이 기종 체계 연동을 위한 객체 모델 변환 프레임워크 연구

이규호*, 심준용*, 김대영*
*LIG넥스원 Maritime 연구소
e-mail: kyuholee@lignex1.com, jyshim79@lignex1.com, dykim@lignex1.com

A Study on The Framework of Converting an Object Model for Interoperating between Heterogeneous Systems

Kyu-Ho Lee*, Jun-Yong Shim*, Dae-Young Kim*
*Maritime R&D Center, LIG Nex1 Co., Ltd.

요 약

국방 분야의 시스템 개발은 다양한 데이터표준 및 프로토콜로 인해 체계 간 상호 연동에 어려움이 있다. 특히, 다양한 아키텍처가 적용되는 LVC 연동훈련 체계는 적용 프로토콜 간 데이터 교환을 위해서 공통의 표준기술이 필요하다. 본 논문은 이러한 이기종 체계들 간 상호 연동을 위해서 객체 모델 변환 프레임워크를 제시하였다. 제안 프레임워크는 다양한 이기종 체계가 가지는 객체 모델들과 프로토콜들을 구분하여 독립적으로 연동할 수 있도록 하였다. 또한, 체계 간의 종속성을 줄이고 구성 요소의 유연한 설계를 제공하기 위해서 3-Part 지원 모듈의 플러그인 방식 구조로 설계하였다.

1. 서론

최근의 전쟁 양상은 네트워크 중심전(NCW; Network Centric Warfare)이라고 할 만큼 정보의 획득과 전달이 중요해지고 있다. 무기 체계 역시 단독으로 운용되었던 것과 달리 다른 체계(System)들과의 연동을 통해 복합적인 시스템을 구축해가고 있다. 특히 국방 시스템 개발에 있어서 여러 응용체계들 간 정보를 실시간으로 교환 및 공유할 수 있도록 이기종 체계 간 상호운용성(Interoperability)이 강조되고 있다[8].

하지만 지금까지의 국방 체계는 군별 및 무기체계별로 단독의 목적만을 고려한 연통형(stove-pipe) 체계로 개발되어 연동체계 운용을 위한 상호운용성이 어려운 실정이다. 특히 이러한 문제는 실제 시스템(C4I) 간의 연동 뿐만 아니라 시뮬레이션 시스템들(LVC; Live Virtual Constructive)간의 연동에 있어서도 나타나고 있다. 따라서 수많은 시스템 아키텍처들(HLA, DIS, TENA 등)과 객체 모델들이 가지는 복잡성으로 인해 상호운용에 한계를 가지고 있다[3,7].

이러한 문제 해결을 위해 각 나라와 군별로 이기종 체계 간 연동을 위한 다양한 연구가 진행되고 있다. 본 연구에서도 이러한 이기종 체계를 연동하기 위한 관련 연구의 기본 개념을 바탕으로 실제 개발에 있어 필요한 현실적인 대안을 찾고자 한다. 2장에서 기존의 체계 연동 기술에 관해서 알아보고, 3장에서 제안되는 변환 프레임워크 구조를 설명하며 4장에서는 변환 프레임워크 내부의 동작에 관해서 설명하고, 5장에서 결론 및 향후 연구과제를 다룬다.

2. 관련 연구

이기종 체계 간 연동 방법에는 기존에 있는 아키텍처를 수정하지 않고 연동할 수 있는 방식과 공통 아키텍처를 제시하여 기존의 아키텍처에 변화를 가하는 방식이 존재한다. 2.1절의 JLVCDT는 기존의 아키텍처를 수정하지 않고 연동할 수 있도록 변환 게이트웨이 형식을 취하고 있고, 2.2절의 JCOM은 기존의 객체 모델들을 공통모델로 합성하는 방식을 제안하고 있다.

