

# 데이터 중심의 국방 시뮬레이션 네트워크 미들웨어에 관한 연구

이규호\*, 심준용\*, 위성혁\*, 김세환\*  
 \*(주)LIG넥스원 S/W연구센터  
 e-mail:kyuho.lee@lignex1.com

## A Study on the Data-Centric Middleware for the Military Simulation Network Environment

Kyu-Ho Lee\*, Jun-Yong Shim\*, Soung-Hyouk Wi\*, Sae-Hwan Kim\*  
 \*S/W R&D Center, LIG Nex1 Co., Ltd

### 요 약

국방 시뮬레이션 소프트웨어 개발 시 급변하는 요구사항과 반복적인 구현에 빠르게 대처하기 위해 플러그인 기반의 개발프레임워크가 사용되어왔다. 그러나 개발된 소프트웨어는 내부 컴포넌트의 재활용성은 증대되었으나, 수많은 노드가 연결되는 분산 시뮬레이션 환경에는 적합하지 못했다. 이에 기존의 개발된 데이터 모델인 NOM을 기반으로 객체 간 통신이 가능하고, 분산 환경에서 노드의 확장이 유연한 통신 미들웨어인 *nComm+*를 제안한다.

### 1. 서론

국방 시뮬레이션 소프트웨어를 개발함에 있어 어려운 점은 급변하는 요구사항에 대응해야 함과 동시에 유사한 기능의 반복된 구현에 있다. 이러한 어려움을 해결하기 위해 소프트웨어 재사용성을 높이고자 플러그인 기반의 객체모델을 이용한 개발 프레임워크인 *nFramework*를 개발하였다[1]. 개발자는 재사용의 주 단위인 플러그인 모듈을 구현 시 NOM(Nex1 Object Model)[2]이라는 객체모델을 이용하여 XML 기반의 인터페이스를 정의한다.

플러그인 기반의 컴포넌트를 재사용하거나, 간단히 요구사항에 맞는 컴포넌트를 추가하면 되므로 보다 빠른 개발이 가능해졌다. 하지만 이 프레임워크로 개발된 프로그램은 하나의 어플리케이션 개발을 위한 용도만으로 사용되었다. 이 때문에 다른 네트워크 기반의 어플리케이션과의 통신을 위해서는 개발자 각자가 HLA/RTI[3]와 같이 통신 모듈을 따로 만들어야 한다.

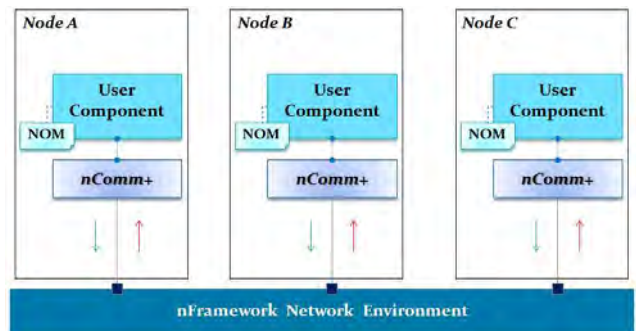
또 다른 문제점은 노드 간 다양한 연결 고리를 가질수록 통신 컴포넌트의 복잡도는 높아져 이를 이용한 소프트웨어의 성능이 크게 떨어진다는 점이다. 네트워크에 연결된 노드가 많아질수록 명시적으로 송신자와 수신자를 나타내야하므로 데이터 중심의 네트워크 환경에서는 그 복잡도가 높아지게 된다. 또한 기존의 NOM 사용자 정의 객체 모델의 상위 개념을 그대로 다른 노드와 공유할 수 없는 단점을 가지고 있다.

따라서 본 논문에서는 *nFramework*에 특화된 NOM 객체모델을 이용하여 분산 네트워크로의 확장이 용이한 Publish-Subscribe 기반의 통신 미들웨어인 *nComm+*를

제안하였다.

### 2. 계층화된 *nComm+* 구조

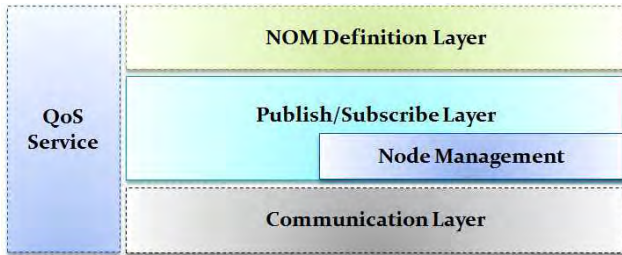
*nComm+*를 이용한 통신 개념은 그림 1과 같다. 사용자는 XML 기반의 NOM을 인터페이스로 객체 모델을 정의하며, *nComm+* 통신 미들웨어를 사용하여 다른 노드의 *nComm+*를 통해 해당 NOM을 네트워크로 송수신하는 방법을 제공한다. 따라서 이 시뮬레이션 환경에 다른 노드가 추가되기 위해서는 해당 XML을 이용해 NOM을 정의하고, *nComm+* 라이브러리를 이용하여 네트워크에 가입하는 것으로 절차는 단순화된다.



