

전력운영시스템 데이터 교환을 위한 메시지 버스

오도은*, 정남준*, 송재주*, 양일권*, 김현수**

*한전 전력연구원

**충남대학교 전기정보통신공학부 컴퓨터전공

e-mail:hifive@kepri.re.kr

A Message Bus for Data Exchange of Power Operation Systems

Do-Eun Oh*, Nam-Joon Jung*, Jae-Ju Song*, Il-Kwon Yang*, Hyeon-Soo Kim**

*Korea Electric Power Research Institute

**Computer Science & Engineering, Chungnam National University

요 약

전력시스템은 발전에서 수요에 이르기까지 다양한 설비 및 기기들이 유기적으로 결합되어 있고, 모든 과정이 실시간으로 운영되고 있는 매우 큰 규모의 시스템이다. 전력시스템에 IT 기술이 접목되면서 다양한 구성요소 간에 수많은 정보교환이 요구되는 시점에서 각 운영시스템별 독립적인 필요에 의해 상이한 플랫폼으로 개발되었기 때문에, 시스템 상호간 연계와 원활한 데이터 교환 및 유통을 통한 전체 전력시스템의 최적화를 이루는 데까지는 도달하지 못하고 있다. 다양한 운영시스템 상호간 데이터 교환을 위한 표준화 요구에 부응하여, 향후 전력시스템은 표준화되고 공개된 정보모델을 활용한 전사적 통합시스템 개발이 요구된다. 본 논문에서는 전력운영시스템 데이터 교환을 위한 메시지 버스에 대하여 기술하였다. 서로 다른 운영환경에서의 단위 운영시스템들을 동일한 플랫폼으로 재구성, 재개발하는 것이 어려운 상황에서 전력시스템 내 유통정보를 총체적으로 관리하고 이를 필요로 하는 운영시스템에 효율적으로 제공할 수 있는 인프라의 구축은 미래의 전력시스템의 부가가치를 높일 수 있는 기반을 제공할 것이다.

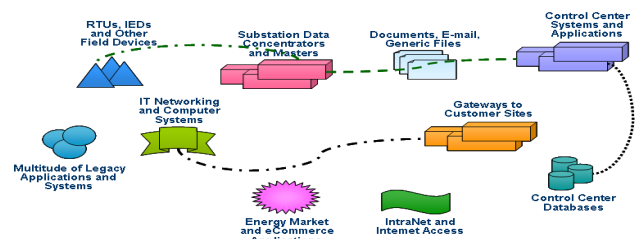
1. 서론

지난 수 십 년 동안 전 세계적으로 유례없는 속도의 정보통신기술 발전이 이루어져 왔고, 이 기술들이 빠른 처리능력과 편리성 제공을 목적으로 사회 전반에 도입되었듯이 다른 기술 분야에 비해 상대적으로 발전 속도가 느리다고 인식되어온 전력분야에도 어김없이 도입되었다. 이러한 정보통신기술은 전력산업이 시작된 이래 고수되어 온 수동처리 업무 방식을 빠르고 간편한 자동화 방식으로 전환함으로써 업무처리 능력은 대폭 향상되었으나, 독자적인 필요에 의해 상이한 플랫폼으로 개발되어 왔기 때문에, 시스템 상호간에 연결이 되고, 필요한 정보를 교환 처리하여 전체 전력망의 최적화를 이루는 데까지는 도달할 수 없었다. 한편, 전 세계적으로 전력산업 디지털화가 확대되면서 전력공급자에 의한 중앙집중형·단방향 제어 및 정보의 흐름을 Plug-and-Play가 가능한 양방향·분산형 전력시스템 구조로 전환하고자 하는 요구가 증가하고 있다. 또한 전력시스템 전 영역에서 발생하는 데이터의 활용과 이를 통한 자가 치유형 전력망으로 일컬어지는 지능형 전력시스템을 구현하기 위한 다양한 연구들이 시도되고 있다[1]. 이러한 지능형 전력시스템은 각 시스템별 독립적 필요에 의한 부문별 자동화가 아닌 정보의 통합화, 제어의 분산화 그리고 기능의 상호운용성을 통해 전력시스템이 하나의 유기적인 시스템으로서 동작하는 체계

를 필요로 한다. 본 논문은 전력산업 디지털화가 확대되면서 전력시스템 전 영역에 걸쳐 발생하는 새로운 데이터의 활용과 이를 이용한 새로운 서비스 개발을 위한 전력 운영시스템 데이터 교환을 위한 메시지 버스에 대하여 기술한다.

