□ 研 究 論 文 □

情韓國國防經營分析學會誌
第35卷 第1號, 2009. 4. 30.

# HLA Interface Specification 1.3를 이용한 OO기반의 페더레이트 모델링 및 구현

## (Object-Oriented Federate Modeling and Implementation using HLA Interface Specification 1.3)

최 웅 철(Woong-Chul Choi)*, †유 기 훈(Ki-Hun Yu)**

초 록

HLA는 이기종 시뮬레이션간의 연동을 위한 IEEE 1516 표준이며, RTI는 연동을 위한 하부 인프라 서비스를 제공하는 기술이다. 본 논문에서는 HLA/RTI에 호환 가능한 OO기반의 페더레이트 모델 구조를 제안함으로써 코드의 재사용성을 높인다. 이는 시스템 개발 프로세스의 효율을 높이고, 개발시간의 단축 및 소프트웨어 개발 비용의 절감효과를 가져 온다. 제안된 페더레이트 모델은 이를 구현한 페더레이트에 대한 HLA 인증으로 그 실질적인 효과를 검증 받는다.

ABSTRACT

HLA is the IEEE 1516 standard for the interoperation among heterogeneous simulations and RTI is a technology which provides a lower infrastructure service to interoperation. In this paper, we propose a Object-Oriented federate model architecture to enhance code reusability. It improves an efficiency of the system development process, and results in development time reduction and cost saving. It also is verified its practical effect through acting HLA certification on an software developed proposed architecture.

**Keywords : HLA/RTI, Object-Oriented Model Architecture, Federate**

**논문접수일 : 2009년 1월 13일 논문게재확정일 : 2009년 4월 22일**

* 광운대학교 컴퓨터과학과 교수

** 광운대학교 컴퓨터과학과 석사과정

† 교신저자

## 1. 서 론

HLA(High Level Architecture)는 IEEE 1516 표준으로 이기종 시뮬레이션 모델간의 상호 운용을 위한 아키텍처이다. RTI(Run-Time Infrastructure)는 이러한 아키텍처를 기반으로 구현된 라이브러리 형태의 소프트웨어로, 실제 시뮬레이션, 가상 시뮬레이션, 구성 시뮬레이션, 무기체계 및 C4ISR (Command Control Communication Computer and Intelligence, Surveillance Reconnaissance)간의 상호 연동을 위한 국방 연동 체계이다. 이러한 RTI는 국방 연구의 M&S(Modeling&Simulation) 뿐만 아니라, 네트워크 가상환경의 각종 애플리케이션이 상호 연동되어 실행될 수 있는 민수 분야의 하부 인프라 서비스를 제공하는 기술이다 [1][2][3][4].

이러한 HLA/RTI 기술은 미래 국방 모델링 및 시뮬레이션 환경 구축을 위한 핵심기술이며, 현재 혹은 향후에 개발될 각 군의 독자 모델들을 상호 운용하는데 활용되는 연동 기반 기술이다. 국외에서는 HLA/RTI에 관한 연구가 최근에도 활발하게 이루어지고 있으며, 다양한 모델링 구조, 유지보수, 컴플라이언스 평가 등 구현뿐만 아니라 유지보수 및 평가 도구에 있어서까지도 많은 연구들이 행해지고 있다[5][6][7]. 하지만 현재 국내에서는 RTI 관련 기술 연구가 개념정립 단계에 있으며, 실제 HLA/RTI 인증에 있어서는 미 국방성에서 공개하는 무료 RTI를 사용하고 있는 상황이다. 이러한 상황 하에서, 국내 HLA/RTI 관련 기술 보유 및 개발의 필요성이 요구되고 있으며, HLA/RTI 기술의 연구는 국방 M&S 분야를 넘어 폭넓은 활용 가능성을 가지고 있다. HLA/RTI는 원격 가상 교육, ITS(Intelligent Traffic System)등, 고부가 소프트웨어의 가능성을 가지고 있으며, 이 외에도 각종 애플리케이션의 상호 연동과 관련되어 많은 관련 연구가 진행되고 있다[8][9][10].

