

## 서버클라이언트 구조의 전력통신서비스 UIB 설계

박정진, 김건중, 황인준, 양민욱, 몽흐푸랩, 신만철, 오성균  
충남대학교 전기공학과

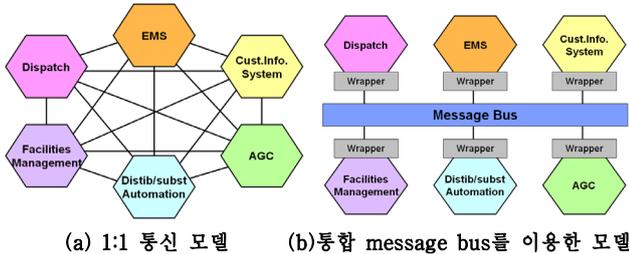
### The Architecture of Power system Communication Service UIB based Server Client

JeongJin Park, GunJoong Kim, InJun Hwang, MinOok Yang, Mungkhpurev, mancheol Shin, Sungkyun Oh  
Chungnam National University

**Abstract** - Legacy utilities can not adapt to changing business conditions without having complete control of operational systems and a flexible infrastructure that supports change. What is needed is a framework that enables data to be coalesced and transformed in to usable knowledge. This framework should be robust enough to that as new situations arise, information can be shared in ways not previously anticipated. Utility Integration Bus (UIB) provides an integration framework which solves many of these issues while controlling complexity. This paper provides UIB's overview and course.

#### 1. 서 론

전력 수요의 증가와 도시의 거대화 등 경제, 사회적인 발달에 따라 전력 시스템의 관리의 중요성과 편리성, 안정성이 대두되고 있다. 오늘날의 Supervisory Control And Data Acquisition(SCADA)이나 Energy Management System(EMS), Distribution Facilities Management System(DFMS), Automatic Meter Reading(AMR) 등과 같은 전력시스템들은 하위 시스템들의 제어와 감시가 실시간으로 이루어져야 한다. 서로 연계된 시스템들을 보다 효율적으로 관리하고 통제하기 위해서는 각각의 시스템 간 연계된 통신서비스가 필요하다. 그림 1은 각 시스템 간 통신서비스를 나타낸다.



<그림 1> 시스템간의 통신 모델

그림 1 (a)와 같이 통신서비스를 제공한다면 각 하위 시스템들의 수가 적을 때는 각 하위시스템 간에 1:1로 통신이 이루어져 복잡하지 않아 간단하고 심플하게 통신이 이루어져 구현과 유지보수가 편리할 수 있다. 하지만 하위 시스템의 숫자가 늘어나면 늘어날수록 통신 장비의 수와 비용이 더욱 더 늘어나게 된다. 그래서 등장하게 된 개념이 그림 1 (b)와 같은 메시지 버스를 이용한 미들웨어인 Utility Integration Bus(UIB)이다. UIB는 각 하위 시스템간의 통신을 가운데에서 중계한다. 본 논문에서는 이러한 메시지 버스를 이용하여 어플리케이션간의 통합에 관한 내용을 서술하며 그의 필요성과 이점에 대해 언급하고자 한다.

#### 2. 본 론

##### 2.1 UIB의 이점

시스템통신의 통합으로 그 시스템 안에서는 다음과 같은 이점이 있다. 우선 데이터 교환에 있어 그 양을 최소화 시킬 수 있고 각 시스템 간 통신을 위하여 서로 다른 그리고 많은 접속들을 하나의 통합버스로 대체할 수 있게 된다. 그리고 각 어플리케이션의 데이터를 통합이 가능해지고 다른 시스템에 의해 종속적으로 이루어지는 시스템을 제거하고 독립적인 시스템구축이 가능하다. 또한 UIB는 각 시스템간의 연락이 가능하고 메시지 교환이 가능하여 데이터전송 뿐만아니라 긴급한 연락체계나 서로간의 정보 교환, 공유 등이 매우 유기적으로 연관되어 있어 시스템을 가동, 운영하는

데 관리자의 입장에서 매우 편리함을 제공한다.

