모든 공유 정보(objects, attributes, associations, and interactions) 목록.
SOM	Simulation Object Model: 특정 시뮬레이션 내의 objects, attributes, interactions 등을 기술하며, federation 내의 시뮬레이션들 간에 외적으로 사용된다.
OMT	Object Model Template: HLA object model 문서들에 대한 공통 골격을 제시. 이를 통하여 시뮬레이션들과 시뮬레이션 구성 요소들 간의 상호운용성과 재사용성을 촉진.

Rule 2

Federation 내에서 모든 object 표현들은 runtime infrastructure(RTI)가 아닌 federates 내에 존재한다.

Rule 3

Federation 실행 중에 federates간의 모든 FOM 데이터 교환은 RTI를 경유하여 수행된다.

Rule 4

Federation 실행 중에 federates들은 HLA 공유영역 사양(interface specification)에 준하여 RTI와 상호 작용한다.

Rule 5

Federation 실행 중에 어느 순간 한object의 attribute는 단 하나의 federate에 의하여 소유된다.

Rule 6

Federates들은 OMT에 준하여 문서화된 동일한 양식의 SOM을 갖는다.

- 각 시뮬레이션은 federation에 제공될 각자의 기능성을 OMT 양식으로 기술할 수 있어야 한다.
- SOM에 표시되는 objects, attributes, 상호작용 등은 federation에 따라 일부만 사용될 수도 있다.

Rule 7

Federates들은 자체 SOM에 포함된 objects들의 attributes들을 갱신할 수도 있고, 반영할 수도 있다. 또한, SOM에 규정된 대로 외부와 상호작용을 주고 받을 수 있다.

Rule 8

Federates들은 federation의 집행 중에 SOM의 규정에 따라 동적으로 attributes들의 소유권을 주고 받을 수 있다.

Rule 9

Federates들은 SOM에 규정된 데로 objects들의 attributes들을 갱신하는 조건들을 변경할 수 있다.

Rule 10

Federates들은 각자의 지역 시각(local time)들을 관리할 수 있으며, 이 경우 federation 내의 다른 구성원들과 데이터를 교환하는데 문제가 없도록 하여야 한다.

- Federation 내의 시뮬레이션들은 외관상 하나의 시계에 따라 관리되는 것처럼 보여야 한다.
- 시뮬레이션 내의 시계는 자체 관리될 수 있으며, 이 경우 federation 내 시뮬레이션들 간의 상호작용에 문제가 야기되어서는 안 된다.

나. HLA Object Model Template

Object model은 특정 응용에서 실제의 세계를 대표하는 object의 신분증이 되며, 다음을 포함한다:

- object의 속성(attributes)
- object의 정적 관계(class hierarchies, associations, aggregations)
- object의 동적 관계(interactions)
- 개별 object의 거동

HLA object models에는 Federation Object Model(FOM)과 Simulation Object Model(SOM)이 있으며, 개략적인 정의는 <표 1>에 나타나 있다. 이들을 지원하기 위한 Object Model Template(OMT)을 포함하여 보다 상세히 정리하면 다음과 같다.

Federation Object Model (FOM)

특정 federation을 묘사하는데 필수적인 모든 공유 정보(objects, attributes, associations, and interactions)의 목록으로서 기본 정보와 선택 정보로 나누어진다.

- 기본 정보: object classes, object interactions, attributes/parameters, lexicon

- 선택 정보: object associations, composition relationships, object model metadata

Simulation Object model (SOM)

특정 시뮬레이션 내의 objects, attributes, interactions 등을 기술하며, federation내의 시뮬레이션들 간에 외적으로 사용된다. SOM은 각 시뮬레이션 또는 federate의 입출력 데이터를 논리적으로 표현하며, 시뮬레이션 시스템들이 HLA federation들과 호환성이 있는가를 판별하는 도구가 된다.

Object Model Template (OMT)

OMT는 HLA object models들을 묘사하는 표준화된 표현 양식으로서 FOM의 종합적인 개발을 용이하게 하며, federation 구성원 후보들의 자격을 표현하는 통일된 수단을 제공한다. 또한, 공통된 FOM 개발 도구 세트의 설계와 개발을 가능케 한다. OMT는 다음과 같은 요소들로 구성된다:

- Object Class Structure Table
- Object Interaction Table
- Attribute/Parameter Table
- FOM/SOM Lexicon

