

GID 표준인터페이스를 이용한 SCADA 메시지 교환에 관한 연구

오도은*, 정남준*, 송재주*, 양일권*, 김현수**
한전 전력연구원*, 충남대**

A Study on SCADA Message Exchange using Generic Interface Definition

Do-Eun Oh*, Nam-Jun Jung*, Jae-Joo Song*, Il-Kwon Yang*, Hyeon-Soo Kim**
KEPRI*, Chungnam University**

Abstract - 전력계통 SCADA시스템은 그 목적상 실시간의 대규모 측정정보 취득을 목적으로 한다. 취득정보는 RTU(Remote Terminal Unit)를 통해서 SCADA시스템에 전해진 후, 필요에 따라 상위 시스템으로 다시 전송된다. 이때 RTU에서 SCADA 시스템에 취득 포인트 정보는 DNP(Distribute Network Protocol) 프로토콜을 이용해서 전송하며, RTU는 측정값의 변동에 상관없이 데이터를 SCADA 시스템에 전송한다. SCADA 시스템을 구축하기 위해서는 우선 RTU에 감시 선로 정보를 입력하고 이 정보를 다시 SCADA 시스템의 DB에 포인트 정보와 매핑을 수행해야한다. 따라서 SCADA 시스템을 구축하는데 비용과 시간이 많이 소요되고 운영 시에는 불필요한 데이터 요청과 응답이 발생하여 네트워크 트래픽을 유발한다. IEC에서 제안하는 GID 표준인터페이스를 사용할 경우에는 좀더 효과적인 데이터 교환이 가능하다. 본 논문은 GID 표준인터페이스를 이용한 효과적인 SCADA 메시지 교환에 관해 기술하였다.

1. 서 론

IEC에서 제정한 61970 시리즈는 EMS 애플리케이션의 API를 정의하고 있다. EMS는 전력계통 컨트롤센터에서 전력계통을 운영하는 중앙 시스템이다. EMS에는 전력계통 운영을 위한 여러 애플리케이션이 탑재되어 있다. EMS에서 취급하는 데이터는 전력계통 데이터 그 자체이고, 전력계통의 특성상 전력계통을 구성하고 있는 요소의 형태와 거의 유사하다. 지금까지는 각 EMS 제조회사마다 고유의 데이터모델을 정의하여 사용하였기 때문에, 제조회사가 개발한 애플리케이션만을 실행 할 수 있었다. 따라서, 사용자는 제조회사에 종속적일 수밖에 없었고, 애플리케이션을 유지보수 하는데 많은 시간과 비용을 할애하였다. 또한 더 좋은 기능을 제공하는 계통운영 애플리케이션이 개발되었다하더라도 제조회사가 제공하지 않는다면 사용할 수 없었다.

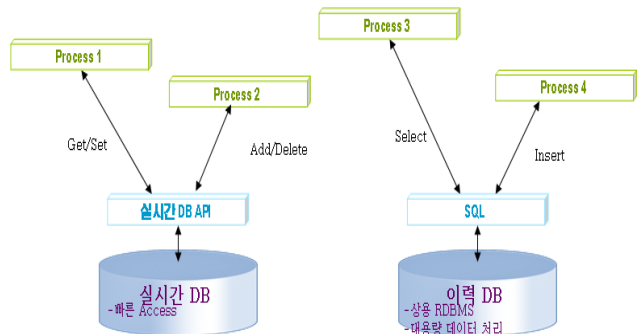
SCADA 시스템은 EMS의 하위 시스템으로서 계통을 감시하고 제어하는 기능을 수행한다. EMS에서는 SCADA 시스템에서 취득된 데이터를 사용하므로, 두 시스템이 동일한 형태의 데이터를 취급할 수 있어 SCADA에도 IEC 61970규격을 적용할 수 있게 되는 것이다. IEC 61970을 적용한 SCADA 시스템은 현재의 SCADA 시스템과 다른 구성을 가지고 있는데, 가장 큰 차이점은 애플리케이션 간에 데이터 교환을 위해 공통정보모델이 사용된다는 점이다. 현재의 SCADA 애플리케이션의 정보교환 방식은 취득 포인트 매핑에 의한 방법이 주를 이루었지만 IEC 61970을 적용한 SCADA 시스템은 계통구성에서부터 취득정보까지 모두 공통정보모델 데이터를 기반으로 하고 있다. IEC 61970에서는 또한 애플리케이션에서 공통정보모델 데이터를 효과적으로 교환하기 위한 표준 인터페이스를 정의하고 있다. 이것은 애플리케이션의 종류에 상관없이, 심지어 데이터모델에 상관없이 데이터를 교환할 수 있다. 본 논문은 공통정보모델 기반의 GID 표준 인터페이스를 적용한 효과적인 SCADA 메시지 교환에 관해 기술하였다.