##### 2.2 데이터 교환에 필요한 메카니즘

UIB는 그림 1의 통신모델을 그림 2의 모델로 변환하는 과정에서 각 하위시스템사이에서 중계자 역할을 하는 하나의 매개체이다. 이렇게 중간에서 서로간의 통신을 연결함에 있어 보다 편리하고 안정감 있게 그리고 빠르게 하기위한 기능들이 요구된다.

##### 2.2.1 Wrapper

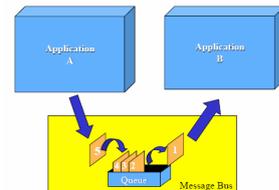
그림 1 (b)의 통신서비스는 통합버스를 구현해야 한다는 점에서 초기에 약간의 투자비용이 들겠지만 하위 시스템이 많아지면 많아질수록 그 비용은 현저히 감소된다. 그래서 여기에 추가되는 개념이 Wrapper이다. 통합버스의 구현이 완료된 후 추가되는 하위 시스템은 구현된 통합버스에 맞게 모델링을 시켜 통합버스에 접속에 아무런 지장이 없지만 기존에 사용 중인 하부시스템은 구현된 통합버스와 코드가 맞지 않으므로 직접 연결하여 사용할 수 없다. 그래서 기존의 하부시스템과 통합버스 사이에서 서로의 코드를 변환해주는 일종의 아답터 역할을 하는 것이다.

##### 2.2.2 Metadata

UIB에서는 각 데이터들의 모델이 여러 어플리케이션들에 의해 사용되어야 하므로 이러한 점에 사용과 관리의 편의를 위해 메타데이터를 사용한다. 메타데이터는 속성정보라고도 하는데 데이터베이스에 저장되어 있는 데이터들에 관한 정보를 가지고 있다. 저장된 데이터는 데이터가 어디에 있는지, 누가 작성하고 관리하는지, 어떤 어플리케이션에서 처리하는지, 다른 데이터와 어떠한 관계에 있는지, 그리고 어떻게 변화될 수 있는지에 관한 정보를 가지고 있다. 이러한 기능은 이전에는 사용자들이 쉽게 이용하지 못했던 것들이나 이용이 불가능했던 것들을 가능하게 하고 데이터 베이스에 여러개의 접속 대신에 하나의 데이터 접속 통로를 제공하게 된다. 또한 메타데이터의 제어를 위한 중앙집중방식은 정보의 일관성과 정확성을 확실하게 한다.

##### 2.2.3 Message Broker

메시지 브로커는 어플리케이션 간의 메시지를 통해 서로간의 강한 결합을 가능하게 한다. 그림 2에서처럼 메시지 브로커 기술로 어플리케이션은 각각의 메시지를 통합버스 안의 메시지 큐로 보낸다. 그러면 메시지 큐는 미리 정의되는 기준에 의거하여 상대방인 다른 내부의 어플리케이션이나 외부의 어플리케이션에 메시지를 전송한다. 또한 자료 교환이나 일의 흐름에 자동적으로 적용되는 시스템이 가능하게 된다. 그리하여 다수의 어플리케이션을 모두 포함하여 정보의 흐름을 하나의 제어점에서 관리할 수 있게 한다. 이는 UIB의 핵심기술로 데이터 처리의 집중화를 가능하게 해주고 데이터를 처리하는데 있어 그 지속적인 흐름을 제어하는 추가되는 도구들을 제공해준다.



<그림 2> 메시지 버스안에서 메시지큐를 통한 데이터 교환 모델

##### 2.2.4 High Speed Data Access(HSDA)

HSDA는 대용량의 데이터가 실시간으로 교환이 요구되는 특별한 경우에 적합한 데이터 교환 방식이다. 예를 들어 SCADA에서 요구되는 실시간적인 상태변화나 결과 등을 고속 그리고 대용량의 정보처리의 작업에 사용된다.

### 2.2.5 Time Series Data Access(TSDA)

TSDA는 순차적으로 데이터에 대한 접근을 나타내는 것으로 각 클라이언트에서 들어오는 데이터들에 대한 중요도를 선별하여 시간 순으로 데이터를 교환할 수 있게 한다. 예를 들어 온도측정과 같이 각 시간별 데이터 체크가 필요한 분야에서 사용된다.