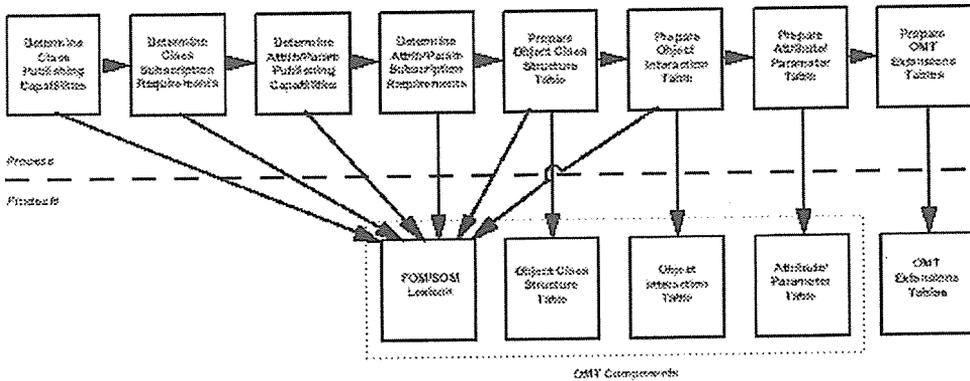
〈표 2〉 Object Class Structure Table의 양식과 예

Object Class Structure Table			
<class> (<ps>)	<class> (<ps>)	<class> (<ps>)	<class> (<ps>) [, <class> (<ps>)]* [<ref>]
		<class> (<ps>)	<class> (<ps>) [, <class> (<ps>)]* [<ref>]
<class> (<ps>)	<class> (<ps>)	<class> (<ps>)	<class> (<ps>) [, <class> (<ps>)]* [<ref>]
		<class> (<ps>)	<class> (<ps>) [, <class> (<ps>)]* [<ref>]
<class> (<ps>)	<class> (<ps>)	<class> (<ps>)	<class> (<ps>) [, <class> (<ps>)]* [<ref>]
		<class> (<ps>)	<class> (<ps>) [, <class> (<ps>)]* [<ref>]
...
Air Vehicle(S)	Fixed Wing (S)	Fighter-Attack (S)	F-14 (PS) F-16 (PS) F-18 (PS)
		Bomber (S)	B-1B (PS) B-2 (PS)
		Rotary Wing (PS)	

〈그림 8〉 SOM 개발 공정

Object/Interaction	Attribute/Parameter	Data Type	Cardinality	Units	Resolution	Accuracy	Accuracy Condition	Update Type	Update Condition	T/A	U/R
<class> <interaction>	<attribute> <parameter>	<datatype>	{<size>}	<units>	<resolution>	<accuracy>	<condition>	<type>	<rate> <condition>	<ta>	<ur>
<class> <interaction>	<attribute> <parameter>	<datatype>	{<size>}	<units>	<resolution>	<accuracy>	<condition>	<type>	<rate> <condition>	<ta>	<ur>
<class> <interaction>	<attribute> <parameter>	<datatype>	{<size>}	<units>	<resolution>	<accuracy>	<condition>	<type>	<rate> <condition>	<ta>	<ur>
<class> <interaction>	<attribute> <parameter>	<datatype>	{<size>}	<units>	<resolution>	<accuracy>	<condition>	<type>	<rate> <condition>	<ta>	<ur>
Task	Area	Final	1	m2	0.1 m2	perfect	always	conditional	scene event	TA	UR
	Velocity	Double	1	m/sec	1 m/sec	0.1 m/sec	none	periodic	10 Hz	TA	UR
	State	Task_Type	1	n/a	n/a	n/a	n/a	conditional	scene event	TA	UR
	Position	Rectang_Type	1	n/a	n/a	n/a	n/a	periodic	10 Hz	TA	UR
Weapon Detonate	Workload	Wh_Type	1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

