

M&S PlugIn-Based Architecture Framework 개발

(M&S PlugIn-Based Architecture Framework Development)

원 강 연 [†] 최 상 영 ^{**}
 (Garng Yun Won) (Sang Yeong Choi)

요 약 국방 획득업무에 있어 통합적 협력에 의한 M&S의 활용을 지향하는 SBA가 적용 중에 있다. 이를 효율적으로 수행하기 위해서는 공통하부기반을 통해 M&S 구성요소의 재사용성, 재구성, 확장성 등의 제고가 요구된다. PlugIn-Based Architecture(PBA)는 컴포넌트 간 상호독립적인 인터페이스 및 상호작용이 가능함으로써 기존 아키텍처에 비해 컴포넌트의 추가 및 조합이 용이하고 재사용을 높일 수 있는 공통하부기반을 제공할 수 있도록 설계되었다. 또한, 이러한 PBA를 적용한 시뮬레이터 개발을 지원할 수 있도록 PBA Framework을 구현하였다. PBA Framework은 공통하부기반으로서 M&S 업무 적용 시 효율성을 제고시킬 수 있을 것으로 기대된다.

키워드 : 모델링, 시뮬레이션, 아키텍처, 재사용, 컴포넌트, 프레임워크, 플러그인, 시뮬레이터, 모의, 이산, 사건, 미들웨어

Abstract Simulation Based Acquisition(SBA) which pursues to use M&S in manner of integrated collaboration is being applied in defense acquisition. To accomplish SBA efficiently, reusability, reconfiguration and scalability of M&S components are important factors. To avoid constraints caused by coupling of components, PBA is designed to add and configure components easily by enabling independent interface and interaction among the components and provides common development infrastructure also. And PBA framework is implemented to support the development of a simulator which uses the PBA. It is expected that deployment of PBA framework as common development infrastructure can raise efficiency of M&S works.

Key words : modeling, simulation, architecture, reuse, component, framework, plugin, simulator, discrete, event, middleware SBA, HLA, JMASS, RTI

1. 서 론

최근의 무기체계 및 정보체계는 점점 규모가 커지고 복잡해지고 있으며 단일 플랫폼 중심에서 시스템 복합체계(SoS: System of Systems), 더 나아가서는 네트워크 중심의 전장(NCW: Network Centric Warfare) 능

력을 요구하고 있다. 이러한 상황 하에서 기술적 위험과 개발 비용, 기간은 늘어나고 계획범위를 초과하는 등의 어려움이 있지만 획득 비용과 기간 면에서는 더욱 절감을 요구받고 있다.

이러한 무기체계 획득 상의 당면한 요구 및 과제에 대처하기 위하여 무기체계 소요분석, 개발, 시험평가, 훈련 등의 무기체계 순기(life cycle) 상에서 각 단계에 대해 상호 통합적 협력으로 모델링 및 시뮬레이션(M&S)을 활용함으로써 획득 전 과정 상의 시간, 자원, 위험을 실질적으로 감소시키고 체계 품질 및 효과도의 향상을 목표로 하는 Simulation Based Acquisition(SBA)[1]을 추진하고 있다.

획득 프로세스에 있어 협업을 통한 M&S의 효율적 활용을 위해서는 이를 지원해줄 수 있는 하부 공통기반환경(enabler)의 구축이 필요하다. 이러한 기반환경은 M&S 컴포넌트(components)의 재사용성(reusability) 및 상호

[†] 정 회 원 : 국방과학연구소 연구원
wongyn@empal.com

^{**} 비 회 원 : 국방대학교 무기체계학과 교수
sychoi@kndu.ac.kr

논문접수 : 2008년 9월 5일

심사완료 : 2008년 12월 12일

Copyright©2009 한국정보과학회: 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 시스템 및 이론 제36권 제2호(2009.4)

운용성(interoperability)을 보장할 수 있어야 한다.

독립적으로 개발된 시뮬레이터간의 상호 운용성을 보장하기 위해 미국은 High Level Architecture(HLA)[2] 분산 시뮬레이션 아키텍처를 개발하여 미 국방 시뮬레이션 표준으로 제정하였으며 2000년에는 IEEE 1516 표준으로 제정되었다. 또한 미해군의 JMASS Joint Program Officer(JPO)는 1998년 JMASS[3] 아키텍처 프레임워크의 개발을 시작하여 지속적인 성능개선 및 적용을 추진하고 있다. JMASS는 시뮬레이션 엔진 및 관련 서비스, 인터페이스 규칙, 제반 도구 등으로 구성되며 최상위 모델 컴포넌트인 Player와 이들 Player의 구성을 통해 컴포넌트의 재사용을 구현한 아키텍처이다.

