

DDS vs OPC-UA: 두 산업용 IoT 표준

DDS는 데이터-중심적이고 OPC-UA는 장치 상호운용성을 위해 설계되었습니다.

OMG의 DDS와 OPC-UA (Open Platform Communications-Unified Architecture)는 일반적으로 산업용 IoT(IIoT)에서 데이터 및 정보 관리를 위한 가장 적합한 기술로 간주됩니다. 그러나 두 기술이 어떤 차이가 있는지 그리고 두 기술은 각각 어느 IIoT 사용 사례에 적합한지에 대해 많은 혼란이 남아 있습니다. 이 기사에서는 IIoT에서 DDS 및 OPC UA의 차이점과 적합성에 대해 명확히 정의하고자 합니다.

우선 두 IoT 표준이 어떻게 생겨났는지 살펴 보겠습니다.

DDS and OPC-UA 기원

DDS: 1990년대 후반 미 국방부는 정보 우위 확보를 위해 글로벌 정보 그리드 (Global Information Grid) 구축을 제안하였습니다. GIG는 전투원, 정책 입안자 및 지원 인력에게 정보를 수집, 처리, 저장, 보급, 관리 할 수 있는 일련의 기능을 제공하는 것이었습니다.

CORBA, COM+/DCOM과 같이 당시에 널리 사용 된 클라이언트/서버 기술은 GIG를 구현하는 데 적합하지 않았습니다. 단단한 결합(tight coupling), 오류에 대한 높은 민감도, 확장성 및 성능 문제 등 GIG에 필요한 요구사항을 맞출수가 없었기 때문이었습니다.

DDS는 이러한 한계를 극복하고 실시간 상황 인식, 운영 인텔리전스 및 작전 계획을 위한 GIG의 데이터 공유에 대한 요구 사항을 해결하기 위해 도입되었습니다. 많은 이들이 GIG는 오늘날 우리가 말하는 IoT의 원형이라고 생각합니다.

그 이후 DDS는 네트워크 중심 시스템 및 GIG와 같은 기능을 위한 기반 기술로서 전 세계적으로 추천/권장되어 많은 곳에 적용이 되었습니다. 이러한 대규모 적용에는 항공 교통 관제 및 관리, 무인항공기, 우주항공 애플리케이션, 통합모듈 형 차량전자화(vetronics), 교육 및 시뮬레이션 시스

템 및 해군 전투 시스템 등입니다.

OPC: OCP-UA의 아버지격인 Open-Platform Communications는 1990년대에 시작되었습니다. 당시 산업자동화 업계에는 제어기와 필드 디바이스와의 상호작용에 대한 표준이 없었습니다. 결과적으로 HMI와 같은 클라이언트 애플리케이션은 상호 작용해야 하는 모든 디바이스에 드라이버와 프로토콜을 내장해야 했습니다. "통합 악몽"의 시대였습니다.

1996년 클라이언트 애플리케이션이 자동화 장비의 세부 사항을 알아야 하는 고역을 없애고 제어 하드웨어 및 필드 디바이스와 상호 작용하는 표준화된 인터페이스를 제공하기 위해 OPC (OLE for process control)가 도입되었습니다.

당초에 OPC specification은 OLE, COM 및 DCOM (distributed component object model) 기술을 기반으로 Microsoft 플랫폼에만 독점적으로 적용되었습니다. Microsoft가 2002년에 DCOM의 죽음을 선언하자, OPC 커뮤니티는 비 Microsoft 플랫폼에서 OPC를 지원해야 한다는 요구와 함께 기술 노후화 문제를 처리해야 했습니다.

이러한 문제를 해결하기 위한 첫 번째 시도로 SOAP/XML에 의존하는 새로운 OPC Data Access 표준의 정의가 대두되었습니다. 이러한 새로운 표준은 이식성의 문제는 해결할 수 있었지만 SOAP/XML으로 인한 오버헤드로 인해 여러 산업적 사용 사례에 호환이 되지 않았습니다.

결과적으로 OPC 커뮤니티에서는 새로운 표준의 필요성이 대두되었고 이로 인해 2009년에 OPC-UA 1.0 버전이 나타나게 됩니다. 이 과정에서 OPC는 "OLE for process control"로부터 "Open Platform Communication"으로 이름이 변경되었으며 새로운 표준은 OPC-UA (OPC Unified Architecture)로 불리어지게 되었습니다.

DDS and OPC-UA evolution

이제 두 기술의 표준과 진화를 살펴 보겠습니다.