〈그림 9〉 FOM의 개발 공정

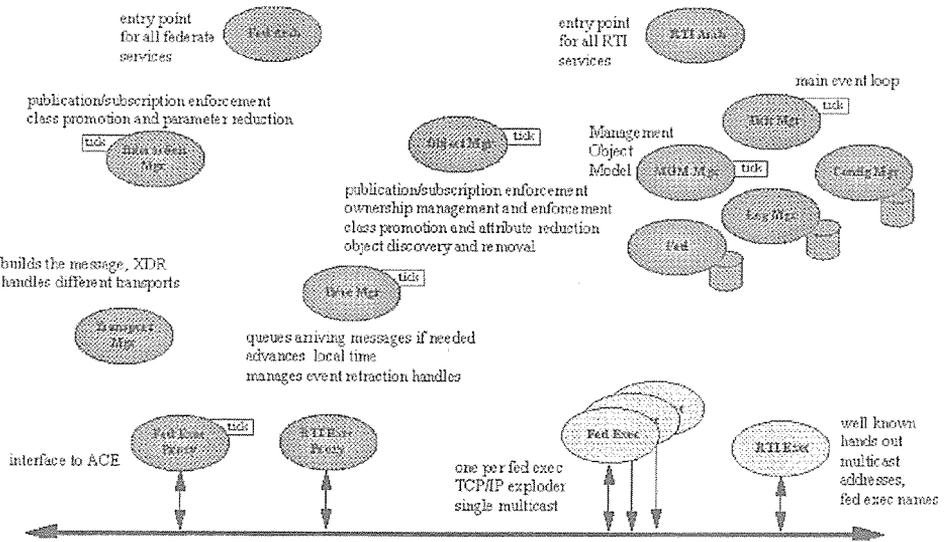


다. Runtime Infrastructure

앞의 〈그림 6〉에서 Runtime Infrastructure(RTI)의 구성과 기능에 대하여 살펴 보았듯이 RTI는 Federation Management, Object Management, Time Management, Declaration Management, Ownership Management, Data Distribution Management 등 6개의 federation 관리 성분으로 구성된다. RTI의 존재는 HLA와 기존 DIS를 구별 짓는 중요한 역할을 하는데, DIS가 진정한 의미에서 분산 시스템에 분산 시뮬레이션이라면, HLA는 분산 시뮬레이션이기는 하지만, 데이터 입출력의 관리를 중앙에서 RTI가 통제하는 중앙 관리식 분산 시뮬레이션이라 할 수 있다. 즉, RTI는 federation에 참여하는 각 federate의

overhead를 최소화함으로써 주어진 시스템들의 성능에서 보다 고성능의, 보다 대응량의, 보다 유연한 통합 시뮬레이션을 가능케 한다. RTI의 구성에 대한 이해를 돕기 위하여 <그림 10>에 RTI와 단일 federate간의 관계를 인용하였으며, 각 구성원의 기능들은 <표 5>에서 <표 10>사이에 표시하였다.

<그림 10> RTI와 단일 federate간의 관계



<표 5> Federation Management

Create Federation Execution	Pause Achieved	Federate Save Begun
Destroy Federation Execution	Request Resume	Federate Save Achieved
Join Federation Execution	Initiate Resume	Request Restore
Resign Federation Execution	Resume Achieved	Initiate Restore
Request Pause	Request Federation Save	Restore Achieved
Initiate Pause	Initiate Federate Save	

〈표 6〉 Object Management

Request Id	Receive Interaction	Change Interaction Order Type
Register Object	Delete Object	Request Attribute Value Update
Update Attribute Values	Remove Object	Provide Attribute Value Update
Discover Object	Change Attribute Transportation Type	Retract
Reflect Attribute Values	Change Attribute Order Type	Reflect Retraction
Send Interaction	Change Interaction Transportation Type	

〈표 7〉 Time Management

Request Federation Time	Request Minimum Next Event Time	Next Event Request Available
Request LBTS	Set Lookahead 6.6 Request Lookahead	Flush Queue Request
Request Federate Time	Time Advance Request Available	Time Advance Grant
Publish Object Class	Subscribe Object Class Attribute	Control Updates
Publish Interaction Class	Subscribe Interaction Class	Control Interactions

〈표 8〉 Declaration Management

Publish Object Class	Subscribe Object Class Attribute	Control Updates
Publish Interaction Class	Subscribe Interaction Class	Control Interactions

〈표 9〉 Ownership Management

Request Attribute Ownership Divestiture	Attribute Ownership Acquisition Notification	Query Attribute Ownership
Request Attribute Ownership Assumption	Request Attribute Ownership Acquisition	Inform Attribute Ownership
Attribute Ownership Divestiture Notification	Request Attribute Ownership Release	Is Attribute Owned by Federate

〈표 10〉 Data Distribution Management

Create Update Region	Associate Update Region	Modify Region
Create Subscription Region	Change Thresholds	Delete Region

〔참고문헌〕

1. Dahmann J., "High Level Architecture - Overview and Rules", <http://www.dmsso.mil/>, Sept. 1997.
2. Dahmann J., "High Level Architecture - Policy and Definition Process", <http://www.dmsso.mil/>, Sept. 1997.
3. Lutz b., "HLA OMT Fundamentals", <http://www.dmsso.mil/>, Sept. 1997.
4. "HLA Glossary", <http://www.dmsso.mil/>, Sept. 1997.
5. Weatherly R., "Runtime Infrastructure F.0 Organization and Function", Simulation Interoperability Workshop, Orlando, Florida, March 1997.
6. "New Directions in Modeling and Simulation", Aerospace America, AIAA, pp. 35-39, Jan. 1997.
7. Bouwens C. and Zabek A., "An Evolution in Advanced Distributed Simulation", Proceedings of 1997 AIAA Modeling and Simulation Conference, pp. 139-146, Aug. 1997.

8. Fullford D., "Transitioning from DIS to the High Level Architecture (HLA)", Proceedings of 1997 AIAA Modeling and Simulation Conference, pp. 147-153, Aug. 1997.