HLA는 구현물인 RTI에 대해 특정한 프로그래밍 언어나 구조를 명시하고 있지 않으며, 이에 따라 현재에는 많은 종류의 RTI가 배포되어 있다. 그러나 대부분의 RTI는 그 설계나 구현기법에 대해서 공개되지 않고 있어, 해당 RTI를 활용하거나 RTI를 개발하는데 어려움을 주고 있다. 이에 본 연구에서는 HLA/RTI를 활용하거나 개발하는데 도움을 줄 수 있는 가이드라인을 제시하고자 HLA/RTI에 호환 가능한 OO(Object-Oriented)기반의 페더레이트 모델 구조를 제안한다. 또한 제안하는 모델 구조를 기반으로 구현된 응용 프로그램은 현재 미 DMSO(Defense Modeling & Simulation Office)의 HLA의 인증 절차 중에 있으며, HLA/RTI에 호환가능한 페더레이트인 응용 프로그램은 HLA Interface Specification 1.3를 활용하여 개발되었다.

본 논문의 구성은 다음과 같다. 2장에서는 기본적인 HLA/RTI의 개념과 HLA 인증 시험 그리고 코드의 재사용성을 향상시키는 OO구조에 대하여 살펴보며, 3장에서는 HLA/RTI에 호환 가능한 페더레이트를 개발하는데 있어서 도움을 줄 수 있는 고려사항, 구조, 구현의 순서로 설명한다. 그 후 마지막으로 4장에서 결론짓는다.

본 논문에서는 객체지향개념에서의 객체(Object)라는 용어와 페더레이트 구현 시, 사용되는 객체(OBject)라는 중의적인 용어의 혼동을 피하기 위하여 전자는 객체, 후자는 오브젝트라는 용어로 서술한다.

## 2. 관련연구

### 2.1 HLA/RTI 개념

2000년도에 IEEE 1516 표준으로 채택 된, HLA는 미 국방성에서 제안된 이기종 시뮬레이터 간의 연동을 위한 상위레벨 개념이다. HLA는 HLA 프레임워크와 규칙(HLA Framework and Rules), 페더레이트 인터페이스 명세서(Federate

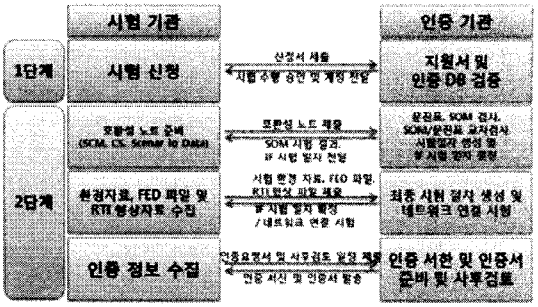
Interface Specification) 그리고 객체 모델 템플릿 (Object Model Template-OMT)의 세 가지로 정의된다.

먼저, HLA 프레임워크와 규칙은 페더레이션 (Federation)에 포함되는 구성 요소들의 역할과 상호 관계에 관한 기본적인 5개의 규칙들과 페더레이트에 포함되는 5개의 규칙, 총 10개의 규칙을 가지고 있다. 다음의 페더레이트 인터페이스 명세는 각 페더레이트와 RTI 간의 기능적 인터페이스에 관한 규약으로 6가지의 서비스 관리 영역으로 나누어 기술하고 있으며, RTI는 페더레이트 인터페이스 명세를 시스템 기종 및 프로그램 언어별로 라이브러리 형태로 구현한 것이다. 마지막으로 OMT는 페더레이션을 구성하는 페더레이트들 간에 이루어지는 공통 데이터 영역을 구조적, 기능적으로 서술하는데 사용된다. OMT는 페더레이션을 구성하는 페더레이트들 사이의 공유 데이터 교환 구조를 서술하는 FOM(Federation Object Model)과 시뮬레이션 시스템이 페더레이션에 제공하는 기능을 표현하는 SOM(Simulation Object Model), 그리고 MOM(Management Object Model)으로 구성된다.

## 2.2 HLA 인증시험

HLA 인증 시험(Certification Test)은 1997년 11월 최초 개시 이후 세계 각국의 개발 페더레이트에 대한 인증 활동을 수행하며 운영되어 왔다. 인증 시험은 구현 모델이 특정 표준과 일치하도록 수행됨을 인증하는 과정으로, HLA 인증시험 체계는 유용성, 이해의 용이성 및 접근성 등 세 가지 개념에 근거하여 개발되었다. 이러한 세 가지 개념을 달성하기 위하여, 시험체계는 세계 어느 곳에서도 접속 및 신청, 시험이 가능하도록 웹을 기반으로 설계되었다.