DDS: DDS specification은 분산 애플리케이션 통신 및 통합을 위한 DCPS (Data-Centric Publish-Subscribe) 모델을 정의합니다. 이 규격은 정보 생산자로부터 소비자에게 정보를 효과적으로 전달할 수 있는 API 와 communication semantics (행동 및 quality of service)를 정의합니다.

또한 이 표준은 서로 다른 벤더의 DDS간의 상호 운용성을 가능하게 하는 DDS 상호 운용성 와이어 프로토콜 (DDSI-RTPS)을 정의합니다. DDS는 애플리케이션이 자율적이고 비동기적으로 데이터를 읽고 쓸 수 있는, 공간적 및 시간적 디커플링 상태에서도 사용이 가능한, 가상화 된 데이터 공간(data space)을 구현하는데 사용할 수 있습니다. DDS specification의 목적은 "적시에 적절한 위

치에 올바른 정보를 효율적으로 전달"할 수 있게하는 것입니다.

최근 표준에 추가 된 내용은 다음과 같습니다.

- DDS-Security는 저장된 상태의 데이터 뿐만 아니라 이동 중인 데이터까지의 포괄적 정보 중심 보안 프레임 워크입니다.
- DDS-RPC는 DDS 애플리케이션 간의 request-reply 상호 작용을 가능하게 합니다.
- DDS-Xtypes : DDS 애플리케이션에 대해 확장 가능하고 dynamic topic type을 사용 가능하게 합니다.

향후에는 DDS-XRCE(eXtremely Resource Constrained Environments) 표준 제정이 예정되고 있으며 이를 통해 100KB 미만의 RAM과 매우 제한된 네트워크로 연결되는 것 같이 매우 극소한 IoT도 지원이 가능할 예정입니다. 마지막으로, 다가오는 DDS/OPC-UA 게이트웨이 표준은 두기술의 완벽한 통합을 제공 할 것입니다.

OPC-UA: OPC-UA는 다양한 시스템 및 장치가 다양한 네트워크를 통해 클라이언트와 서버간에 메시지를 보내어 통신할 수 있는 플랫폼 독립적 표준입니다. 클라이언트와 서버의 신원을 보장하고 공격에 강하고 강력하고 안전한 통신을 지원합니다. OPC-UA는 서버가 제공 할 수 있는 서비스 세트(sets of service)를 정의합니다.

정보는 OPC-UA 정의 및 공급 업체 정의 데이터 유형을 사용하여 전달되며, 서버는 객체모델을 정의하고 클라이언트는 이를 동적으로 발견 할 수 있습니다.

서버는 사용자에게 주요 변경 사항을 알리기 위해 현재 및 과거 데이터뿐만 아니라 경보 및 이벤트에 대한 액세스를 제공할 수 있습니다.

OPC-UA는 독립적으로 개발된 애플리케이션 간에 복잡한 데이터를 교환하기 위한 프로토콜과 서비스를 제공합니다. OPC-UA는 정보를 교환하는 클라이언트와 서버간의 구문적(syntactic) 상호 운용성을 가능하게합니다.

Comparing DDS and OPC-UA

DDS와 OPC-UA는 모두 분산 시스템의 정보 관리 문제를 해결합니다. 둘 다 정보 모델링을 지원합니다. DDS는 관계형 데이터 모델링을 지원하며 OPC-UA는 객체 지향 모델링을 지원합니다. 그러나 유사점은 여기까지 입니다.

- DDS 추상화(abstraction)은 시간과 공간에서 애플리케이션을 분리하는 **분산 데이터 공간**

을 중심으로 이루어집니다. OPC-UA 추상화는 **클라이언트-서버**를 중심으로 이루어집니다.

- DDS 애플리케이션은 **익명적 그리고 비동기적으로** 글로벌 데이터 공간(global data space)에서 **데이터를 읽고 쓰는 방식**으로 상호 작용합니다. OPC-UA 애플리케이션은 하나 이상의 UA 서버에서 **요청(request)**를 호출하여 상호 작용합니다.
- DDS 애플리케이션은 **완전한 위치 투명성(complete location transparency)**을 제공합니다. 데이터는 관심이 있는 곳으로 자동으로 이동합니다. OPC-UA 애플리케이션은 **두 단계의 해결 과정**을 거쳐야 합니다. 첫째, **서버를 검색하고** 두 번째로 **주소 공간에서 데이터를 찾아야**합니다.
- DDS는 **리소스 중심의 선언적(declarative)** 프로그래밍 스타일을 선호합니다. 무엇이 어떻게 되어야 하는지만 선언하면 됩니다. OPC-UA는 **명령형(imperative)** 프로그래밍 스타일을 선호합니다. 무엇을 어떻게 해야 하는지 명령이 필요합니다.
- DDS 데이터 모델링은 **관계형**입니다. DDS 정보 모델은 query, join, project될 수 있습니다. 그러나 OPC-UA 데이터 모델링은 **객체지향적**입니다. 서버가 지원할때에만 OPC-UA를 찾아 쿼리할 수 있습니다.