2.1 Joint Live Virtual Constructive Data Translator

JLVCDT는 미국방성의 훈련용 변환 프로그램의 일환으로 통합적인 LVC 환경을 제공하기 위해 개발된 도구이다. 이전의 훈련환경에는 TENA(Test & Training Enabling Architecture), HLA(High Level Architecture), DIS (Distributed Interactive Simulation) 및 CTIA(Common Training Instrumentation Architecture)와 같은 다양한 LVC 아키텍처가 존재하였다. 하지만 이러한 아키텍처를 사용하여 개발된 응용체계들은 각 아키텍처 환경에만 국한되어 사용되었고, 통합적인 환경을 위한 상호운용성에 관한 어떠한 방법도 고려되지 않았다. 이러한 이유로 완전한 LVC 체계의 통합합성환경을 구축하기 위한 중간 단계로서 JLVCDT와 같은 도구를 사용하여 기존 아키텍처를 서로 연계할 수 있는 방법을 제시하고자 하였다[5].

하지만, 이러한 도구 역시 수많은 이기종 체계를 수용하기 위한 완전한 통합환경을 제공하지 않는다[6].

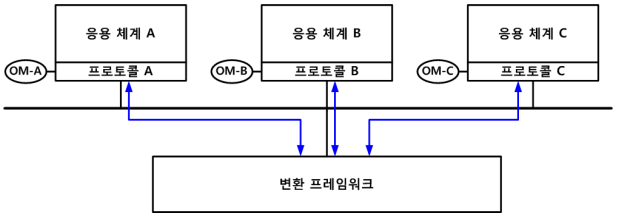
2.2 Joint Composable Object Model

LVC 환경을 목표로 수많은 아키텍처들이 개발되었고 이러한 아키텍처 내에서 다양한 객체 모델(Object Model)들이 사용되었다. 하지만 이러한 객체 모델을 표현하는 방법 사이의 상이성으로 인하여 아키텍처와 별개로 데이터 독립성을 보장받지 못하였다. JCOM 프로젝트는 이러한 객체 모델의 통일성을 위해 계획되었다[4]. 기존 아키텍처에서 사용중인 수많은 객체 모델들을 결합하는 작업은 빈번한 오류와 비호환적인 문제를 내재하고 있다. 따라서 이러한 객체 모델들의 결합과 관련된 문제는 문법적(Synthetic)으로나 의미론적(Semantic)으로 신중히 검토되어야 한다. 본 논문에서는 문법적으로 하나의 공통객체모델을 제시하고, 의미론적으로는 사용자가 결합할 수 있는 환경을 제공하도록 하였다.

3. 이기종 체계연동 변환 프레임워크 구조

3.1 체계 연동 운용개념

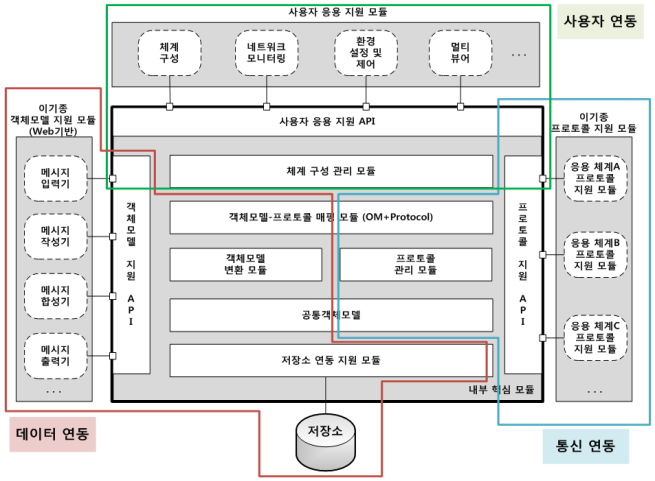
제안된 변환 프레임워크를 사용하여 이기종 응용체계를 연동하는 운용개념은 그림 1과 같다. 이기종 운용체계는 자신들의 객체모델(Object Model)과 프로토콜(통신방식, 미들웨어)이 존재한다고 가정한다. 변환 프레임워크는 크게 두 가지 기능을 갖는다. 첫째는 이기종 응용체계에서 운용중인 객체모델들을 연동할 수 있는 기능이다. 객체모델들은 서로 다른 형식과 의미를 가지고 있기 때문에 프레임워크는 플러그인 형태로 이러한 각자의 객체모델을 읽고 합성할 수 있는 매커니즘만을 제공하고 개발자는 각 응용체계에 맞는 플러그인 모듈을 제작하여 사용할 수 있도록 한다. 둘째는 서로 다른 응용체계에서 사용중인 프로토콜들을 연동할 수 있는 기능이다. 프로토콜은 대부분이 아키텍처에 맞는 미들웨어 형태로 제공되므로 사용자는 쉽게 이러한 미들웨어를 사용할 수 있는 플러그인 모듈을 제작하여 변환 프레임워크에 플러그인(Plug-in)시킬 수 있도록 한다. 이렇게 객체 모델과 프로토콜은 내부적으로 변환 프레임워크에서 매핑되어 서로 다른 응용체계와 연동할 수 있다.