HLA 인증시험 체계의 유용성과 이해성을 촉진하기 위해 <그림 1>과 같이 2단계의 절차로 구성




〈그림 1〉 HLA 인증시험 절차

되어 있으며, 시험 내용으로는 호환성 문진표 시험(Conformance Statement Dependency Test), SOM 호환시험, SOM/CS 교차 시험 및 인터페이스 시험 등으로 이루어져 있다.

## 2.3 OO(Object-Oriented)구조

OO라는 용어는 객체-지향 프로그래밍 언어를 이용한 소프트웨어 개발 접근법을 표기하는데 사용되었으며, 오늘날 객체-지향 패러다임은 소프트웨어 공학의 전 분야에 사용된다. 단순한 OOP (Object-Oriented Programming)만으로는 최선의 결과를 만들어 내지 못하며, 객체-지향 요구 사항 분석 (OORA:Object Oriented Requirments Analysis). 객체-지향 설계(OOD:Object Oriented design), 객체-지향 도메인 분석(OODA:Object Oriented Domain analysis)의 사항이 모두 고려되어야 한다. 이에 본 논문에서는 객체지향은 페더레이트를 구현하는데 있어서 위와 같은 조건을 만족하는 OO구조를 제안하였다. 이러한 OO구조는 재사용성을 강조하고 있으며, 기본 개념으로서 캡슐화, 상속, 추상, 다형성의 특성을 가지고 있다. 이러한 특성은 아래와 같은 이점을 제공해 준다[11].

* 데이터와 절차의 세부적인 내부 구현 사항은 외부로부터 감추어져 있다. 이는 구조 변경 수행 시, 부작용의 전파를 감소시켜 준다.
* 데이터의 구조와 이를 조작하는 동작들은 하나의 개체인 클래스에 통합된다. 이는 컴포넌

트의 재사용을 용이하게 해준다.

* 캡슐화된 객체들간의 인터페이스는 단순해진다. 메시지를 보내는 객체는 수신하는 객체의 내부구조를 알 필요가 없어, 인터페이스는 단순해지며 시스템의 결합도가 낮아진다.

## 3. HLA/RTI에 호환 가능한 페더레이트 구현

페더레이트를 구현하는데 있어서 모듈간 데이터를 주고받는 오브젝트(Object)와 인터랙션(Interaction)를 어떻게 구성하여 관리할 것인지가 매우 중요하다. 또한 페더레이트의 모델은 같으나 내부 데이터와 동작방식이 다른 경우, 내부의 데이터에 상관없이, 외부적 동작 방식만을 독립적으로 구현하면 되는 구조를 제공함으로써 모듈 및 코드의 재사용성을 높이는 방안이 필요하다. 이에 대해 본 논문에서는 OO 구조를 이용하여 모듈 및 코드의 재사용성을 높일 수 있는 모델링과 구조에 대해 제안한다.

### 3.1 구현 시 고려사항

오브젝트는 속성(Attribute)을 가지고 있으며 페더레이트가 실행되는 메모리에 계속 로드된 상태로 되어 있어야 한다. 그리고 다수개의 속성이 있는 경우 부분적으로 값의 수정이 가능하다. 예를 들어, 미사일 오브젝트가 있다고 한다면, 미사일 종류, 미사일 위치 등이 속성이 될 수 있다. 만약 다수개의 페더레이트가 존재하고, 하나의 페더레이트에서 소유하고 있는 오브젝트를 다른 페더레이트에서 등록해서 사용하는 경우, 두 페더레이트는 같은 오브젝트를 공유하게 된다. 여기서 오브젝트의 주체가 되는 페더레이트와 이것을 사용하는 다른 페더레이트는 오브젝트 생성과 해제의 시점이 다르다. 오브젝트를 직접적으로 관리하는 페더레이트는 시나리오에 따라 생성과 해제가 정해지지만, 다른 페더레이트는 등록된 오브젝트를 제거하지 않는 이상, 오브젝트가 발견(Discovery) 통보를 받으면 생성하고 해제(Remove) 통보를 받으면 해제해야 한다. 즉, 페더레이트 간 오브젝트의 차별된 관리를 고려해야 한다. 또한 오브젝트의 속성들의 값은 다른 페더레이트에 의해 개별적으로 수정이 될 수 있다. 이렇게 수정된 오브젝트 속성 값에 대한 변경이 발생되면, 이러한 속성들의 값을 통보받게 되는데, 하나의 통로를 통해 호출되기 때문에, 어떤 오브젝트의 속성 값인지를 검색하는 기능이 필요함으로 해시 테이블 및 맵 등으로 오브젝트 및 속성들에 대한 관리가 필요하다.