HLA는 분산 연합 시뮬레이션을 위한 공통기반 아키텍처를 제공할 뿐 단일 시뮬레이터 자체에 대한 아키텍처를 포함하지 않으며 시뮬레이션 엔진 등의 기반 서비스를 제공하지 않는다. 따라서 시뮬레이터 개발자들은 독자적인 구조로 시뮬레이터를 구현해야 하며 이렇게 구현된 시뮬레이터 구성품은 결과적으로 재사용성이 떨어지게 된다.

JMASS는 시뮬레이션 엔진 및 관련 서비스, 인터페이스 규칙, 제반도구 등으로 구성된다. 모델을 구현하는 최상위 컴포넌트는 Player이며 이들 Player는 Component들로 구성된다. Player간의 통신은 Ports를 생성하고 Ports 간 커플링(coupling)을 통해 이루어진다. 이렇게 Player들이 결합되어 Team이 형성되고 실행이 가능한 모듈이 형성된다.

JMASS는 시뮬레이터 내의 컴포넌트에 대한 재사용성 및 확장성을 제공하지만 재사용 객체인 Players와 Players간의 자료교환을 위한 각각의 Ports와 커플링의 생성은 컴포넌트의 독립성을 결여시키며 아울러 이러한 커플링은 코드 생성 시 이루어지므로 실행 간 시뮬레이션 개체의 동적인 생성, 삭제, 재구성이 불가능하다. 또한, Player에 의해 특정 개체를 구현하는 경우 실행 시 해당 객체에 대한 다수의 인스턴스(instance)를 생성할 수 없는 구조를 가지고 있다[4].

JMASS와 같이 컴포넌트 간 연동을 위해 해당 메시지별로 Port와 커플링을 생성하는 아키텍처에 대한 취약점은 제2장에서 부연 설명하겠다.

본 논문은 이와 같이 기존 M&S 아키텍처가 갖는 제약사항을 고려하여 시뮬레이션 컴포넌트 간의 커플링(coupling)이 없이 상호 독립적인 인터페이스 및 상호작용이 가능함으로써 컴포넌트가 쉽게 추가되고 조합될 수 있으며 시나리오에 따라 plug-and-play 방식으로 재구성이 용이하고 시뮬레이션 실행 시에도 구성품의 생성/삭제(hot-swap)가 가능토록 고안된 PBA[5]의 프레임워크 개발 내용과 이를 적용한 시뮬레이션 구현 사례

를 기술한다.

본 논문의 구성은 제2장에서 PBA의 주요 설계 내용에 대해 설명하고 제3장은 PBA 프레임워크 구현 내용과 이를 통한 시뮬레이션 구현 사례를 들고 제4장에서 결론을 맺는다.

2. PBA

PBA는 그림 1과 같이 사건관리, 스케줄링, 동기화 등을 담당하는 시뮬레이션 엔진과 시뮬레이션 S/W를 구성하는 빌딩블록인 PlugIn으로 이루어지며 S/W 컴포넌트인 이들 PlugIn 간의 커플링(coupling)을 최소화함으로써 상호 독립성을 보장하기 위해 시뮬레이션 엔진을 통해 메시지 교환을 수행하는 Independent Component Architecture[6] 스타일을 갖는다.

그림 2는 구성품 간의 연동을 위한 두 가지 형태의 연결방식을 대비하여 나타내었다. 구성품 간의 연결을 일대일 방식으로 구성하는 경우는 메시지 관리 및 교환 기능을 담당하는 중계자를 통한 방식에 비해 참여하는 노드의 수가 늘어날수록 소요되는 연결수가 상당히 증가되며 송수신하는 메시지별로 포트(port)를 생성하고 이들 간의 연결을 생성해야 하는 등 구성품의 추가, 삭제 등의 구성변경이 어려운 구조이다. 이에 반해, 중계자를 통한 메시지 교환 방식은 참여하는 컴포넌트와 중