2. 본 론

2.1 현재의 메시지 교환 방식

현재의 SCADA시스템의 데이터교환방식은 시스템 제작회사에서 개발한 DB API를 이용하는 것이었다. 이 방식은 각 프로그램마다 제공된 API를 직접 호출하여 데이터를 교환하는 것이다. 또한 대부분의 SCADA 시스템은 효율성을 위해 실시간 데이터와 이력데이터를 별도로 구분하여 저장하였는데, 실시간 데이터를 위한 DB는 메모리기반 DB를 사용하고 이력 데이터를 위한

DB는 상용의 관계형 DBMS를 주로 사용한다. 아래 그림은 현재의 SCADA 시스템에서의 데이터 교환방식을 나타낸 것이다.



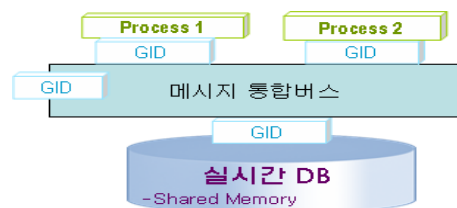
〈그림 1〉 현재의 SCADA시스템 데이터 교환 방식

실시간 DB는 제조회사에서 제작된 고유의 DB이므로 데이터 교환시에는 해당 API를 이용해서 호출해야한다. 이때 새로운 기능이 추가되어 애플리케이션에서 요구하는 데이터가 변경되었을 경우, 실시간 DB를 변경하기는 매우 어렵다. 메모리 구조로 되어 있어 변경하기도 어렵지만, DB뿐만 아니라 API도 변경할 경우가 발생하는데 이러한 경우에는 기존의 동일한 API를 사용하던 다른 애플리케이션에도 문제를 일으킬 수 있게 된다.

DB 구조의 변경문제는 이력DB의 경우에도 마찬가지이다. DB 구조가 변경되면 다른 애플리케이션도 모두 변경해야하는 문제가 발생하여 유지보수 비용이 매우 증가한다. 그래서 애플리케이션과 DB 사이에 중개기능을 하는 프로세스를 두는 방식이 있는데 이를 미들웨어라 한다. 이러한 방식은 서버와 클라이언트 사이에 중개 애플리케이션을 두어서, 각 프로그램에서 별도로 처리하던 작업을 이곳에서 처리하도록 한 방식이다. 그럼으로써 각 애플리케이션마다 기능을 구현해야할 것을 한번만 구현하면 되고 추가 애플리케이션에도 이 기능을 활용하면 되므로 애플리케이션 개발기간을 줄일 수 있다. 또한 DB가 변경되더라도 미들웨어만 수정하면 되므로, 애플리케이션은 수정 없이 그대로 사용할 수 있게 된다. 이 때, 메시지기반 통신이라는 것은 미들웨어와 애플리케이션 간에 통신방식을 플랫폼에 독립적인 XML 메시지를 이용하여 통신하는 것을 뜻한다.

2.2 GID 표준인터페이스를 이용한 메시지 교환 방식

IEC 61970에서는 애플리케이션에서 공통정보모델 데이터를 효과적으로 교환하기 위한 표준 인터페이스를 정의하고 있다. 이것은 애플리케이션의 종류에 상관없이, 심지어 데이터모델에 상관없이 데이터를 교환할 수 있다. 아래 그림은 GID를 적용한 애플리케이션 통합방법을 나타낸 것이다.



〈그림 2〉 GID 기반 메시지 통합

GID는 공통정보모델 메시지 통합버스와 애플리케이션 간 CIM/XML 메시지를 주고받기위한 통신 인터페이스로 각 용도에 따라 GDA, GES, HSDA, TSDA로 구분되어 있는 인터페이스 모듈이다. GID 모듈은 다음과 같이 4개로 구성되어 있다.

- GDA (Generic Data Access)
구조적 데이터에 랜덤하게 조회 및 질의할 수 있는 일반적인 Request/Reply 기반 인터페이스 기능
- GES (Generic Eventing and Subscription)
스키마(class)와 인스턴스 정보를 계층적으로 조회할 수 있는 Publish/Subscribe 기반 인터페이스 기능
- HSDA (High Speed Data Access)
고속 데이터에 대해 스키마와 인스턴스 정보를 계층적으로 조회하고 질의할 수 있는 Request/Reply와 Publish/Subscribe 기반 인터페이스 기능
- TSDA (Time Series Data Access)
시간에 따라 누적되는 데이터에 대해 스키마와 인스턴스 정보를 계층적으로 조회하고 질의할 수 있는 Request/Reply와 Publish/Subscribe 기반 인터페이스 기능

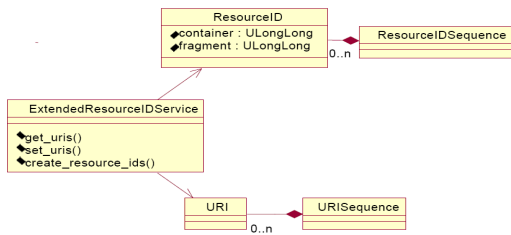
GID는 OMG의 DAF, DAIS를 기반으로 하고 있으며, 전력계통을 위한 인터페이스가 몇 가지 추가되어 있다. GID의 Common Service와 GDA는 DAF를 이용하여 구현되었고, 고속 데이터 전송을 위한 HSDA와 이벤트 전달을 위한 GES는 DAIS를 이용하여 구현되었다.