반면, 인터랙션은 파라미터(Parameter)를 가지고 있으며, 필요에 따라 사용하기 위해 메모리에 적재되고 사용이 끝나면 해제된다. 파라미터는 오브젝트의 속성과 다르게 전체를 부분적으로 값의 변경이 불가능하다. 이렇게 인터랙션은 필요에 따라 사용되기 때문에 성능을 고려한 관리가 필요하다. 예를 들어 미사일 충돌이 발생 했을 때, 이에 따른 피해 결과 통보와 같이 임시적으로 사용될 수 있다. 이러한 일이 빈번히 발생할 경우 메모리의 생성과 해제가 자주 발생되어 처리 속도가 느려질 수 있는 문제점을 가지고 있다. 또한 속성과 비슷하게 인터랙션이 발생되면 해당 페더레이트에 통보가 됨으로 이 또한 검색을 위해 해시 테이블 및 맵과 같은 자료구조로 관리를 할 필요성이 있다.

마지막으로 RTI에서는 빅 엔디안과 리틀 엔디안에 대한 구체적인 언급이 없기 때문에, Integer형 등을 속성 및 파라미터로 사용하기 위해서는 전체 페더레이트가 모두 하나의 빅 엔디안과 리틀 엔디안 중 하나를 선택해 사용해야 한다. 하지만 페더레이트의 수가 늘어나고, 개발하는 곳이 다를 경우 이러한 관리가 어려울 수 있기 때문에 이러한 모든 값들을 문자열로 사용하는 것도 하나의 방안이라고 할 수 있겠다.

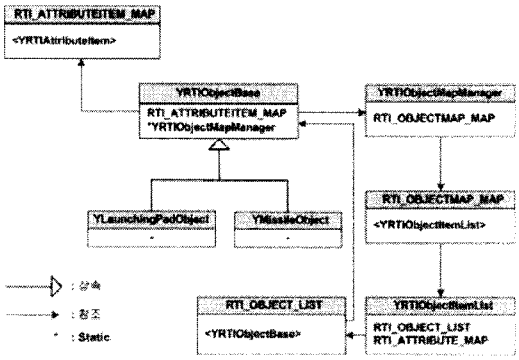
## 3.2 구조(Architecture)

본 논문에서는 오브젝트 클래스, 인터랙션 클래스, 페더레이트 클래스에 대한 구조를 제안하였다. 오브젝트와 인터랙션 클래스는 실제 구현 시나리오에 따라 구현한 클래스도 포함된다.

### 3.2.1 오브젝트 클래스 다이어그램

오브젝트 클래스의 중요한 부분은 속성 값의 관리이다. 페더레이트가 하나인 경우에는 생성과 해제를 오브젝트를 사용하는 페더레이트에서 하기 때문에 관리하는데 어려움이 없다. 하지만 다른 페더레이트와 연동되어 속성 값이 변경되면, 이 값을 동기화하기 위해 변경된 내용을 통보를 해주기 때문에 속성 값을 관리하는데 어려움이 발생하게 된다. 이것은 다른 페더레이트에서 사용되고 있는 오브젝트를 등록하여 사용할 경우, 필요에 따라 해당 페더레이트에 요청하여 속성 값을 가져오는 방식이 아니라, 똑같은 오브젝트와 이에 따른 속성 내용을 메모리에 적재하여 사용하는 RTI 특성 때문에 발생한다. <그림 2>는 본 논문에서 제안한 중요한 값에 대한 오브젝트 클래스 다이어그램을 보여준다. 주요 클래스의 내용은 아래와 같다.