(그림 1) 변환 프레임워크와 이기종 체계 연동 운용개념

3.2 플러그인(Plug-in) 방식 구조

제안된 변환 프레임워크는 그림 2와 같은 구조를 갖는다. 프레임워크는 크게 3-Part(데이터 연동, 통신 연동, 사용자 연동) 연계 기능별로 구조화되어 있으며, 이러한 구조는 개발자가 쉽게 다양한 응용체계를 연동시킬 수 있도록 플러그인 형태로 지원한다. 이러한 플러그인 방식은 다양한 연동 모듈을 처리하기 위해서는 원하는 모듈을 사용자가 쉽게 프레임워크에 설치하고 적용시킬 수 있도록 한다. 또한, 재사용성 측면에서 플러그인 방식의 모듈은 업그레이드 혹은 타 기능으로 전환 시에도 손쉽게 재사용할 수 있다[1].



(그림 2) 변환 프레임워크 내부 구성도

3.3 3-Part 기능 연계 구조

변환 프레임워크는 크게 세 부분의 기능을 담당하고 있다. 3-Part 기능 연계 구조는 데이터 연동, 통신 연동 그리고 사용자 연동으로 이루어진다. 세 개의 파트는 각각 독립적으로 운용되며, 연동을 위해 유기적인 상호작용을 한다. 데이터 연동은 상호운용성을 위해서 서로 다른 체계간 객체 모델을 정의하는 역할을 담당한다. 이는 체계 연동시 필요한 데이터 인터페이스를 정의하는 부분으로 서로 다른 체계에서 사용되는 메시지 포맷과 내용을 공통된 포맷으로 사용할 수 있도록 한다. 기본 원칙은 기존에 사용되는 객체 모델을 변화시키지 않고 프레임워크 내부에서만 공통된 언어로 포맷화하여 변환과 매핑이 이루어지게 하는 것이다. 통신 연동은 메시지를 전달하는 매커니즘을 각 체계에 맞게 정의한다. 이기종 응용체계에서 운용되는 통신 방식은 다양한 프로토콜 형태를 하고 있고, 최근에는 분산 처리를 위해 다양한 미들웨어 방식을 취하고 있다. 이러한 다양한 프로토콜과는 상관없이 변환 프레임워크에서는 통신 방식에 무관하게 내부적인 처리만을 담당하도록 독립적인 구조 형태로 운용된다. 이렇게 이기종 체계에서 사용되는 통신 방식들을 지원하여 데이터와 통신이 결합(Coupling)되지 않고 독립적으로 운용될 수 있도록 한다. 사용자 연동은 제안된 변환 프레임워크를 유용하게 사용할 수 있도록 하는 지원 부분을 담당한다. 이는 특히 사용자가 프레임워크를 통해 체계 연동을 할 수 있도록 하는 편의성에 초점을 맞추고 있다.