(그림 1) Communication Conception with *nComm+*

그림 2와 같이 *nComm+*는 3개의 계층으로 구성되며, QoS Service가 독립적으로 구성되어 다른 계층들과 관계를 맺는다. NOM Definition Layer는 사용자가 정의한 NOM을 기반으로 이를 재해석하여, 데이터를 사전에 정의

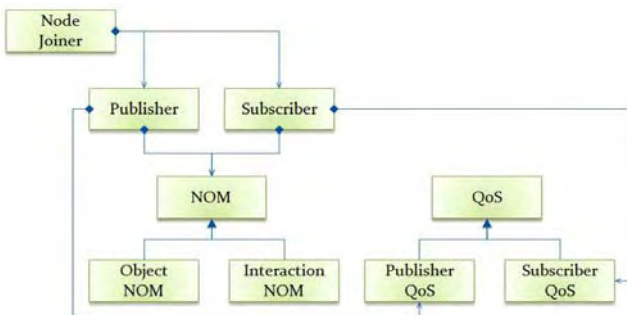
할 수 있도록 한다. 사용자는 객체화된 데이터를 컴포넌트에서 사용하는 그대로 통신 데이터로 사용한다. 다음은 확장성을 위해 Publish-Subscribe 방식을 이용한 NOM의 발행/구독을 담당하는 Publish/Subscribe Layer가 존재한다. 또한, 이 계층은 가입된 노드를 관리할 수 있는 Node Management를 특별히 가지고 있다. 실제 통신은 Communication Layer에서 담당하며, 이 부분은 다양한 플랫폼에 종속되게 동작할 수 있게 구성된다. 마지막으로 통신 간 사용자의 요청과 응답을 위해서 QoS Service를 두어 다양한 서비스 품질을 제공할 수 있도록 하였다.



(그림 2) Layered Structure in nComm+

**3. 확장성을 높이기 위한 Publish-Subscribe 방식**

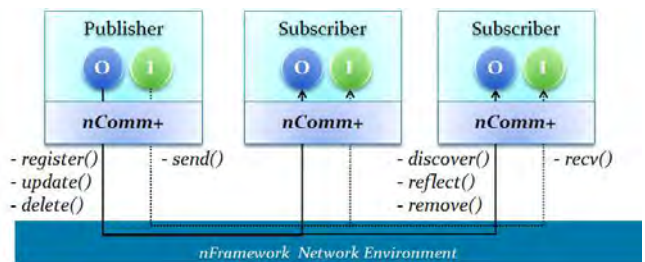
현재 플러그인 모듈이 사용하고 있는 객체 모델은 그림 3의 구조와 같이 실제 송수신 객체를 Object와 Interaction 두 가지 형태로 나누어 정의하고 있다.



(그림 3) Software Class Relationship of nComm+

Object는 실행 중에 메모리에 영속적이며, 속성(attribute) 값만이 변경되는 성질의 데이터로 모델링되고, Interaction은 비영속성을 가지며, 단순한 이벤트성 데이터로 정의된다. nComm+에선 이러한 사용자에 의해 정의가 가능한 NOM을 그대로 통신 데이터로 사용한다. 사용자는 XML에 Object와 Interaction을 구분하여 정의하고, 송신측에서는 추상화된 데이터인 NOM에 값을 설정하여 보내고, 수신측에서도 이 NOM을 그대로 받아서 실제 내부의 값을 확인할 수 있도록 한다. 기존의 TCP/UDP의 문제점을 극복하기 위해서는 명시적인 송수신자의 존재를 통신 미들웨어에게 위임하는 것이다. 이것은 기본적으로 데이터 중심의 네트워크 환경에서 데이터를 생산하고, 소비하는 형태의 Publish-Subscribe 방식을 요구한다. 그림 4와 같

이, Publisher는 NOM을 통해 보내고자 하는 데이터를 미리 등록(register)하고, 객체 속성이 변하면 갱신(update)하고, 객체를 삭제(delete)한다. 한편, Subscriber에서는 받고자 하는 NOM을 미리 선언하면, nComm+에서 등록된 데이터가 있으면 이를 발견(discover)할 수 있게 하고, 갱신되면 발견된 객체에 반영(reflect)되게 하며, 삭제되었을 경우 발견된 객체도 제거(remove)될 수 있게 한다. 이러한 방식은 송수신자를 명시적으로 지정하는 기존의 방식이 아니라 관심이 되는 데이터를 중심으로 프로그래밍 환경이 변경되었기 때문에, 노드가 늘어날수록 확장된 네트워크를 구성할 수 있게 한다. 노드가 늘어도 자신의 송수신할 데이터 객체를 선언하기만 하면, 원하는 데이터를 얻고 보낼 수 있게 된다.



(그림 4) Publish-Subscribe Method

**4. 결론**

국방 시뮬레이션 환경을 구성 시 사용자가 사용하기 쉽고, 자연스럽게 통신이 구성되게 하는 것은 중요한 이슈사항이다. 이에 본 논문에서는 기존의 통신 모듈이 고려하지 못하였던 분산 네트워크 환경 및 데이터 중심의 환경으로 이동하는 시점에서 데이터 모델을 구조화하여 추상화된 객체를 송수신 하게 하는 통신 미들웨어를 제안하였다. 또한, 노드가 늘어날수록 복잡해지는 통신의 복잡도를 해소하기 위해 Publish-Subscribe 방식을 이용하였고, 실제 통신 시 발생 가능한 사용자 QoS를 미리 설정할 수 있도록 하여, 네트워크의 부하를 사용자 요청에 의하여 조절 가능하도록 하였다.

**참고문헌**

- [1] 심준용, 이용현, 김세환, “단일 및 분산 컴퓨팅 환경에서 모델 연동을 위한 플러그인 방식의 인터페이스 설계 방안 연구”, 한국정보처리학회 춘계학술발표대회 논문집 제18권 제1호, 2011.05
- [2] 이용현, 심준용, 김세환, “사용자 정의 형식을 지원하는 XML 기반 객체 모델의 구조 설계”, 한국정보처리학회 추계학술발표대회 논문집 제17권 제2호, 2010.11
- [3] IEEE, “IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification.” IEEE Standard Mo: 1516.1-2000.