- YRTIObjectBase : 필요한 오브젝트가 있을 때 이 클래스를 상속받아 사용한다. 여기에 해당 오브젝트에 대한 속성 정보와 전체 오브젝트를 가지고 있는 YRTIObjectMapManager를 정적 할당으로 선언한다.
- YRTIAttributeItem : 하나의 속성에 대한 값과 업데이트 되었는지를 알려주는 플래그를 가지고 있다. 값이 변경 되면 해당 플래그를 TRUE로 설정하면, 스케줄링에 의해 다른 페더레이트에게 업데이트 되었다고 통보해준다. 그리고 다시 플래그를 FALSE로 설정하여 다시 통보되는 것을 막는다.
- RTI_ATTRIBUTEITEM_MAP : 속성에 대한 값을 가지고 있는 YRTIAttributeItem를 속성 핸들을 키로 하여 맵 자료구조로 관리한다.
- YRTIObjectMapManager : 페더레이트가 보유하고 있는 모든 오브젝트에 대한 정보를 관리한다. 이 클래스에서 오브젝트 이름 및 핸들로 오브젝트를 검색할 수 있도록 하였다. 또한 인터랙션이 발생되면 영향을 받는 오브젝트의 속성이 처리 될 수 있도록 지원한다.
- RTI_OBJECT_LIST : YRTIObjectBase를 리스트 자료 구조로 관리한다.
- RTI_ATTRIBUTE_MAP : 속성 이름을 키로 하여 속성 핸들을 맵 자료구조로 관리한다. 즉, 이곳에서 속성 이름을 입력하여 핸들을 얻어내고, 그 핸들을 이용하여 RTI_ATTRIBUTEITEM_ MAP에서 실제 속성 데이터를 얻는다.
- YRTIObjectItemList : 오브젝트와 속성에 대한 정보를 리스트 자료 구조로 관리한다.
- RTI_ATTRIBUTEITEM_MAP : 속성에 대한 정보를 가지고 있는 YRTIAttributeItem를 속성 핸들을 키로 하여 맵 자료구조로 관리한다.
- RTI_OBJECTMAP_MAP : 오브젝트 이름을 키로 하여 YRTIObjectItemList를 맵 자료구조로 관리한다.




〈그림 2〉 오브젝트 클래스 다이어그램

여기서 속성 같은 경우 오브젝트에 따라 속성 데이터를 가지지 않고 RTI_ATTRIBUTE_MAP에서 RTI_ATTRIBUTEITEM_MAP를 이용하여

속성 데이터에 접근하는 구조로 되어 있는 걸 확인할 수 있다. 그 이유는 오브젝트에 속하는 속성 값은 오브젝트에 따라 상속되거나 일부만 사용 될 수도 있기 때문이다. 이렇게 함으로써 중복된 속성을 가진 오브젝트가 많아질 경우 많은 메모리의 생성과 해제를 줄여 처리 속도를 높이고 보다 적은 메모리를 사용할 수 있도록 지원한다.

### 3.2.2 인터랙션 클래스 다이어그램

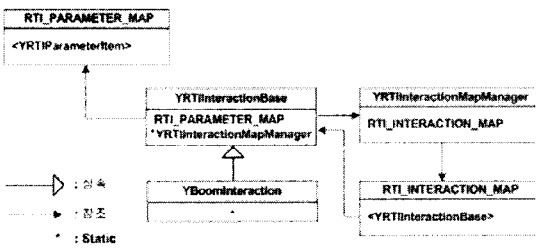
인터랙션의 파라미터는 오브젝트의 속성과는 다르게, 부분적으로 존재 할 수 없기 때문에 구조적으로 보다 단순하다. 하지만 인터랙션 또한 파라미터에 대한 정보를 미리 메모리에 등록해 놓지 않으면 처리할 수가 없으므로 이 또한 관리가 필요하다. <그림 3>은 인터랙션에 대한 클래스 다이어그램을 보여준다. 주요 클래스에 대한 내용은 다음과 같다.