3.4 이기종 객체모델 지원 모듈

이기종 객체모델 지원 모듈은 이기종 체계 간 연동 시 서로 주고 받아야 할 메시지를 정의한다. 이는 송신 체계와 수신 체계 사이에 주고 받을 메시지 형태를 연동 전에 사용자의 수동 작성 혹은 내부적인 자동화 과정에 의해 엄격한 룰을 적용받는다. 이러한 메시지는 기존에 사용되는 메시지를 그대로 사용하거나 완전히 새롭게 작성되어야 하는 경우 혹은 기존에 메시지를 수정하는 경우가 있을 수 있다. 어떠한 경우든지 해당 연동 목적에 맞게 사용자에 의해 알맞게 쓰일 수 있도록 이기종 객체모델 지원 모듈에서 지원 기능을 제공한다. 이러한 목적하에 여러 가지 지원 모듈을 사용자가 플러그인 형태로 구현하도록 내부 핵심 모듈에는 객체모델 지원 API를 둔다. 예를 들어 그림 2에서 보듯이 메시지 입력기는 기존에 응용체계에서 사용되는 메시지를 불러올 수 있고 자동으로 읽을 수 있는 기능을 제공한다. 응용체계 마다 기존에 사용되는 객체모델이 있을 것이고 이러한 객체 모델을 읽을 수 있도록 다양한 방식을 제공한다. 각각의 방식 별로 읽을 수 있도록 다양한 입력기가 존재할 수 있다. 메시지 작성기는 새롭게 작성되어야 할 메시지를 정의하는 기능을 제공하며, 메시지 합성기는 작성된 메시지들을 서로 합성하여 또 다른 메시지로 만들 수 있는 기능을 제공한다. 또한 메시지 출력기는 다양한 포맷으로 변환하여 출력할 수 있는 기능을 제공한다. 변환 프레임워크에서는 이러한 공통 메시지 포맷을 XML(eXtensible Markup Language) 형태로 하여 Web 기반을 지원하도록 하고 있다[1,2].

3.5 이기종 프로토콜 지원 모듈

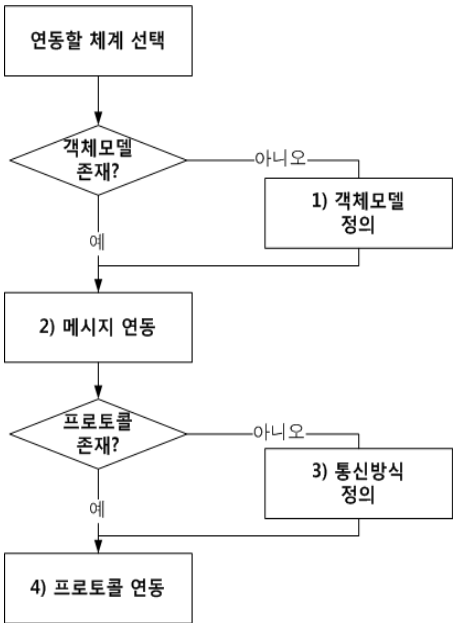
이기종 프로토콜 지원 모듈은 이기종 응용체계에서 사용되는 여러 가지 통신 방식을 플러그인하여 사용하기 위한 모듈이다. 각 응용체계에서 사용되는 프로토콜은 사전에 정의된 방식을 그대로 사용하게 된다. 예를 들어 서버/클라이언트 방식의 TCP/IP 혹은 UDP 통신 방식을 사용하거나 시리얼 통신 방식, CORBA와 같은 broker 중심의 미들웨어 통신, 혹은 시뮬레이션 통신 방식 중에 하나인 RTI 미들웨어를 사용하는 것이 그 예이다. 이러한 통신 방식들은 각 표준에 따라서 다양한 모듈 형태로 지원된다. 따라서 이러한 이기종 프로토콜을 지원하기 위해서 사용자는 각 프로토콜을 지원하는 플러그인 형태의 프로토콜 지원 모듈을 만들어 변환 프레임워크에 연결할 수 있다. 이를 위해 내부 핵심 모듈에는 이러한 프로토콜 지원을 위한 프로토콜 지원 API를 두고 있다.

