

DDS 어플리케이션 개발 및 운영을 위한 플랫폼 설계 방안

정해영, 이성훈, 홍형섭, 최선*, 전승훈

한전KDN

hyj_young9@kdn.com, sunghoon.2@kdn.com, ph.d.hong95@kdn.com, *the.best.6150@kdn.com, junsh_5440@kdn.com

Platform for Development and Operation of DDS Application

Jeong Hae Young, Lee Sung Hoon, Hong Hyung Sub, Choi Sun*, Jeon Seung Hoon

Kepco KDN

요약

DDS는 최근 빅데이터, IoT 등 4차산업혁명으로 급증한 통신 개체를 수용하고 대량 데이터를 효율적으로 배포하기 위한 통신 미들웨어로 급부상하고 있다. 이에 국내외 다양한 산업분야 통신시스템으로 DDS를 적용 및 검토하는 사례가 꾸준히 늘고 있다. 하지만 DDS 표준 객체들에 대한 기본적인 개념과 다양한 옵션들을 이해하고 있어야 원활한 구현이 가능하며, 또한 통신노드의 자유로운 도메인 참여 및 탈퇴가 가능하다는 DDS의 특징으로 인해 통신노드의 종합적인 감시와 통제가 어려운 운영상의 단점도 존재한다. 이에 본 논문에서는 DDS 어플리케이션의 개발 또는 운영 시 편의성을 향상시킬 필수 S/W들을 조사하고, 이들을 포함하는 DDS 플랫폼 설계 방안에 대해 제시하고자 한다.

I. 서론

최근 IoT, CPS 등의 기술발달로 대규모 분산환경에서 발생하는 데이터를 실시간으로 전송하기 위한 미들웨어의 요구가 증가[1]하고 있다. 특히, 다양한 기기 간 상호운영되고, 시스템간 데이터 연계가 가능하도록 표준화된 통신 방식을 요구[2]하고 있으며, 이에 국제표준 통신 미들웨어인 DDS가 주목받고 있다. DDS는 2015년 로봇운영체제인 ROS2의 통신표준으로 채택되었고, 국내에서는 한국정보통신기술협회에서 CPS 통신표준으로 제시하는 등, 주요산업에서의 DDS 적용은 더욱 확대될 것으로 예상된다. 하지만 시스템에 DDS를 적용하고 운영하기 위해서는 DDS 엔진만 가지고는 불편한 상황들이 존재한다. DDS 응용시스템 구현시, 통신을 위해 필요한 엔티티의 종류와 설정값이 다양하여 DDS 관련 경험과 기본 지식을 필요로 한다. 또한 통신의 의명성을 기반으로 자유롭고 유연한 연결을 지향하는 DDS 특성상 안정적인 운영을 위해서는 통신 네트워크의 감시 및 제어를 위한 별도의 도구가 필수적이다. 따라서 본 논문에서는 DDS 응용시스템의 효율적인 구현 및 운영을 지원하는 DDS 플랫폼을 제시하고자 한다.

II. 본론

1. DDS통신

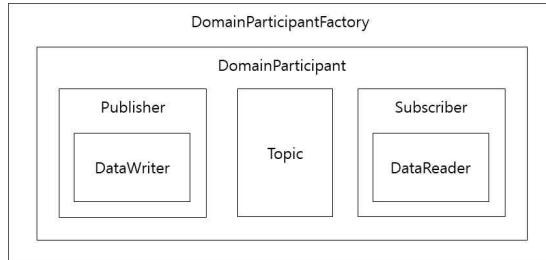


그림 1. DDS 엔티티 구조

DDS 통신 구현을 위해 기본적으로 필요한 엔티티의 종류는 [그림2]와 같이 7가지이다. DomainParticipantFactory는 가장 최상위 엔티티로, 시스

템에 하나만 존재하는 싱글톤 객체이다. DomainParticipant는 도메인에 참여하는 통신노드의 기본단위가 되는 엔티티로, 참여할 도메인 번호를 지정할 수 있다. Publisher와 DataWriter는 데이터 발간 프로그램에 생성하는 엔티티로, 데이터의 발간과 관련된 역할을 한다. Subscriber와 DataReader는 데이터 구독 프로그램 측에 존재하는 엔티티로, 데이터의 구독과 관련된 기능을 수행한다. Topic은 통신할 데이터를 나타내는 엔티티로, DataWriter와 DataReader에 각각 하나의 토픽이 연결된다.

발간응용과 구독응용이 통신하기 위해서는 1차적으로 각 DomainParticipant의 도메인 ID가 동일해야 한다. DDS는 구동시에 도메인ID값을 이용하여 계산된 통신 포트번호를 적용하기 때문에 도메인ID가 다른 응용끼리는 포트번호가 상이하여 통신할 수 없다. 이후에는 동일 도메인에 참여중인 통신노드끼리 멀티캐스트 기반의 Discovery과정을 거쳐 Topic이 동일한 노드를 찾아내고, 이를 간에 데이터의 송수신이 이루어 진다.