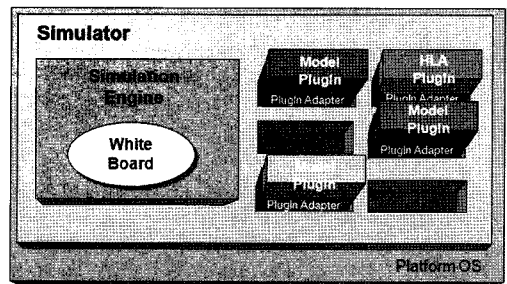


그림 1 PBA 구조

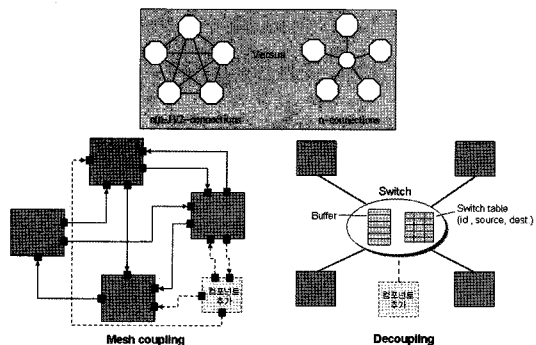


그림 2 연결 구조 비교

게자간의 연결만 필요하고 추가적으로 컴포넌트 간의 메시지 교환 정보를 관리함으로써 각 메시지 교환을 구현할 수 있으며 특정 컴포넌트의 추가/삭제/수정이 발생하는 경우에도 이와 상호작용을 하는 다른 컴포넌트에 대한 추가적인 수정을 최소화 할 수 있는 특징을 갖는다.

PBA는 후자와 같은 구조를 취함으로써 PlugIn이 상호 독립적인 특성을 갖도록 하였고 이로 인해 자체 수정이 용이할 뿐만 아니라 정적 혹은 동적으로 plug-and-play 방식의 시뮬레이션 구성과 독립적인 PlugIn 구현이 가능함으로써 재사용성을 높일 수 있도록 해준다.

2.1 Plugin

PlugIn은 시뮬레이션 개체 혹은, 개체에 대한 구성품 등의 시뮬레이션 구성 요소에 대한 모델을 구현한다. 그림 3에 나타낸 바와 같이 PlugIn은 시뮬레이션 프레임에 접속되어 작동될 수 있도록 공통 기반 인터페이스와 자체 시뮬레이션 엔진을 제공하는 PlugIn Adapter와 모의 대상에 대한 모델(model)부로 구성된다. 이와 같이 인터페이스 규칙과 공통 기반 서비스를 바탕으로 시뮬레이터와 모델이 분리된 구조는 시뮬레이터 개발 부담 없이 모델 개발만으로 시뮬레이션 구현이 가능하며 독립적으로 기 개발된 PlugIn을 사용목적에 따라 모델 충실도(fidelity) 수준이나 사양 일부분을 수정, 보완함으로써 재사용을 용이하게 할 수 있는 특징을 제공한다.

PlugIn의 모델부분은 단위모델들로 구성된다. 모델은 일반적으로 계층적인 구조의 하부모델로 표현되며 모델의 구현 측면에서는 모델 계층의 단말노드에 해당하는 단위모델들과 이들 간의 인터페이스만으로 표현이 가능하다. 이러한 단위모델 간의 인터페이스 역시 1:1 포트와 커플링을 생성하여 사건을 전달하는 구조가 아닌 PlugIn Adapter의 White Pad(WP) 서비스를 기반으로 송수신 선언을 통한 메시지 중계로 구현된다. 따라서 모델 간 커플링이 최소화된 구조이며 아울러 모델 갱신

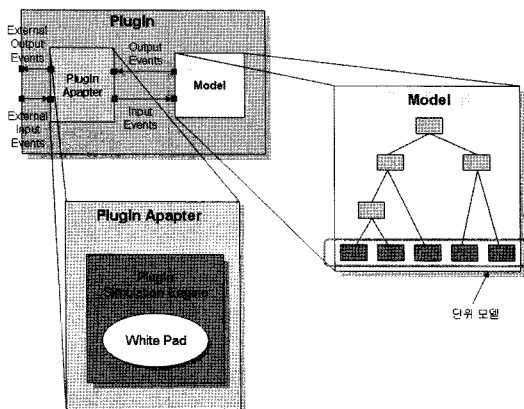


그림 3 PlugIn 구조

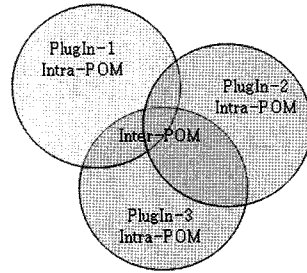


그림 4 PlugIn Object Model

및 수정에 따른 유지보수 노력을 줄일 수 있다.

