제36회 한국정보처리학회 추계학술발표대회 논문집 제18권 제2호 (2011. 11)

객체 모델을 이용한 HLA기반 시뮬레이션의 게이트웨이 설계 방법

심준용*, 이용헌*, 김세환*
*LIG넥스원(주) Maritime연구소
e-mail:jyshim79@lignex1.com

A Design Method of Gateway for HLA based Simulation using Object Model

Jun-Yong Shim*, Yong-Heon Lee*, Sae-Hwan Kim*
*Maritime R&D Center, LIG Nex1 Co., Ltd.

요 약

HLA(High Level Architecture)는 분산 환경의 모델링 및 시뮬레이션(Modeling & Simulation)을 위한 공통 아키텍처를 제공하는 기술 표준이며, RTI(Run-Time Infrastructure)를 통해 HLA 서비스를 제공한다. HLA는 연동 객체 모델인 FOM(Federation Object Model)을 기반으로 시뮬레이션 환경을 구성하며, 시뮬레이션에 참여하는 모든 시뮬레이터는 동일한 FOM을 소유해야 한다. 따라서 시뮬레이션 체계 간 연동을 수행하기 위해서는 FOM을 통합하거나 FOM 간 연동을 위한 게이트웨이를 구현해야 한다. 한편, FOM을 통합하는 방법은 각 시뮬레이션의 연동 인터페이스 수정이 필요하기 때문에 게이트웨이를 구현하는 방법이 기존 시스템의 변경을 최소화할 수 있다. 따라서 본 논문은 HLA기반 시뮬레이션의 체계 간 연동을 제공할 수 있는 게이트웨이 구조를 제시한다. 특히, XML 형태의 객체 모델을 기반으로 교환 메시지를 정의하고, 메시지 처리 모듈을 게이트웨이에 플러그인 함으로써 시뮬레이션 체계 간 연동의 용이함을 보여준다.

1. 배경

국방 소프트웨어 산업에서는 무기체계 또는 비 무기체계 개발 시 구성 요소의 재사용을 늘리고 체계 간 상호 연동을 제공하는 기술이 핵심이 되고 있다. 특히, 국방 분야의 모델링 및 시뮬레이션(Modeling & Simulation, 이하 M&S) 시스템 구축 시 개발 모델의 이식성을 높이고, 시뮬레이션의 상호 연동성을 높이기 위한 기술 표준으로 High Level Architecture(HLA)의 적용을 늘리고 있다. HLA는 HLA 규칙[1], HLA 인터페이스 명세서(Interface Specification)[2] 및 HLA 객체모델 템플릿(Object Model Template)[3]의 주요 컴포넌트들로 구성되며, 모의 단위인 Federate와 연동 환경인 Federation을 정의한다. 특히, 연동 객체 모델(Federation Object Model, 이하 FOM)을 생성하여 교환 정보를 기술하고, 인터페이스 명세서를 구현한 RTI(Run-Time Infrastructure)를 통해 서비스를 제공한다. HLA기반 시뮬레이션은 FOM을 기반으로 시뮬레이션 환경을 구성하며, 시뮬레이션에 참여하는 시뮬레이터는 동일한 FOM을 소유해야 한다.

최근 시뮬레이션 목적이 다양해지고 복잡해지면서 개발 시뮬레이터를 연합하여 대규모 시뮬레이션 환경을 구축할 수 있는 방법을 요구하고 있다. 한편, HLA기반 시뮬레이션은 시뮬레이터 연합을 위해서 FOM을 통합하거나, 서로 다른 FOM을 연동할 수 있는 게이트웨이를 구현할 수 있다. FOM을 통합하는 방법은 각 시뮬레이션의 연동 인터페이스를 수정해야하기 때문에 기 개발된 시뮬레이터의 경우 적용이 어렵다. 게이트웨이를 구현하는 방법은 기존 체계의 변경을 최소화할 수 있는 반면, 체계 간 교환 메시지가 변경될 때마다 게이트웨이를 수정해야 하는 단점이 있다. 따라서 본 논문은 HLA기반 시뮬레이션 체계 간 연동을 위해 교환 메시지의 변경이 용이한 구조를 갖는 게이트웨이 구조를 제시한다. 제안 구조는 XML 형태의 객체 모델을 기반으로 교환 메시지를 정의하고, 메시지 처리 모듈을 게이트웨이에 플러그인 함으로써 시뮬레이션 체계 간 연동이 용이하다.