2. DDS 플랫폼 구조

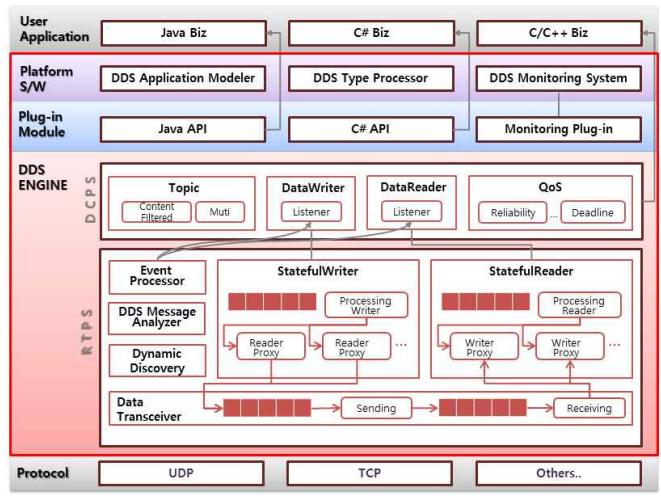


그림 2. DDS 플랫폼 구조

[그림 2]은 DDS 플랫폼의 구성을 나타낸 그림이다. 플랫폼에는 DDS 엔진 외에 Plug-in Module과 Platform S/W로 구성된다. 엔진은 주로 C/C++로 구현되기에 Java 및 C#용 어플리케이션 구현을 지원하기 위한 API를 Add-on 형태로 지원한다. 또한 DDS 네트워크 모니터링을 위해 모니터링 데이터를 실시간으로 생성 및 발간하는 Monitoring Plug-in 모듈도 추가되었다. Platform S/W는 DDS 어플리케이션의 구현이나 운영을 원활하게 하기 위해 필요한 단독 실행가능한 형태의 S/W로, 구현 시 필요한 응용 모델러, 타입 처리기와 운영시 필요한 모니터링 시스템이 있다.

DDS 어플리케이션을 구현할 때는 앞의 절에서 설명한것과 같이 DomainParticipant, Topic, Publisher, Subscriber, DataWriter, DataReader 등의 엔티티를 생성 및 구성하고, 각각의 엔티티에 20가지가 넘는 설정값을 세팅해주어야 한다. 이러한 작업들을 사용자가 직관적으로 그래픽 UI를 통해 할 수 있도록 도와주는 프로그램이 응용 모델러이다.

타입처리기는 사용자가 통신하고자하는 데이터 타입을 idl 형태로 선언하면, 원하는 언어로 변환해 주는 역할을 한다. 이때 사용자 정의타입의 직렬화/역직렬화 코드까지 자동 생성되어 개발 부담을 덜어줄 수 있다.

DDS는 넓은 지역의 대규모 통신노드가 Plug&Play 구조로 별도의 설정없이 네트워크에 연결 즉시 통신이 가능하여 통신 네트워크 구성이 실시간으로 변경되는 특성을 갖는다.[3] 이러한 동적인 DDS 네트워크를 원격으로 감시 및 제어하기 위한 S/W가 모니터링 시스템이다.

3. DDS Application Modeler

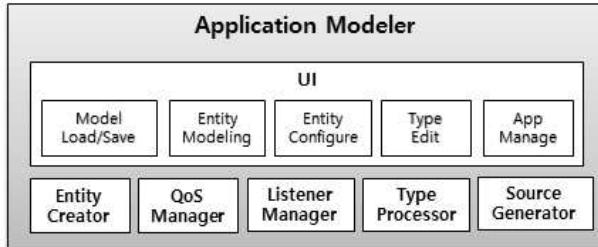


그림 3. DDS 응용모델러 구조

응용모델러는 크게 UI부와 5가지의 모듈로 구성되는데, UI 그래픽 편집기를 통해 Drag& Drop으로 DDS 엔티티를 손쉽게 구성하며, 각종 설정 값 및 데이터 타입을 편집할 수 있다. DDS 어플리케이션 모델링이 완료되면, 그에 맞게 엔티티들을 생성해주는 Entity Creator, QoS와 리스너 등을 설정해주는 QoS/Listener Manager, 통신 데이터 타입을 생성해주는 Type Processor, 이들을 종합하여 최종적으로 소스코드를 생성해주는 Code Generation으로 구성된다. 응용모델러는 대부분의 코드를 자동으로 생성해주어 개발자로 하여금 최소한의 비즈니스 구현만 하면 되도록 지원하는 도구로, DDS를 잘 모르는 개발자도 간단하게 응용 구현이 가능하다.