2.2 PlugIn Object Model(POM)

POM은 PBA 구성요소 간에 교환되는 모든 데이터 종류와 형식을 정의하며 PBA 구성요소 간 사건 전달은 이러한 연동 데이터 모델인 POM에 따라 이루어진다. POM은 PlugIn 간의 정보교환 모델인 Inter-POM과 PlugIn 내부 단위모델 및 구성요소 간의 정보교환 모델인 Intra-POM으로 구성된다. Intra-POM은 PlugIn 내부 연동을 위해 각 PlugIn 마다 독자적으로 갖는 모델이며 Intra-POM의 일부분은 타 PlugIn과의 연동을 위해 공통으로 사용되는 부분이다. 그림 4는 Intra-POM과 Inter-POM의 관계를 나타내었다.

2.3 PlugIn 연동

PlugIn 및 PlugIn 내부의 단위모델 간의 사건전달은 각각 시뮬레이션 엔진과 PlugIn Adapter를 통해 이루어진다. 시뮬레이션 엔진과 PlugIn Adapter는 각각 PlugIn 및 단위모델 정보를 등록하여 관리유지하고 이들 간에 주고받는 사건 종류와 송수신 맵을 구성한 후 store-and-forward 방식으로 수신되어 저장된 사건을 이를 수신하는 PlugIn과 단위모델에 각각 전달한다. 그림 5는 PlugIn이 생성되어 자신을 등록하고 송수신 선언과정을 수행한 후 타 PlugIn과의 사건전달 과정을 나타내었다. 연동 데이터 유형은 Object와 Message로 구분된다. Object는 시뮬레이션 개체의 속성정보를 포함하며 개체가 존재하는 동안의 영속성을 갖는다. Message는 사건정보를 갖는다.

2.4 PlugIn 스케줄링

PBA는 하나 이상의 PlugIn들이 연합하여 시뮬레이션을 수행하는 구조이며 PlugIn 역시 단위모델들이 연합하여 구현된다. 따라서 이들 PlugIn과 단위모델에 대한 작업시기 배정과 시뮬레이션 시간 동기화가 이루어져야 한다. PBA의 스케줄링 및 동기화는 사건의 전달을 통해 이루어진다. 시뮬레이션 엔진의 White Board(WB) 및 PlugIn Adapter의 WP에 배포할 사건이 있는 경우 현재의 시뮬레이션 시각을 Time Stamp로 하여 FIFO (First-In-First-Out) 방식으로 해당 이벤트를 수신하는

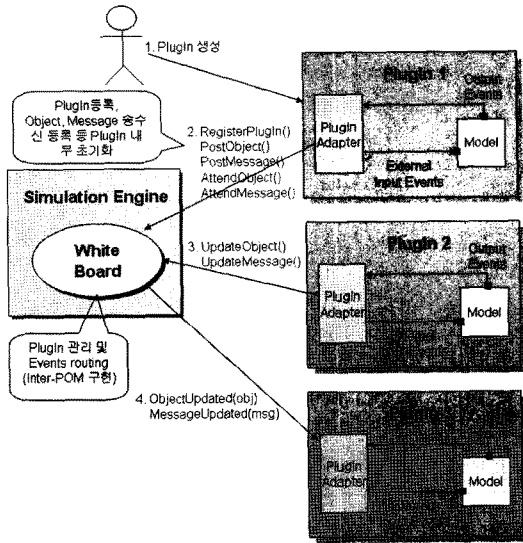


그림 5 Plugin 연동

각 PlugIn 및 단위모델에 저장된 사건들을 동기방식으로 순차적으로 전달(forward)한다. 사건을 수신한 PlugIn 및 단위모델은 현재의 시뮬레이션 시각을 전진시키고 해당사건을 처리한다. WB에 배포할 사건이 없는 경우 시뮬레이션 엔진은 각 PlugIn들의 Expire Timer 중 최소 시간을 갖는 시간으로 시뮬레이션 시각을 전진시킨 후 해당 PlugIn에 대해 Expire 이벤트를 전달함으로써 시뮬레이션 시각 전진 및 작업배정을 수행한다.