논문의 구성은 다음과 같다. 2장에서 HLA기반 시뮬레이션 연동 구조와 특징을 살펴보고, 3장에서 HLA기반 시뮬레이션 체계 간 연동 방법을 소개한다. 4장은 논문에서 제안하는 객체 모델 기반의 게이트웨이 구조를 기술하고, 그 특징을 보여준다. 마지막으로 5장에서 결론을 맺는다.

2. HLA기반 시뮬레이션 연동

HLA는 RTI를 통해 연합관리, 선언관리, 객체관리, 시간관리, 소유권관리 및 데이터분산 관리 서비스를 제공한다. 특히, 선언관리와 객체관리 서비스는 시뮬레이션의 연동 인터페이스인 FOM의 정보를 교환하기 위한 주요 기능을 포함한다.

FOM의 객체모델은 OMT 표준을 준수하는 ObjectClass와 InteractionClass로 구성되며, 시뮬레이션에 참여하는 Federate 간 연동 메시지로 사용된다. 그림 1은 FOM에 작성된 ObjectClass의 일부 구조이다.

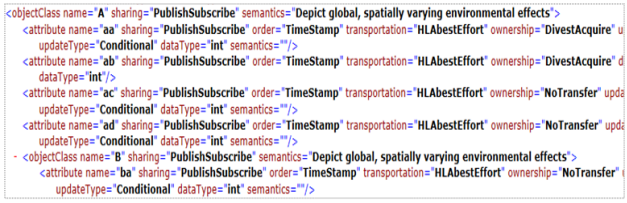

그림 1 FOM의 ObjectClass 구조 예

객체 모델은 상속관계를 이루며, ObjectClass의 속성인 Attribute와 InteractionClass의 속성인 Parameter를 통해 값을 설정한다. RTI는 Publish-Subscribe 방식의 연동구조를[4] 제공하며, Federate는 객체 모델의 'Sharing'속성을 통해서 해당 값이 'Publish'이면 송신, 'Subscribe'이면 수신을 선언한다. Federate 간 데이터 교환 방법은 그림 2와 같다.

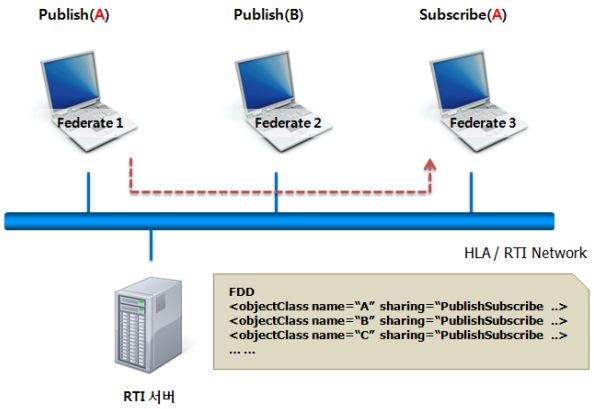

그림 2 Federate 간 데이터 교환 방법

3. HLA기반 시뮬레이션 체계 간 연동 방법

HLA기반 시뮬레이션은 Federation에 참여한 Federate가 동일한 FOM을 통해 연동하기 때문에 다수의 시뮬레이션 체계 연동을 수행하기 위해서는 FOM을 통합하거나 게이트웨이를 구현해야 한다. 각 방법은 다음과 같다.

3.1. FOM 통합 방식

FOM을 통합하는 방식은 다수의 Federation 환경을 하나의 Federation으로 통합하기 위해서 각 Federation이 사용하는 FOM을 하나의 FOM으로 작성하는 것이다. FOM을 통합하면 기존 Federate의 연동 인터페이스를 통합된 FOM에 맞도록 수정해야 한다. 그림 3은 FOM을 통합하여 서로 다른 시뮬레이션 체계를 연동하는 구조를 보여준다. 만약 기 개발된 Federate를 수정할 수 없다면 개별 인터페이스에 대한 어댑터(Adapter)를 구현해야 한다.