4. DDS Type Processor

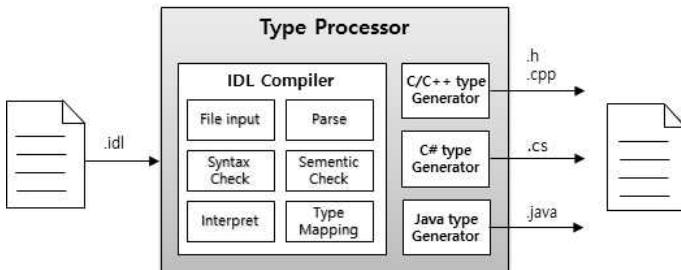


그림 4. DDS 타입처리기 구조

타입처리기는 그림 4와 같은 구조로, idl 파일의 해석 및 타입 맵핑을 지원하는 IDL Complier모듈과 C++, Java등의 언어로 변환하여 소스코드

를 생성해주는 Type Generator 모듈로 구성된다. 생성되는 소스코드에는 DDS 통신을 위한 타입 직렬화/역직렬화 코드가 포함되며, 개발자는 DDS 어플리케이션 프로젝트에 생성된 소스파일을 import만 하면 된다.

5. DDS Monitoring System

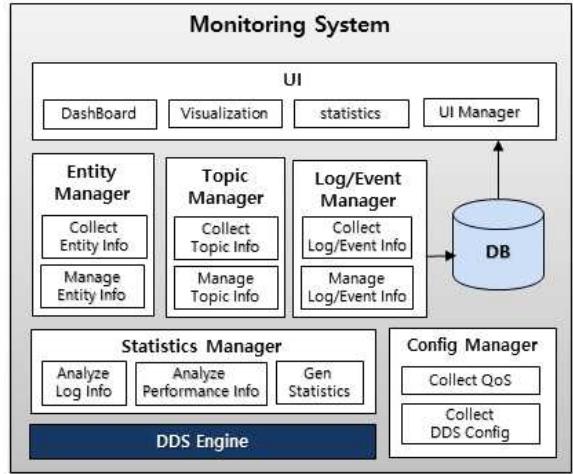


그림 5. DDS 모니터링 시스템 구조

Monitoring Plug-in 모듈을 장착한 DDS 통신노드는 특정 모니터링 전용 도메인으로 모니터링 데이터를 주기적으로 발간한다. 단, Log/Event 정보는 이벤트성 데이터로 실시간으로 발간한다. 모니터링 시스템에서는 이러한 Entity, Topic, Log/Event, Config 정보를 DDS통신을 통해 수집하여 DB에 저장하며, DB 데이터를 기반으로 각종 분석 및 통계 등의 연산을 수행한다. 사용자는 UI를 통해 DB에 저장된 모니터링 데이터를 조회 및 제어한다. 모니터링 시스템을 통해 DDS 네트워크에 참여중인 통신 노드들의 구성정보와 실시간 상태정보, 상세정보 등을 확인가능하며, 평균 성능 및 통신이력 통계정보 등을 조회할 수 있다.

III. 결론

본 논문에서는 DDS 어플리케이션을 구현하고 운영하는데 있어 사용자의 불편함을 제거하고, 편의성을 향상시키기 위해 필요한 S/W들을 조사하고 설계하여, DDS 플랫폼을 구성하는 방안에 대해 설명하였다. 국내에서는 아직 DDS 적용을 연구·실증하는 단계로 사례가 많지 않아, 다소 생소한 미들웨어일 수 있지만, 본문에서 제시한 플랫폼을 활용하면 DDS 적용에 대한 어려움을 극복할 수 있을 것으로 기대된다. 하지만 이외에도 실산업에서 안정적으로 적용하기 위해서는 WAN환경 운영을 위한 라우터, 각종 서비스 게이트웨이, 보안모듈 등 필요한 S/W는 더 존재할 것으로 판단된다. 향후 연구에서는 DDS의 한계 및 산업특성을 보다 심층적으로 분석하여 DDS 플랫폼을 확장하고자 한다.

참 고 문 헌

- [1] 김대근, 박만곤, 인더스트리4.0 기반 사이버물리시스템과 생산관리시스템간의 미들웨어 구축을 통한 수평적 통합, 한국멀티미디어학회 논문지, 1484-1493, 2014
- [2] Seongjin Yun, JunHong Park, WonTae Kim, Data-centric middleware based digital twin platform for dependable cyber-physical systems, 2017 Ninth International Conference on Ubiquitous and Future Networks(ICUFN)
- [3] 김현옥, 최선, 데이터분산서비스 기반 전력계통 통신미들웨어 테스트를 위한 도구, 한국컴퓨터종합학술대회 논문지, 1327-1328, 2018