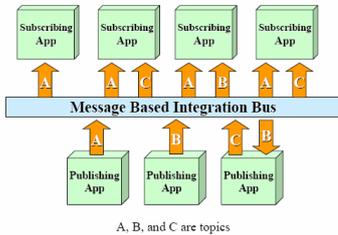
### 2.2.6 Generic Data Access(GDA)

GDA는 데이터베이스 안에서 자료들에 관한 읽기와 쓰기를 할 수 있는 권한을 주고 데이터가 서버에서 업데이트 되었을 때 클라이언트들에게 알리는 기능을 한다. 예를 들어 서버에서 데이터가 업데이트가 되었을 때 통합버스는 GDA를 통해 각각의 데이터베이스와 Asset 모델쪽 서버에 업데이트사실이 동기화 되게끔 한다.

### 2.2.7 Generic Eventing and Subscriptions(GES)

GES는 메시지 기반 통합버스 그림 3의 Publish와 Subscribe의 방법으로 확장을 하는 방법으로 메시지의 소스는 각 주제(A,B,C)에 따라 분리 전송하고 각 메시지를 받는 클라이언트들은 그들이 요구한 주제에 맞게 메시지를 전송받는다. 메시지 기반의 통합버스를 매개로 사용함으로써 Publish와 Subscribe는 다음과 같이 데이터 소스로부터 어플리케이션을 분리하고 범위를 정한다.

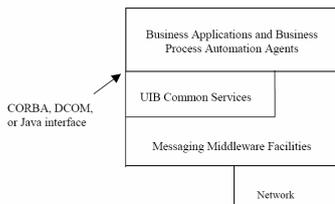
- Publishers(또는 Subscribers)는 Subscribers(또는 Publishers)에 관한 정보가 필요 없다.
- 다수의 Subscribers는 Publishers의 구성에 관한 정보없이 데이터 수신 가능하다



<그림 3> Publish / Subscribe Mechanism

### 2.3 UIB Architecture

아래의 그림 4는 어플리케이션을 두 개의 인터페이스 구조로 나타낸다. 첫 번째 인터페이스 구조는 어플리케이션이 하나 이상의 UIB서비스를 어떤식으로 호출하는지를 정의하고 두 번째 인터페이스는 미들웨어에서 데이터를 제공받게 된다. 이러한 미들웨어는 메시징 서비스와 작업 처리 서비스의 두 가지로 구성되어진다. 이러한 어플리케이션은 각 통신 서비스를 통해 UIB 서비스를 사용할 수 있다.



<그림 4> UIB Architecture

### 2.4 외국의 경우

외국의 경우, 특히 발·송전분야에서의 이러한 각 어플리케이션간의 통합버스 운영은 벌써 진행되어 현재 운영중에 있다. 각 분야별로 운영되고 있는 실태를 보면 다음과 같다.

#### 2.4.1 Florida Power and Light(FPL), Integration Bus

FPL은 플로리다주 마이애미의 새로운 계통통제 센터로 각 시스템 간 정보교환을 효율적으로 하기위하여 계통통제센터, 고객정보, 발전단 통신관리, 기술지원, 자산관리, 계획, 응급복구, 인력관리 등의 다양한 데이터 시스템 간 통신을 구현하고자 하였다. 그리하여 FPL에서는 각 어플리케이션 중간에 연결점 중간에 미들웨어를 두는 방안을 도입하게 되었다. 이곳의 통합버스는 이벤트, 함수, 속성 등의 어플리케이션 데이터 교환을 위해 컴포넌트 인터페이스를 이용하고 새롭게 도입되는 어플리케이션은 기존의 컴포넌트 인터페이스에 호환되도록 제작되어진다.

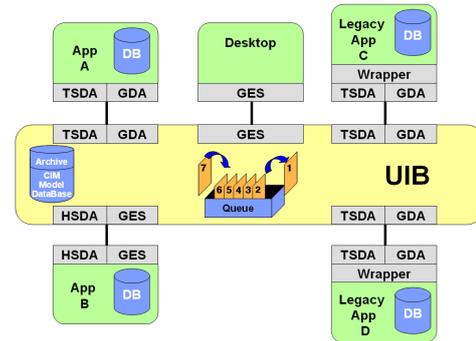
#### 2.4.2 Southern California Edison(SCE) Bus Project