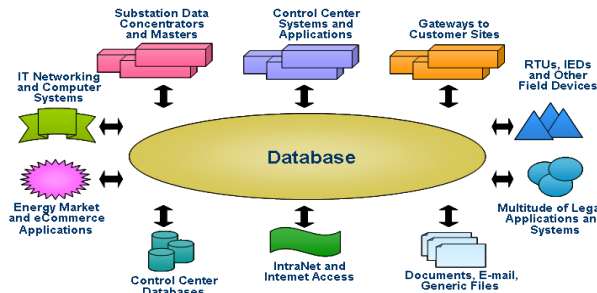
2. 전력운영시스템 데이터 교환방식 발전 형태

전력시스템의 여러 운영시스템 간 데이터 교환을 위하여 가장 먼저 시도된 방식은 ‘전력계통 데이터’라는 공통의 데이터를 이용하여 각자의 기능을 수행하는 형태이다. 이 방식은 중앙의 데이터베이스를 공유하여 사용하였고, 운영시스템 간 데이터 교환을 위해서는 데이터베이스를 통하여 간접적으로 전달하거나 또는 직접 point-to-point 방식으로 연결되었다. 그러나 이러한 방식은 운영시스템을 추가하고 통합하는데 그림 1에서 보는바와 같이 ‘Island of Automation’ 형태의 문제점을 가지고 있다.[1]



(그림 1) ‘Islands of Automation’방식의 통합

그 다음으로 그림 2와 같이 데이터베이스를 중심으로 한 애플리케이션 통합 방식이 도입되었는데 초기에는 문제가 없지만, 점차 유지보수가 힘든 단점이 있다. 만약 데이터베이스의 테이블이 하나라도 변경된다면 데이터베이스와 관련 있는 다른 운영시스템들도 모두 수정을 해야 한다. 즉 확장하거나 업그레이드하는데 시간과 비용이 과다 발생하게 된다.



(그림 2) 데이터베이스 중심의 데이터교환

3. 공통정보모델

공통정보모델은 전력분야 에너지 관리 및 배전시스템에서 적용되는 공통 언어이다. 공통정보모델에 대한 규격은 국제표준인 IEC 61970에 자세히 설명되어 있다. 공통정보모델 자체에 대한 설명은 301, 302에 기술되어 있으며, 공통정보모델 데이터 교환에 대한 내용은 501과 503에 자세히 설명되어 있다.[2][3]

공통정보모델은 전력계통의 모든 구성요소를 객체지향 방법을 사용하여 모델링 한 것으로서 UML(Unified Modeling Language)을 이용해서 정의한다. UML은 다른 프로그램 언어에 독립적인 모델링 언어이며, 이것을 객체지향 언어인 C++, Java, C#등을 이용하면 구조적인 변경 없이 그대로 구현할 수 있다. 시스템 통합에 활용된 통신 인터페이스는 CORBA(Common Object Request Broker Architecture)의 IDL(Interface Definition Language)을 이용하여 인터페이스를 정의하고 있고, 개발언어에 독립적인 인터페이스를 정의하고 있다. 그리고 향후에는 이것을 C++, Java, C#, Web Services 방식으로 구현하기 위한 명세서가 작성될 계획이다.

4. 엔터프라이즈 서비스 버스

인터페이스를 구현하기 위한 분산객체 기술은 크게 CORBA, DCOM, RMI, Web Service 방식으로 나눌 수 있는데, DCOM은 마이크로소프트 진영의 분산객체 기술이고 유닉스 계열의 OS에서는 실행될 수 없다. RMI는 Java 진영의 기술로 Java VM(Virtual Machine)이 설치된 곳이면 실행될 수 있다. CORBA와 Web Service 기술은 마이크로소프트 계열과 Java 계열에서 모두 구현할 수 있다. SOA의 핵심 목표인 비즈니스 함수의 가상화는 서비스 정의와 사용을 서비스 구현으로부터 분리시킴으로서 실현