3.6 사용자 응용 지원 모듈

사용자 응용 지원 모듈은 변환 프레임워크를 위해 꼭 필요한 기능은 아니지만, 사용자에게 유용한 도구를 지원하기 위해 제공되는 모듈이다. 예를 들어 그림 2에서 체계 구성은 연동할 체계 구성을 설계할 수 있거나 배치할 수 있도록 하는 기능을 제공한다. 네트워크 모니터링은 현재 네트워크에 미치는 영향을 사용자가 실시간으로 모니터링 할 수 있도록 한다. 환경 설정 및 제어는 변환 프레임워크에서 사용하게 될 환경 변수 및 파라미터를 입력하고 각 연동 체계를 제어할 수 있도록 한다. 멀티 뷰어는 다양한 그래픽 도구를 지원할 수 있다는 것을 의미한다. 이러한 모듈들은 플러그인 형태로 각 사용자에 의해 제작될 수 있도록 내부 핵심 모듈에서 사용자 응용 지원 API를 제공한다.

4. 내부 모듈 동작

변환 프레임워크는 사용자에 의해 결합된 3-Part의 플러그인을 내부적으로 동작시키는 내부 핵심 모듈을 가지고 있다. 사용자는 그림 3에서와 같은 순서를 통해 응용체계들을 연동하게 된다.



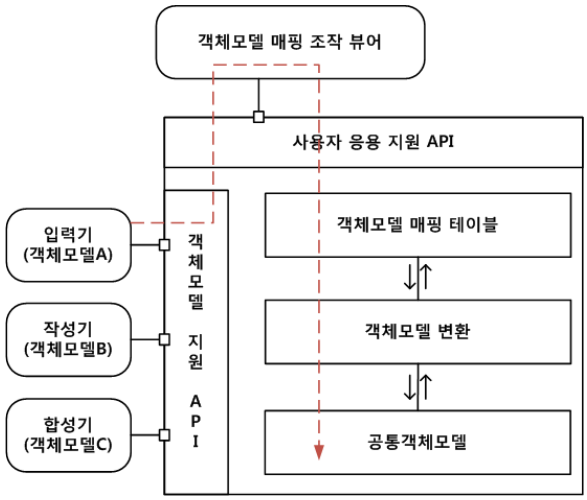
(그림 3) 응용체계 연동 순서도

1) 객체모델 정의

객체모델이 존재하지 않는 응용체계이므로 새롭게 객체모델을 정의할 필요가 있다. 선택된 응용체계에서 사용될 객체모델은 독립성과 재사용성을 위해 사용자가 직접 입력할 수 있는 방식을 택해 가시화할 수 있도록 한다. 사용자는 변환 프레임워크로부터 공통객체모델을 제공받아 새롭게 객체 모델을 정의할 수 있다.

2) 메시지 연동

그림 4와 같이 객체모델이 존재하는 가운데 사용자는 객체모델 뷰어를 통해 응용체계의 객체 모델을 볼 수 있다. 예시와 같이 객체 모델 매핑 테이블은 사용자가 정의해서 공통객체모델과 의미적으로 매핑될 수 있도록 하여 정의한다. 이렇게 작성된 객체모델은 객체모델 변환 모듈을 통해 공통객체모델로 변환된다.



메시지	공통객체모델	객체모델A	객체모델B	객체모델C
메시지1	M1	A.M1	B.M1	C.M1
메시지2	M2	A.M2	B.M2	-
메시지3	M3	-	B.M3	C.M3
…	…	…	…	…