2.5 HLA PlugIn

PBA는 단일 노드에 의한 단독형(stand alone) 시뮬레이션뿐만 아니라 HLA를 통한 분산 연합 시뮬레이션에 호응되는 구조를 갖는다. HLA 분산 시뮬레이션 구현은 그림 6과 같이 RTI 가입을 대행하고 Federate Object Model(FOM)[7]에 따라 사건의 송수신 등 RTI와의 인터페이스를 담당하는 HLA PlugIn만을 삽입함으로써 특정 Federate으로서 HLA/RTI 네트워크에 참여할 수 있다. 따라서 독립적으로 개발된 시뮬레이터의 상호운용을 통해 시뮬레이터 수준에서의 재사용성을 높일 수 있다.

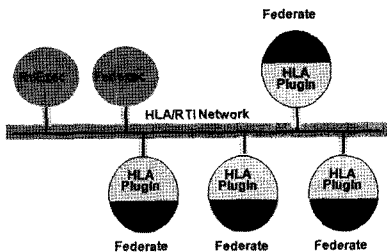


그림 6 HLA 분산 시뮬레이션 구성

3. PBA Framework 개발

PBA를 적용한 시뮬레이터 개발 비용을 줄일 수 있도록 PBA 인터페이스 규칙과 시뮬레이션 엔진, PlugIn 프레임(frame) 등의 하부기반 요소를 제공하는 Framework을 개발하였다. 본 Framework을 활용함으로써 개발자로 하여금 시뮬레이터 자체에 대한 개발 부담을 최소화하고 모델 개발에 집중할 수 있도록 하였다.

PBA Framework의 구성요소는 시뮬레이션 엔진, PlugIn, 연동데이터모델, HLAPlugIn 등으로 이루어지며 Class Library로 제공되도록 개발하였다.

3.1 Simulation Engine

시뮬레이션 엔진은 생성 및 삭제되는 PlugIn 정보와 이들 PlugIn의 이벤트 송수신 정보를 White Board (WB)를 통해 관리하고 PlugIn이 발생시킨 이벤트를 역시 WB에 저장한 후 이를 수신하는 PlugIn들에 순차적으로 전달하는 역할을 수행한다. 이를 통해 시뮬레이션 엔진은 각 PlugIn 간의 연동기능을 제공하며 PlugIn으로 하여금 전달된 이벤트에 포함되는 time stamp로의 시간 동기화를 가능하게 함과 아울러 작업배정이 이루어지도록 한다. WB에 배포할 사건이 없는 경우, 시뮬레이션 엔진은 WB에 저장된 PlugIn의 Expire Timer 값 중 최소 시간을 갖는 PlugIn에 대해 작업배정과 시뮬레이션 진행을 통제한다. 특히, 실시간 제약조건 하에서는 Real Time Clock(RTC)을 바탕으로 실시간 시뮬레이션 스케줄링을 제공한다.

시뮬레이션 엔진의 White Board(WB)는 생성되거나 삭제되는 PlugIn의 관리를 담당하며 Post/Attend 방식에 의해 접속된 PlugIn과 이들 간의 자료 교환 정보에 대한 유지관리와 사건배포를 수행함으로써 사건중심 PlugIn 연동 방식을 제공한다. WB는 그림 7에 나타낸 바와 같이 PlugIn Table, Event Map, Event Queue 등으로 구성된다. PlugIn Table은 Class Name, PlugIn Handle, Timer Expire Time 등의 PlugIn 정보를 관리한다. Event Map은 각 PlugIn에 의한 객체 및 메시지 형태의 이벤트에 대한 배포 및 수신 선언을 바탕으로 각 이벤트의 배포 PlugIn, 수신 PlugIn 정보를 관리한다. Event Queue는 각 PlugIn이 발생한 이벤트를 저장(store)한 후 이를 수신하는 PlugIn들에 순서에 따라 전달(forward)하기 위한 저장소이다.