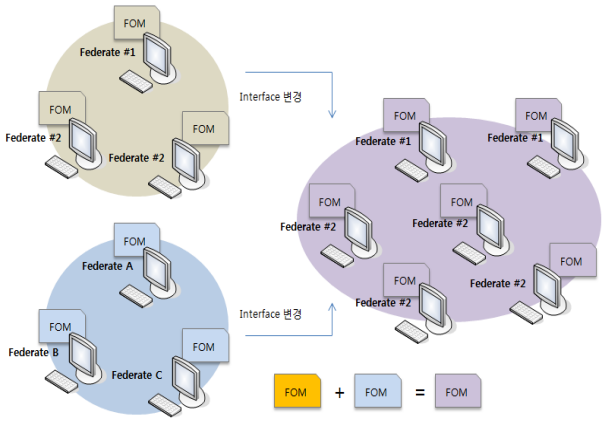

그림 3 FOM 통합 방식

3.2. 게이트웨이 방식

게이트웨이 방식은 각 Federation이 사용하는 FOM으로부터 교환 메시지를 추출하고, 연관시킴으로써 다수 시뮬레이션 연동을 위한 인터페이스를 구현할 수 있다.

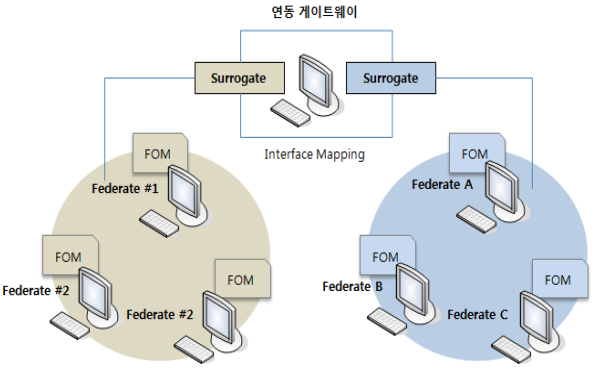

그림 4 게이트웨이 방식

그림 4는 게이트웨이를 적용하여 시뮬레이션 체계를 연동하는 구조를 보여준다. 특히, Surrogate는 Federation에 참여하여 상대 Federation과 교환할 메시지를 전달하는 역할을 수행한다. 게이트웨이 방식은 기존 시스템의 인터페이스 수정을 최소화하기 때문에 FOM을 통합하는 방식에 비해 구현이 용이하다. 반면, 교환 메시지가 변경될 때마다 게이트웨이가 수정되기 때문에 메시지 변경을 용이하게 할 수 있는 구조를 제공해야 한다.

4. 객체 모델 기반의 게이트웨이

본 장은 HLA기반 시뮬레이션 체계 간 연동을 위해 게이트웨이를 설계한다. 특히, Federation 간 교환 메시지의 변경이 용이하도록 교환 정보를 위한 스키마(Schema)와 XML[5] 형태의 객체 모델을 정의하고, 게이트웨이의 각 Surrogate가 플러그인하여 정의된 객체 모델을 통해 메시지를 교환하는 구조를 보여준다. 게이트웨이는 객체 모델을 Instance로 생성하고, 각 Surrogate가 해당 Instance를 공유함으로써 시뮬레이션 연동을 지원한다.

4.1. 교환 정보를 위한 객체 모델

Federate는 시뮬레이션의 목적에 따라 ObjectClass,와 InteractionClass로 구분하지만 게이트웨이는 시뮬레이션 목적에 관계없이 메시지 연동을 위한 역할을 수행하기 때문에 객체 모델의 종류를 구분하지 않는다. 객체 모델의 스키마 파일은 그림 5와 같다.

그림 5 객체 모델 Schema

그림 6은 해당 스키마를 통해 교환 메시지를 작성한 예를 보여준다.

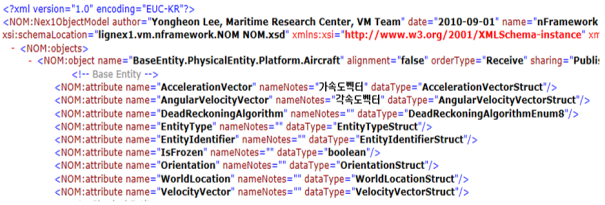

그림 6 객체 모델 기반의 교환 메시지

4.2. 게이트웨이 구조

객체 모델 기반의 게이트웨이 구조는 그림 7과 같다.