- YRTIInteractionBase : 필요한 인터랙션을 만들기 위해 상속하는 클래스이다. 인터랙션 및 파라미터에 대한 데이터를 가지고 있다. 오브젝트 클래스와 유사하게 YRTIInteractionMap Manager 클래스를 정적 할당으로 선언한다.
- YRTIParameterItem : 하나의 파라미터에 대한 핸들과 값을 가지고 있다.
- RTI_PARAMETER_MAP : 파라미터에 대한 데이터를 가지고 있는 YRTIParameterItem를 파라미터 이름을 키로 설정하여 관리한다.
- RTI_INTERACTION_MAP : 인터랙션 이름을 키로 하여 YRTIInteractionBase를 관리한다.
- YRTIInteractionMapManager : YRTI Interaction Base에 정적으로 할당되어 있으며, 페더레이트에서 사용하는 모든 인터랙션에 대한 데이터를 관리한다. 인터랙션은 임시로 생성되고 해제되기 때문에 잦은 메모리 생성과 해제로 인한 성능 저하가 발생한다. 그렇기 때문에 하나의 인터랙션이 발생하면 그 때 생성된 이후에, 계속 인터랙션이 사용되면 참조 카운트를 증가시켜 값을 가지고 있다가 해당 인터랙션이 모두 사용하지 않을 때, 즉 참조 카운트가 0이 될 때, 해제를 함으로써 성능 저하를 최소한으로 할 수 있도록 지원한다.

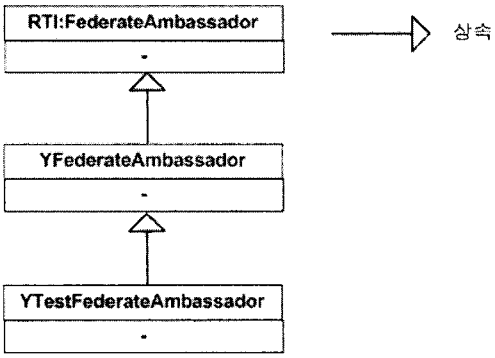
### 3.2.3 페더레이트 클래스 다이어그램

페더레이트 클래스에는 RTI 기본 설정 정보와 통보를 받을 수 있는 인터페이스를 제공하는 것뿐만 아니라, Time Management 관리를 통한 시간에 따른 오브젝트 업데이트 시점, 인터랙션 발생, 시간의 진행 등 전반적인 기능을 포함한다. 페더레이트 클래스 다이어그램은 <그림 4>에서 보여준다. 그리고 해당 클래스에 대한 내용은 다음과 같다.

- RTI-FederateAmbassador : RTI 라이브러리에서 제공하는 클래스로써 해당 페더레이트의 모든 이벤트의 인터페이스를 제공한다.
- YFederateAmbassador : RTI-FederateAm




〈그림 3〉 인터랙션 클래스 다이어그램




〈그림 4〉 페더레이트 클래스 다이어그램

bassador 클래스의 인터페이스를 상속받아 이벤트에 대한 모든 통보를 받을 수 있다. 또한 RTI에 페더레이트 생성, 참가, 탈퇴, 해제 등의 기능부터 Time Management에 관련된 기능을 제공한다. 오브젝트 및 인터랙션 등록 시점을 알려주는 기능도 지원한다.

## 3.3 구현

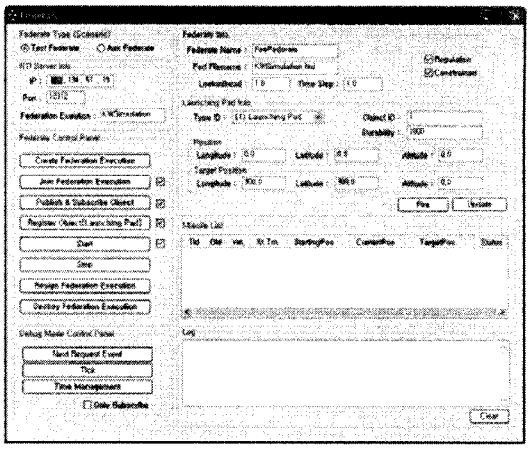
앞서 언급한 클래스를 기반으로 오브젝트, 인터랙션, 페더레이트 클래스를 상속받아 구현하였다. 개발 툴킷은 Visual C++ 6.0를 사용하였으며, HLA Interface Specification 1.3 및 MFC를 이용하여 페더레이트를 구현하였다.