Part 2. Security Model	DDS-Security
Part 3. Address Space Model	DDS
Part 4. Services	There is no server concept in DDS. Negotiation is done via the discovery services
Part 5. Information Model	DDS, DDS-XTypes
Part 6. Service Mappings	Platform Specific Models of each specification define implementation when necessary
Part 7. Profiles	Each Specification has its own conformance points.
Part 8. Data Access	DDS, DDS-XTypes
Part 9. Alarm & Conditions	Alarms / Events are represented as topics in DDS
Part 10. Programs	DDS-RPC
Part 11. Historical Access	DDS, DDSI-RTPS, DDS provides built-in support for history. In addition DDS vendors also support seamless integration with time series stores
Part 12. Discovery	DDS, DDSI-RTPS. DDS Has built-in decentralised discovery
Part 13. Aggregates	DDS promotes micro-service architectures, thus aggregates are typically provided by analytics, or similar applications

그림 1. OPC-UA 및 DDS 의 대응

- DDS의 Dynamic Discovery는 애플리케이션 과 데이터가 자동으로 발견되는 매우 동적인 시스템 구축이 가능합니다. 애플리케이션은 관련 정보가 발견되면 즉시 통보를 받게 됩니다. DDS는 즉시 사용할 수 있는 "플러그 앤 플레이" 기능을 갖추고 있습니다. 반면에 OPC-UA 애플리케이션은 명시적으로 사물을 "검색"해야 합니다. 즉, "플러그 앤 플레이"

를 위해서는 추가 프로그래밍 방식의 기술이 필요합니다.

- DDS는 비 기능적 속성의 캡처를 위해 정보에 주석(구성 가능한 QoS 정책)을 허용합니다. OPC-UA는 QoS 사양을 지원하지 않습니다.
- DDS Security는 저장되어 있는 데이터뿐 아니라 이동중인 데이터까지 처리합니다. 또한 DDS Security는 인증, 액세스 제어, 암호화, 로깅을 제공합니다. OPC-UA 보안은 클라이언트와 서버간 안전한 채널을 설정하는데 중점을 둡니다.

결론

이 기사에서 가장 중요한 점은 DDS와 OPC-UA가 비슷한 문제를 매우 다른 방식으로 다루고 있다는 것입니다. DDS는 데이터 중심적이며 GIG와 유사한 시스템과 대규모 데이터 통합을 지원하기 위해 설계되었습니다. peer-to-peer 통신에 대한 지원은 IIoT 시스템에서 디바이스, 엣지 노드, 엣지 애플리케이션, 클라우드 애플리케이션 간의 데이터 공유를 지원합니다.

OPC-UA는 장치 상호 운용성을 지원하고 센서, 내장 장치 및 PLC 컨트롤러를 대형 시스템에 통합하도록 설계된 장치 중심 기술입니다. OPC-UA는 자동화 산업에서 출현하여 주로 사용되며, 따라서 생태계가 가장 큰 강점입니다.

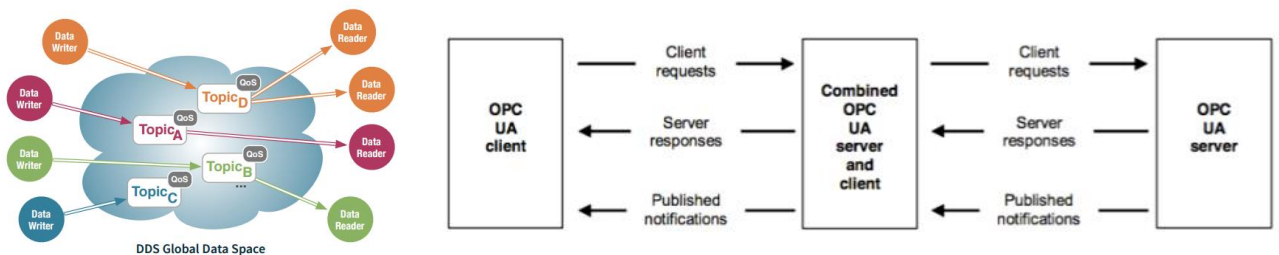


그림 2. DDS vs OPC-UA