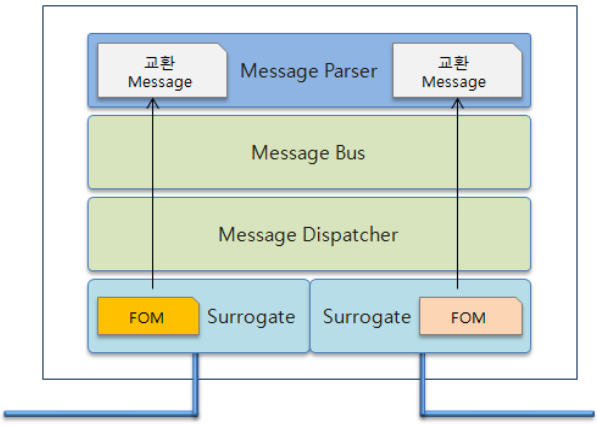

그림 7 게이트웨이 구조

제안 구조는 Message Parser, Message Bus, Message Dispatcher 및 Surrogate로 구성된다.

· Message Parser

각 시뮬레이션 체계의 FOM으로부터 교환 메시지를 추출하여 작성한 객체 모델을 해석하고, 객체 모델의 송수신 여부를 메시지 테이블에 저장한다.

· Message Bus

메시지 테이블에 있는 객체 모델을 Instance로 생성하여 읽기 및 쓰기를 할 수 있도록 공유 메모리를[6] 제공한다.

· Message Dispatcher

Surrogate 간 메시지를 교환하기 위한 프로토콜을 정의하며, Send() 또는 Receive() 함수를 제공한다.

· Surrogate

참여하고 있는 Federation에 접근을 통제하고[7], 해당 FOM으로부터 교환 정보를 얻어온다.

4.3. 교환 메시지 설계

교환 메시지를 설계하기 위해서는 먼저 그림 8과 같이 FOM으로부터 교환 정보를 추출한다.

그림 8 교환정보 추출

교환 정보는 제안 스키마에 맞게 XML을 작성한다.

그림 9 객체 모델 정의

그림 9에서 볼 수 있듯이 FOM A의 'Tank'를 'FOM B의 'GroundVehicle' 객체로 전달하기 위해서 게이트웨이의 객체 모델은 'Tank'를 사용할 수 있으며, Surrogate는 FOM과 객체 모델을 연관시킨다. 연관된 객체 모델의 Instance를 통해 메시지 교환을 수행한다. 만약 교환 메시지가 변경된다면 해당 Surrogate에 해당하는 XML을 수정하고 게이트웨이를 실행하면 된다.

5. 결론 및 향후과제

국방 모델링 및 시뮬레이션 분야에서 분산 시뮬레이션의 표준으로 사용되는 HLA는 연동 모델인 FOM을 통해 참여하는 Federate가 정보를 교환하는 특징을 갖는다. 한편, 하나의 Federation이 하나의 FOM을 사용하므로 서로 다른 시뮬레이션을 연동하는 경우 FOM을 통합하거나 FOM 간 게이트웨이를 구현해야 한다. 본 논문은 이러한 HLA기반 시뮬레이션 체계 간 연동을 지원하기 위해서 게이트웨이를 설계했다. 특히, 교환 메시지를 XML기반의 객체 모델로 정의하고, 각 Surrogate가 플러그인하여 객체 모델을 기반으로 연동하는 구조를 제시했다. 제안 구조는 교환 메시지의 변경과 Surrogate 구성이 용이하다.

향후, 대규모 시뮬레이션 연동을 통해서 게이트웨이의 기능시험 및 성능 검증이 필요하며, 교환 정보의 스키마뿐만 아니라 의미적(Semantic) 객체 모델을 제안하여 연동 인터페이스의 신뢰성을 제공해야 한다.

참고문헌

[1] IEEE, "IEEE Standard for Modeling and Simulation(M&S) High Level Architecture (HLA) - Framework and Rules" IEEE Std. No.: 1516 - 2000

[2] IEEE, "IEEE Standard for Modeling and Simulation(M&S) High Level Architecture (HLA) - Federate Interface Specification." IEEE Std. No.: 1516.1 - 2000

[3] IEEE, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template(OMT)." IEEE Standard No.: 1516.2-2000

[4] Frank Buschmann, Regine Meunier, Hans Rohnet, Peter Sommerlad, Michael Stal, "Pattern-Oriented Software Architecture A System of Patterns", Willey & sons, 1996.

[5] http://www.w3c.org/XML/

[6] http://www.boost.org/doc/libs/1_44_0/index.html

[7] Erich Gamma, Rchard Helm, Ralph Johnson, John Vlissides, "Design Patterns: Elements of Resuable Object-Oriented Software", Addison-Wesley, 1994.