| | Sequence | Contents | Federate Operator's Action | | |
|---|---|---|---|---|---|
| | | | Test Federate | Aux. Federate | FCTT |
| Ready | 1 | Execute RTI | Waiting | Waiting | Waiting |
| | 2 | Execute Federate | Firstly | Secondly | Lastly |
| | 3 | Create Federation | Waiting | Waiting | Check |
| | 4 | Join Federation | Waiting | Waiting | Check |
| | 5 | Publish and Subscribe object | Waiting | Waiting | Check |
| | 6 | Initialize object state | Waiting | Waiting | Check |
| | 7 | Display the object in GUI | Check | Check | Check |
| Execution | 8 | Move the object | Move | Check | Check |
| | 9 | Detect the object | Detect | Detect | Check |
| | 10 | Bomb the object | Bomb | Bomb | Check |
| | 11 | Check the object state | Check | Check | Check |
| | 12 | Resign Federation | Waiting | Waiting | Check |
| End | 13 | Finish RTI | Waiting | Waiting | Check |

〈그림 5〉 인증 시나리오 단계

### 3.3.1 구현 시나리오

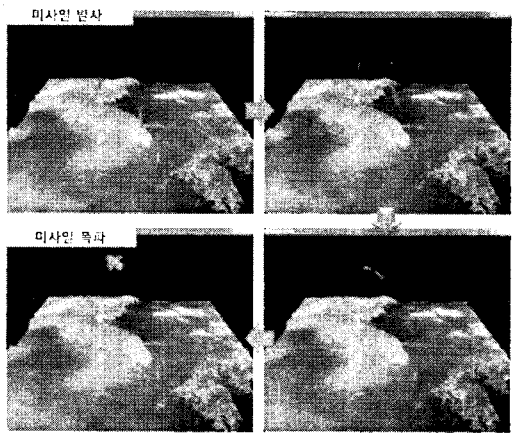
두 개의 발사대 오브젝트(YLaunchingPadObject)를 적당한 위치에 설정하고, 프로그램의 버튼 이벤트가 발생하면 상대 발사대에게 미사일 오브젝트(YMissileObject)를 생성하여 발사하도록 한다. 상대 발사대에서 미사일이 발사되었다면, 경로가 같으므로 충돌이 일어나는데 이때 폭발 인터랙션(YBoomInteraction)를 발생시켜 피해 정보를 통보한다. 만약 상대 발사대에서 미사일이 발사되지 않으면, 발사한 미사일이 발사대에 도착하여 충돌이 발생하고 폭발 인터랙션을 발생시켜 발사대에게 피해를 입히도록 한다.



〈그림 6〉 구현된 페더레이트 프로그램

### 3.3.2 구현 내용

구현 시나리오에 따라 <그림 6>과 같은 프로그램을 개발하였다. 하나의 프로그램에서 테스트 페더레이트 모드와 보조 페더레이트 모드로 동작할 수 있도록 하게 하였으며, 이를 통해 인증 절차를 진행하는데 편의를 도모하였다. 또한 <그림 5>와 같은 인증 시나리오 단계에 따라 실행될 수 있도록 Create Federation Execution, Join Federation Execution, Publish & Subscribe Object, Register Object[Launching Pad], Start, Stop, Resign Federation Execution, Destroy Federation




〈그림 7〉 페더레이트 프로그램의 GUI

Execution 버튼을 나누어 기능을 제공한다. <그림 7>은 이러한 시나리오에 따른 GUI(Graphic User Interface)의 구현 모습으로, 미사일을 발사와 충돌의 모습을 보여준다.

## 4. 결 론

본 논문에서는 국제 표준인 HLA의 인증 절차에 따른 HLA/RTI 페더레이트 개발에 도움을 주고자, OO구조를 활용하여 모듈의 재사용성을 높일 수 있는 페더레이트 모델링 및 구조를 제안하였다. 또한 제안된 페더레이트 모델 구조를 이용한 구현물을 활용하여 DMSO의 HLA 인증을 수행함으로써 그 실효성에 대한 검증이 진행 중에 있다.