2.2.1 Common Service

데이터 통신을 위해서는 데이터 모델을 전달하기 위한 기반요소가 필요하다. 그 기반요소로는 각 데이터 요소를 구분하거나 의미를 갖는 하나의 묶음으로 전달하는 기능 등이 해당한다. 이러한 기반 역할을 하는 것이 Common Service이다. IEC 61970 GID에서는 Identifier, Description, View등의 세가지 인터페이스를 제공한다.

○ Resource Identifier

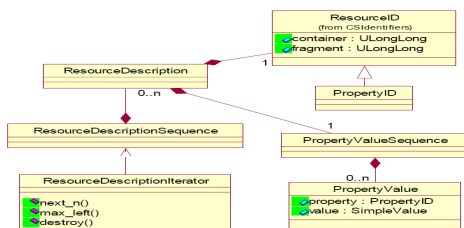
DAF에서는 각 리소스를 URI를 이용하여 구분한다. URI의 구조는 <scheme-name>:'<opaque-part> '#'<fragment-identifier>와 같이 구성되어 있다. opaque-part는 URI의 메인으로써 도메인 명이나 pathname을 포함한다. 예를 들어, 변압기에 대한 클래스에 대해서는 "http://utility.com/Planning/Production/CIM-schemacimu09a#Transformer"와 같이 표시한다. 여기서 주의할 것은 실제로 utility.com이라는 사이트가 존재해야 하는 것이 아니라, 단지 영역을 구분하기 위해 사용될 뿐이라는 것이다. #을 기준으로 컨테이너와 프레그먼트를 나타낸다. 아래 그림은 GDA에서 확인된 Identifier Model을 나타낸다. 그림에서 알 수 있듯이 이 서비스는 URI를 가져오거나 설정하는 함수를 제공한다.



<그림 3> Identifier Model

○ Description

이 모듈은 클래스나 클래스의 속성, 객체 인스턴스를 전달하기 위한 수단을 제공한다. 즉, 하나의 클래스에 대한 여러 속성을 한꺼번에 전달할 수 있는 기본 단위를 제공한다. Description 서비스는 아래 그림과 같은 클래스 다이어그램을 가지고 있다.

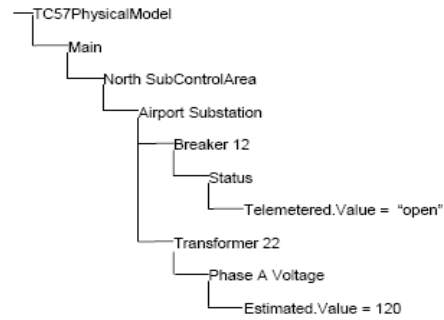


<그림 4> ResourceDescription

DAF에서 모든 value는 SimpleValue로 표시할 수 있는데, 이것은 모든 기본 데이터 타입을 포함하고 있다. 그리고 클라이언트에서 서버에 질의시에 반환 값인 ResourceDescriptionSequence는 ResourceDescription의 배열이다. 하지만 만약 질의에 대한 결과 값이 대량일 경우 이 데이터를 한번에 전달하는 것은 클라이언트에 부담을 줄 수 있고, 만약 클라이언트 측에서도 데이터를 전부 사용하는 것이 아니라 일부만 사용할 경우에는 비효율성이 발생한다. DAF에서는 효율성을 위해 ResourceDescriptionSequence을 반환하는 대신 ResourceDescriptionIterator라는 Iterator를 반환한다. 이것은 데이터 자체는 서버에 놔두고 포인터만을 전달하는 방식이다.

○ View

공통정보모델 데이터는 요소간에 상속관계를 가지고 있다. 특히 전력계통 구성자체가 상속구조로 되어 있기 때문에 데이터를 표시하는데 있어서도 트리방식을 사용하면 효율적으로 사용자에게 보여줄 수 있다. 아래 그림은 물리적인 계층구조에 따라 구성요소를 표시한 TC57PhycalModel View의 예이다.



<그림 5> TC57PhycalModel의 예

3. 결 론

현재의 SCADA 애플리케이션의 정보교환 방식은 취득 포인트 매핑에 의한 방법이 주를 이루었지만 IEC 61970을 적용한 SCADA 시스템은 계통구성에서부터 취득정보까지 모두 공통정보모델 데이터를 기반으로 하고 있다. IEC 61970에서는 또한 애플리케이션에서 공통정보모델 데이터를 효과적으로 교환하기 위한 표준 인터페이스를 정의하고 있다. 이것은 애플리케이션의 종류에 상관없이, 심지어 데이터모델에 상관없이 데이터를 교환할 수 있다. 본 논문은 공통정보모델 기반의 GID 표준 인터페이스를 적용한 효과적인 SCADA 메시지 교환에 관해 기술하였다.

[참 고 문 헌]

- [1] Alan W.McMorran, "An Introduction to IEC61970-301 & 61968-11: The Common Information Model", University of Strathclyde, 2007
- [2] "Data Acquisition from Industrial Systems Specification", OMG, 2005
- [3] "CIM Data Views for Bulk Power Systems", EPRI, 1999