3.2 PlugIn

PlugIn은 기본 구성요소인 PlugIn Adapter와 Model로 이루어지며 특정 PlugIn 개발을 위한 Base Class로 제공될 수 있도록 구현하였다. PlugIn Adapter는 하부 구성요소인 White Pad에 등록된 송수신 사건 목록을 바탕으로 PlugIn 내부 단위모델간의 지역사건 배포를

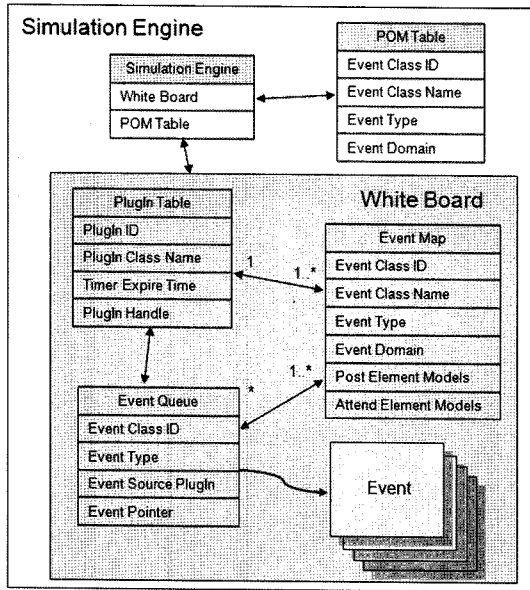


그림 7 Simulation Engine 구조

수행하며 PlugIn 간의 정보교환 모델인 Inter-POM을 바탕으로 전역 시뮬레이션 엔진의 WB에 PlugIn 외부와의 송수신 사건을 등록하고 시뮬레이션 수행 시 PlugIn 외부와의 사건 송수신을 대행한다. 이러한 내/외부 사건의 중계 서비스는 해당 사건을 수신하는 단위모델에 대해 수신 사건의 time stamp로의 시간 전진과 작업배정 기능을 하고 PlugIn 내부 사건의 배포가 완료된 경우 외부 사건과 엔진 자신의 Timer 값을 시뮬레이션 엔진에 갱신시키는 등 PlugIn 자체에 대한 시뮬레이션 엔진 기능을 제공한다.

Model은 단위모델의 집합으로 이루어지며 이러한 각 단위모델 개발을 위한 틀로써 Abstract Class인 Ele-

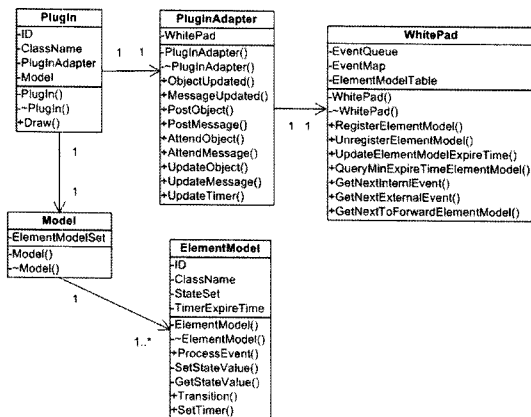


그림 8 PlugIn Class Diagram

ment Model을 구현하였다. PBA는 이산사건 시스템(Discrete Events System)을 월드뷰(World View)로 하므로 Element Model Class는 Time Stamp를 갖는 사건에 대한 액션을 수행하는 관련 함수들로 구성된다.

그림 8은 PlugIn 구성요소에 대한 Class Diagram이다.

3.3 HLA PlugIn

HLA PlugIn은 분산시뮬레이션 표준인 HLA를 통한 다중 노드로 구성된 분산시뮬레이션 구현과 단독 시뮬레이터로 구현된 시스템을 향후 연합 분산시뮬레이션으로의 확장을 용이하게 할 수 있도록 고안하였다. 그림 9에 나타난 바와 같이 HLA PlugIn은 원하는 외부 연동주기에 따라 Run Time Infrastructure(RTI)를 통해 외부와의 송수신 사건의 중계 기능을 구현하였다.