된다. 서비스는 IBM WebSphere MQ, IBM CICS, IBM IMS, Java2 Platform, Enterprise Edition(J2EE) Enterprise JavaBeans(EJB), Java classes, IBM DB2 Queries, Java Message Services(JMS), Microsoft .NET 등 다양한 기술들을 사용하여 구현된다. Enterprise Service Bus(ESB)는 참여자들 사이에서 발생하는 서비스 인터랙션의 관리와 가상화를 지원하는 아키텍처 패턴이다. 서비스 공급자와 요청자들 사이를 연결하고, 정확히 매치되지 않더라도 인터랙션을 가능케 한다. 이 패턴은 다양한 미들웨어 기술과 프로그래밍 모델을 사용하여 구현될 수 있다. 그림 3은 일반적인 ESB 패턴을 나타낸 것이다. 엔드포인트가 ESB와 인터랙팅하는 사이트를 서비스 인터랙션 포인트(SIP)라고 한다. SIP는 웹 서비스 엔드포인트, WebSphere MQ 큐, RMI 원격 객체용 프로록시가 될 수 있다. 서비스 레지스트리는 SIP의 요구 사항과 기능, 다른 SIP와 인터랙팅을 하는 방법(동기식, 비동기식, HTTP, JMS), QoS 요구사항(보안, 신뢰성, 인터랙션), 기타 SIP 정보 등을 설명하는 메타데이터를 파악한다.[4]

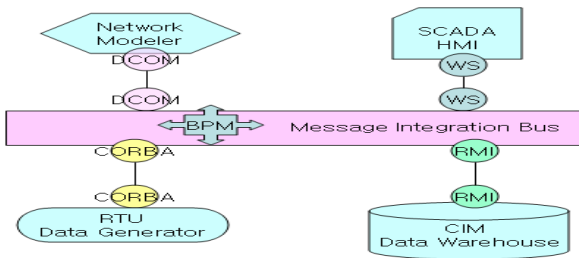
5. OMG DAF, DAIS

OLE for Process Control(OPC)은 각종 애플리케이션들이 서로 다른 프로세스 컨트롤 장비(DCS, PLC 등)와 데이터를 주고받을 수 할 수 있게 적용된 표준 인터페이스다. 애플리케이션들은 각기 다른 여러 종류의 OPC 호환 서버(DCS, PLC 등)들로부터 데이터를 주고받는데 단지 하나의 OPC 호환 드라이버만 설치하면 된다. OPC가 산업체에서 널리 받아들여 활용되었으나, 기본적으로 마이크로소프트 Windows를 바탕으로 하고 있기 때문에 유닉스 시스템을 주로 사용하는 전력계통에서는 활용하기가 불가능 했다. 또한 OPC는 공장자동화 같은 정도의 소형 시스템에 주로 적용되었기 때문에 좀 더 일반화될 필요가 있었다. 이에 Open Management Group(OMG)에서는 Data Access Facility Specification(DAF)와 Data Acquisition from Industrial System(DAIS)라는 스펙을 정의 하였다. DAF는 산업시스템에서 데이터 교환을 효과적으로 하기위해 개발되었고 데이터를 읽어들이거나 내보낼 수 있을 뿐만 아니라 데이터 모델도 전달할 수 있도록 설계되었다. 또한 데이터에 변경사항이 발생하였을 때 통지할 수 있는 기능도 고려되었다. DAIS는 OPC를 대형 산업시스템에도 적용할 수 있도록 확장한 형태이다.

6. 메시지 버스

운영시스템 간 데이터 교환을 하는데 있어서 기존의 시스템과 통합은 필수적이다. 하지만 기존의 시스템은 IEC 61970의 표준 인터페이스를 적용하지도 않았을 뿐만 아니라 서로 다른 분산객체 기술을 사용하고 있다. 서로 다른 분산객체 기술은 서로 호환되지 않는다. 다음 그림 3

은 전력운영시스템 데이터 교환을 위한 IEC의 공통정보모델을 이용한 메시지버스를 나타낸다. 메시지버스는 여러 분산객체 기술을 한꺼번에 탑재할 수 있다. 따라서 앞서 언급한 분산객체기술인 DCOM, CORBA, Web Service, RMI를 모두 동시에 탑재가능하고 이들 기술 간에 연동을 가능하게 한다. 또한 IEC 61970의 GID를 구현하지 않은 운영시스템들에 대해서도 메시지 변환을 통해서 연동이 가능하다.