이러한 HLA Interface Specification 1.3을 활용한 HLA/RTI에 호환 가능한 OO기반의 페더레이트 모델 구조 연구는 협소한 연구 영역 안에서 HLA/RTI를 활용하여 응용을 개발하거나 HLA인증을 고려하고 있는 많은 연구원들에게 연구 가이드라인과 같은 역할을 할 수 있을 것이라 사료된다. 향후 연구 계획으로는 현재 진행 중인 HLA 인증 절차를 완료하고자 하며, 나아가 RTI의 6가지의 서비스 관리 영역 중 하나인 시간관리 메커니즘의 성능 개선 방안을 도출하고, 본 연구에서 제안하는 페더레이트 구조에 적용하기 위한 연구를 진행하고자 한다.

## 참고문헌


〔1〕 IEEE, IEEE Standard for Modeling and Simulation(M&S) High Level Architecture (HLA)-Framework and Rules, Std 1516, 2000.

〔2〕 IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification, Std 1516.1, 2000.

〔3〕 IEEE, IEEE Standard for Modeling and Simulation(M&S) High Level Architecture (HLA)-Object Model Template (OMT), Std 1516.3, 2000.

〔4〕 B. P. Zeigler, H. Praehofer, Tag Gon Kim, Theory of Modeling and Simulation, London, Academic Press, 2000.

〔5〕 Ronghua Zhong, Jian Huang, Bin Chen, Yong Peng, Research on Federate Compliance Test method, ICSC 2008, pp. 851-856, 2008.

〔6〕 Changzheng Qu, Liu Zhang; Yongli Yu, Songshan Wang, Modeling of Equipment Maintenance Support System Domain Library and its Application in HLA Simulation System, ICCEE 2008, pp. 648-651, 2008.

〔7〕 Boukerche, A,; Shadid, A., Ming Zhang, A Formal Approach to RT-RTI Design Using Real Time DEVS, HAVE 2007, pp. 84-89, 2007.

〔8〕 Ahmad, L. Boukerche, A. Al Hamidi, A. Shadid A. Pazzi, R. Web-based e-learning in 3D large scale distributed interactive simulations using HLA/RTI, IPDPS 2008. IEEE International pp. 1-4, 2008.

〔9〕 Ke Liu, Xiaojun Shen, El Saddik, A. Boukerche, A, SimSITE: The HLA/RTI Based Emergency Preparedness and Response Training Simulation, DS-RT 2007. 11th IEEE International Symposium, pp. 59-63, 2007.

〔10〕 홍정희, 안정현, 김탁곤, IEEE 1516 HLA/RTI 표준을 만족하는 시간 관리 서비스 모듈의 설계 및 구현, 한국 시뮬레이션학회 논문지, Vol. 17, No. 1, pp. 43-52, 2008.

〔11〕 Sommerville, Ian, Software Engineering, Addison Wesley Publishing Company, 2000.

## 저자소개

**최 웅 철**(E-mail : wchoi@kw.ac.kr)

1989 서울대학교 컴퓨터공학과 졸업(학사)

1991 서울대학교 컴퓨터공학과 졸업(석사)

2001 Univ. of Illinois, Urbana-Champaign, IL., USA, ComputerScience(Ph.D.)

현재 광운대학교 컴퓨터과학과 부교수

관심분야 정보시스템 감리 및 감사, 정보시스템 보안, 망 설계 및 관리

〈주요저서 / 논문〉

- 이원준, 안상현, 최웅철, "컴퓨터 네트워크," ITC Press, 2005
- Study on L-V-C(Live-Virtual-Constructive) Interoperation for the National Defense M&S(Modeling & Simulation), IEEE ICISS, 2008.
- Multi-channel Enhancement for IEEE 802.11-Based Multi-hop Ad-Hoc Wireless Networks, Lecture Notes in Computer Science(LNCS), 2007.
- Energy Efficient PNC Selection Procedure for the IEEE 802.15.3-Based HR-WPAN(LNCS), 2006.

**유 기 훈**(E-mail : playboyyu@kw.ac.kr)

2007 광운대학교 컴퓨터 소프트웨어학과 졸업(학사)

현재 광운대학교 컴퓨터과학과 석사과정 재학 중

관심분야 무선 네트워크, 무선 센서네트워크, 라우팅 알고리즘 등

〈주요저서 / 논문〉

- Study on L-V-C(Live-Virtual-Constructive) Interoperation for the National Defense M&S(Modeling & Simulation), IEEE ICISS, 2008.