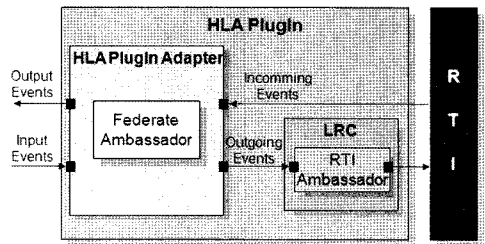


그림 9 HLA PlugIn 구조

HLA PlugIn은 HLA PlugIn Adapter와 Local RTI Component(LRC)로 구성된다. HLA PlugIn Adapter는 Federate 밖으로 배포할 사건을 시뮬레이션 엔진으로부터 수신하여 바로 LRC의 RTI Ambassador에 전달하여 이를 수신하는 다른 Federate들에게 배포하도록 의뢰하는 역할을 한다. HLA PlugIn이 HLA 분산 시뮬레이션 연동주기에 따라 작업이 배정되었을 때 RTI Ambassador에 누적된 사건들은 Federate 외부 즉, RTI로 출력되며 동시에 Federate Ambassador는 외부로부터 수신된 사건들을 callback 루틴에 의해 일괄적으로 WB에 전달하는 역할을 한다.

3.4 PBA Framework을 통한 시뮬레이션 구현

PlugIn의 재사용성, 재구성, 확장성 등을 확인하기 위하여 PBA Framework를 이용하여 단일노드와 3개의 노드로 구성된 유도탄 교전 효과도 분석용 유도탄 교전 시뮬레이션을 각각 구현하였고 이를 확인하였다.

단일노드 시뮬레이션은 하나의 PC에 시뮬레이션 상황진시, 플랫폼, 트랙, 유도탄 등의 PlugIn을 구성하여 구현하였다.

표적모의와 모의수행상황 전사용 노드, 표적탐지/발사 통제를 수행하는 플랫폼과 유도탄 모의를 담당하는 노드 각 2개로 구성된 분산 시뮬레이션은 단일노드용으로 개발된 PlugIn 들을 재사용 및 재구성하여 구현하였

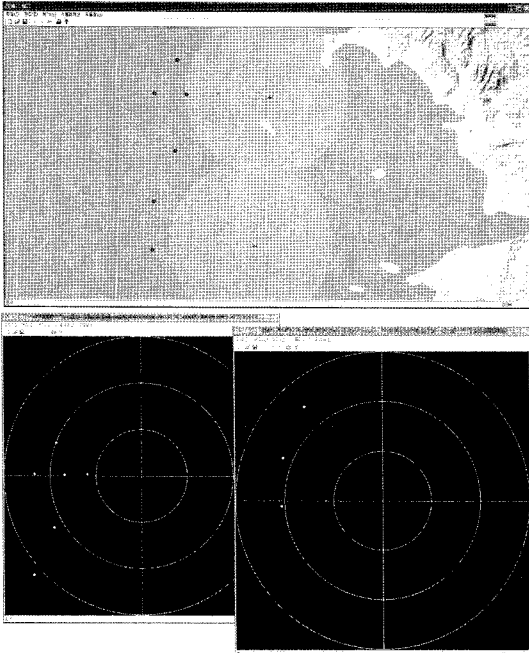
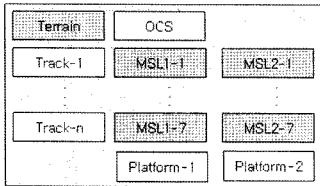


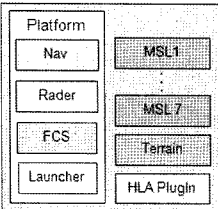
그림 10 유도탄 교전 모의 화면

상황전시 통제 / 표적모의 / Platform1,2 / 유도탄 모의

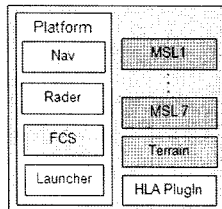


1. 단독 시뮬레이션

Platform / 유도탄 모의기 1

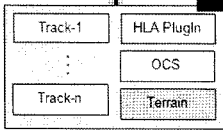


Platform / 유도탄 모의기 2



HLA/RTI Network

2. 분산 시뮬레이션



상황전시 통제 / 표적 모의기

그림 11 시뮬레이션 구성도

으며 이들 노드 간 연동은 HLA PlugIn을 추가하여 HLA 표준을 따르도록 구현하였다. 그림 10은 시뮬레이션 수행 화면이다.