(그림 3) 메시지 버스를 이용한 데이터 교환 방식

다음은 WSDL 정의와, DAF, DAIS Web Service 가운데 DAF Web Service에 대해 설명한다.

○ WSDL의 정의

WSDL은 SOA 컴포넌트 간에 인터랙션을 위한 IDL로서, 데이터 타입과 메시지, 바인딩, 서비스에 대한 정의를 한다. WSDL에서는 wsdl:types 항목에서 데이터 타입을 기입한다. 메시지는 일반 프로그램에서 함수를 호출할 때 사용되는 입출력 변수라고 할 수 있다. 입력변수의 타입을 미리 정의하고 공개함수를 정의하는 portType에서 사용된다. 바인딩은 wsdl:binding 형태로 외부에 노출할 port들을 기술하고, SOAP메시지의 style과 use를 정의한다. 서비스는 최종적으로 서비스가 제공되는 부분을 기술하는 부분이다. 서비스의 최상단은 Port이며, 여기에 바인딩 방법과 서비스의 주소가 작성된다.

○ DAF web Service 설계

▪ 데이터 타입 정의

기본 데이터타입으로 String으로 된 URI와 URI의 배열인 URISequence 그리고, 두 개의 long long 타입으로 이루어진 ResourceID 구조체이다. 제공되는 함수로는 ResourceIDSequence를 반환하는 get_resource_ids, URISequence를 반환하는 get_uris로 되어 있다. DAF에서 가장 핵심적인 데이터 타입이 SimpleValue이다. 이것은 모든 형태의 값을 하나의 타입으로 접근하기 위해 정의되었다. 설계된 데이터 타입의 개수는 다음과 같다.

<표 1> DAF 데이터 타입

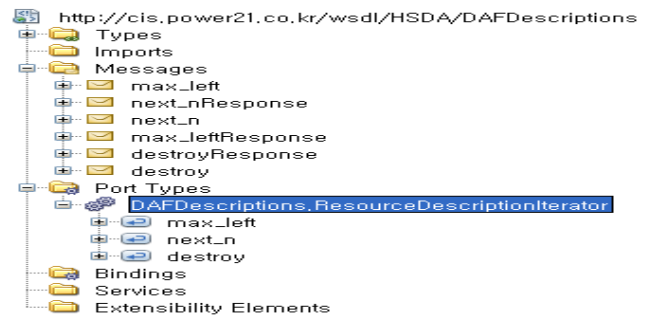
모듈명	Simple Type	Complex Type
DAFIdentifier	1	4
DAFDescription	4	20

▪ WSDL 구현

OMG의 DAF에서 정의된 인터페이스를 웹서비스로 구현하기 위하여 앞서 정의된 데이터 타입을 이용하여 WSDL을 작성한다. DAF에서는 크게 DAFIdentifier와 DAFDescription 두 가지의 모듈이 존재한다.

▪ Web Service 구현

PropertyValue와 ResourceDescription 두 개의 타입을 추가로 정의하였고, 이를 이용하여 ResourceDescriptionIterator에서 max_left(), next_n(), destroy() 세 개의 함수를 제공한다. 다음은 Web Service로 구현된 구조이다.



(그림 4) DAFDescription WSDL 구조도

7. 결론

본 논문에서는 IEC 공통정보모델과 애플리케이션 통합 기술을 이용한 전력운영시스템 데이터 교환을 위한 메시지 버스에 대하여 기술하였다. 메시지 버스는 여러 분산객체 기술을 한꺼번에 탑재할 수 있다. 따라서 다양한 분산객체기술인 DCOM, CORBA, Web Service, RMI를 모두 동시에 탑재가능하고 이들 기술 간에 연동을 가능하게 한다. 또한 IEC 61970의 GID를 구현하지 않은 운영시스템들에 대해서도 메시지 변환을 통해서 연동이 가능하다.

참고문헌

[1] Intelligrid Architecture Volume I, EPRI, 1999
 [2] IEC 61970-301, “EMS-API Part 301 : Common Information Model Base”
 [3] IEC 61970: Energy Management System Application Program Interface (EMS-API) Part 402: Common Services.
 [4] Websphere Business Integration Primer: Soa, Web Services, and Esb, Michele Chilanti, Vinod Jessani, 2007.12