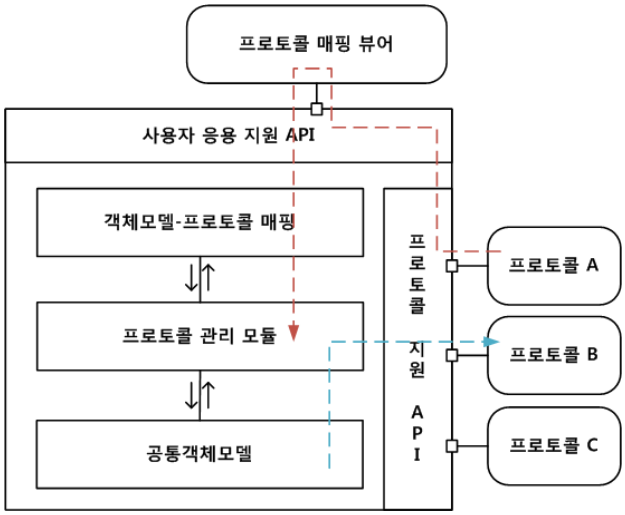
(그림 4) 객체모델 연동방법과 매핑 테이블 예시

3) 통신방식 정의

프로토콜이 지원되지 않는 응용체계는 응용체계에서 사용될 통신방식을 정의할 필요가 있다. 기존에 플러그인된 프로토콜 지원 모듈에서 선택하는 방식 또는, 새롭게 플러그인 지원 모듈을 작성하는 방식으로 나뉜다.

4) 프로토콜 연동

그림 5와 같이 응용체계에서 사용되는 통신방식이 정의되어 해당 프로토콜 지원 모듈이 내부 모듈과 연동되어야 한다.



메시지	공통객체모델	응용체계A	응용체계B	응용체계C
메시지1	M1	P, S	-	S
메시지2	M2	S	P	P, S
메시지3	M3	S	S	P
…	…	…	…	…

(그림 5) 프로토콜 연동방법과 매핑 테이블 예시

선택된 프로토콜은 앞에서 선택된 객체모델과 매핑 테이블을 통해 다시 한번 매핑된다. 변환 프레임워크에서는 다양한 프로토콜 방식을 지원하기 때문에 통신방식은 사용자가 선택할 수 있다. 응용체계별로 송수신되는 메시지를 정의하여 해당 프로토콜로 통신이 이루어질 수 있다. 예시에는 Publish-Subscribe 방식일 경우를 가정하여 송수신하는 메시지와 응용체계 간 매핑을 보여준다.

5. 결론

지금까지 이기종 체계연동을 위한 변환 프레임워크를 제안하였다. 변환 프레임워크는 연동하고자 하는 응용체계를 3-Part(데이터 연동, 통신 연동, 사용자 연동)로 구분된 플러그인 방식의 지원 모듈을 통해 연동할 수 있도록 하여 유연성과 재사용성을 높이고 있다. 하지만 실제 응용체계에서 사용되는 객체모델과 프로토콜은 많은 구현상의 도전과제를 가지고 있다. 공통객체모델을 만들기 위해선 보다 정교한 객체지향 기술을 요구하고 있다. 공통객체모델과 더불어 공통 프로토콜에 대한 설계 및 구현 이슈는 지속적인 연구대상이 될 것이다.

참고문헌

[1] 심준용, 조규태, 이용헌, 이승영, 김세환, “플러그인 아키텍처 프레임워크를 위한 Publish-Subscribe 메시지 프로토콜 설계”, 2010.05

[2] 이용헌, 심준용, 김세환, “RPR-FOM 기반 HLA 시뮬레이터 개발을 위한 XML 기반 Object Model의 적용”, 2010.10

[3] Francis H. Carr, “C4ISR/Sim Technical Reference Model Applicability to NATO Interoperability”, MITRE Technical Paper(2003.10)

[4] Richard Rheinsmith 외, “Joint Composable Object Model and LVC Methodology”, 2009 I/ITSEC Conference

[5] Warren Bizub, “The Joint Live Virtual Constructive Data Translator Framework Interoperability for a Seamless Joint Training Environment”, RTO-MP-MSG-045

[6] Gary W. Allen, “Live, Virtual, Constructive, Architecture Roadmap Implementation and Net-Centric Environment Implications”, ITEA Journal 2010

[7] 유호동, 정구돈, “NCW 구현을 위한 전술시스템 연동 방안”, 제1194호(08-10)

[8] 2010국방백서, “야전부대의 과학화 교육훈련”, pp.148