표적모의, 시뮬레이션 통제, 모의상황전시 등은 각 표적에 대해 Track PlugIn을 생성하여 구현하였고 시뮬레이션 통제 및 모의상황전시는 OCS PlugIn을 통해 구현하였다. Platform과 각 탑재 유도탄 및 유도탄 교전 모의는 Platform PlugIn과 MSL PlugIn을 통해 구현하였다. Platform은 Nav, Radar, FCS, Launcher 등의 PlugIn 집합으로 구성된다. 그림 11은 시뮬레이션 노드 및 노드별 PlugIn 구성을 나타내었다.

4. 결론

PBA는 아키텍처 기반요소와 plug-and-play 방식으로 사용될 수 있는 소프트웨어 컴포넌트인 PlugIn을 통하여 재사용성, 재구성(reconfiguration), 확장성(scalability) 등을 높일 수 있도록 고안한 M&S 소프트웨어 아키텍처이다. 개발자는 PBA를 통해 아키텍처 구성요소 간 공통 연동규칙, 인터페이스를 준수함으로써 소프트웨어를 구성하는 빌딩블록인 PlugIn을 재사용하고 조합하여 시뮬레이션 시스템을 개발할 수 있다. 또한, 이들 PlugIn 간의 커플링(coupling)을 최소화하고 상호 독립성을 보장하는 아키텍처 스타일을 가짐으로써 동적인 PlugIn의 생성 및 삭제가 용이하며 실험 목적에 따라 PlugIn의 조합을 통해 시뮬레이션의 재구성 및 확장이 용이한 특성을 제공한다.

그림 12는 컴포넌트 기반 개발 및 아키텍처 기반 개발 등을 근간으로 하는 PBA위상을 나타내었다.

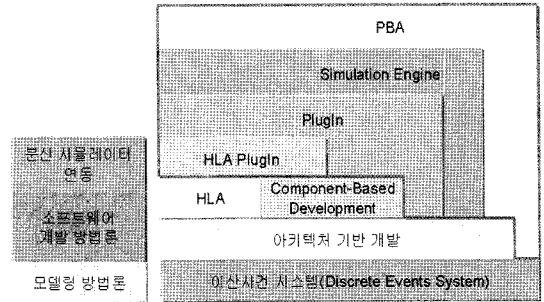


그림 12 PBA의 위상

PBA Framework은 시뮬레이션 엔진, PlugIn 프레임, HLA PlugIn 등으로 구성되며 PBA 기반 개발 프레임워크를 제공한다. 개발자는 PBA Framework으로 제공되는 공통 기반 서비스를 활용함으로써 PBA의 적용과 시뮬레이터 대한 개발 부담을 최소화할 수 있을 것이다.

PBA Framework은 시뮬레이터 구성품뿐만 아니라 시뮬레이터 단위의 재사용성을 높일 수 있는 아키텍처 개발 프레임워크로서 M&S 업무 적용 시 효율을 증대시킬 수 있을 것으로 판단한다.

참 고 문 헌

- [1] Simulation Based Acquisition Industry Steering Group(SBA ISG), "SBA Functional Description - Version 1.1," 1999.2.
- [2] IEEE Std 1516-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules.
- [3] LCDR Doug Buchy, "Joint Modeling and Simulation System," Joint Modeling and Simulation System Program Office, 2000.1.
- [4] Robert J. Meyer, "Joint Modeling and Simulation System(JMASS)-What it does,... and What it doesn't!"
- [5] 원강연, 최상영, "SBA를 위한 M&S PlugIn-Based Architecture(PBA) 구조에 대한 연구", 한국군사과학기술학회지, 2007.3.
- [6] Len Bass and Paul Clements and Rick Kazman, "Software Architecture in Practice," Addison-Wesley, 1998, pp. 101-102.
- [7] IEEE Std 1516.1-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification.



원 강 연

1993년 숭실대학교 전산과 졸업. 2007년 국방대학교 대학원 무기체계과 졸업(석사). 1993년~현재 국방과학연구소 근무 관심분야는 모델링 및 시뮬레이션, 소프트웨어 아키텍처, 미들웨어, 실시간 시스템 등



최 상 영

1982년 육군사관학교 졸업(학사). 1985년 국방대학교 대학원 국방과학 졸업(석사) 1989년 University of Cranfield(영) 대학원 졸업(박사). 2003년~현재 국방대학교 교수. 관심분야는 모델링 및 시뮬레이션, 소프트웨어 엔지니어링, SoS 엔지니어링, 체계 아키텍처 등