SCE사 기존의 송배전망은 1:1통신방식의 저미줄같은 인프라 구조였다. 그러다보니 각 통신망들이 너무 복잡해져서 문제해결을 위해 통합버스 프로젝트를 시작하게 되었다. 이 프로젝트의 최종목표는 CIM을 기반

으로 하는 미들웨어를 사용하여 송배전의 SCADA, OMS, 배전망 스위치를 실시간으로 통합, 이용하는 것이다. 그 결과 데이터의 동기와 로깅이 자동으로 이루어졌고 송배전 운영시스템은 실시간으로 통신이 가능하게 되었다. 이 시스템은 표준화 인터페이스를 통한 실계통 실시간 운영정보와 리포트 시스템의 중복된 수동 데이터 항목의 제거와 처리비용을 감소시켰다.

### 2.5 한국형 UIB의 설계

UIB는 각 하위 시스템의 Utility나 Application들의 통합을 이루어 내어 현재 우리나라의 환경에 가장 적절하게 만들어야 한다. 이것의 효율성은 현재 송배전이나 변전의 EMS에 적용될 수도 있고 SCADA망에 적용시킬 수도 있다. 현재의 산개되어있는 각각의 시스템을 통합하고 시스템의 관리나 유지를 보다 쉽게 하기 위해서는 현재 상태의 유지가 아니라 보다 진보된 방향으로의 전환이다. 변화를 두려워 말고 현재보다 앞선 선진기술을 받아들여 산업적으로나 경제적으로 보다 효율을 따져 시스템 구축을 한다면 현재의 복잡한 전문기술자가 있어야 할 수 있는 작업들을 엔지니어들이 쉽고 그 시스템에 대한 내용들을 장악할 수 있도록 하는 것이 사교나 기타 위급상황이 발생했을 경우 보다 빠른 조치나 대책이 뒤따를 수 있다.



<그림 6> 한국형 UIB 개략도

그림 6은 UIB의 개략도이다. UIB를 설계시 그것과 호환이 되도록 만든 것이 어플리케이션 A와 B이다. 그리고 기존의 어플리케이션은 C와 D이다. 기존의 어플리케이션은 새로 구현된 UIB와 호환이 어려우므로 기존의 어플리케이션을 감싸는 Wrapper를 써서 UIB에 연결을 시키고 시스템을 구성한다. 각 하위 시스템의 데이터베이스안에 있는 데이터들을 UIB안의 Archive안에 메타데이터로써 구성하고 정보를 저장한다. 그리고 각 어플리케이션 간의 메시징은 UIB의 메시지 큐를 통해 전송이 이루어지게 한다. 어플리케이션 A의 데이터를 B와 C에서 요구한다면 그림 3과 같이 A의 데이터 하나의 전송으로 B와 C 두곳에서 동시에 자료를 전송받을 수 있게끔 한다. 또한 각각의 TSDA나 GES 등과 같은 것들은 어플리케이션과 UIB 두곳다 존재하게 된다. 어플리케이션이 정보를 요청하게 되면 그 어플리케이션에 존재하는 것은 클라이언트가 되고 UIB는 서버가 된다. 그러면 요청받은 곳의 어플리케이션은 서버가 되고 다시 그곳과 연결되는 UIB는 클라이언트가 된다. 이렇게 각 어플리케이션과 UIB는 서버 또는 클라이언트만 되는 것이 아니라 서버와 클라이언트 둘다의 기능을 가지고 있어야 제대로 된 UIB통신이 가능해진다.

### 3. 결 론

지금까지 우리는 UIB에서 필요한 메커니즘과 기반기술들에 대해 알아보았다. UIB를 이용하여 성공한 외국의 사례도 알아보았다. 유틸리티가 되는 어떤 시스템이 되는 현재의 상태에 머물러 안주한다면 미래는 현재의 연속이 될 것이다. 하지만 이러한 안정적인 흐름에 몸을 맡기기 보다는 새로운 방식의 문을 두드리는 것이 이제는 요구되는 시대이다. 게다가 역동적인 에너지시장과 더욱더 중요시되는 e-business에서 어플리케이션 통합의 틀의 분야를 더욱 넓혀가고 있다. 이러한 통합 실현의 키는 기업단에서 그의 정보를 얼마나 공유하느냐에 달려있다. 기존의 표준들과 제품 또는 시스템의 안정성과 경제성 모두를 생각해본다면 UIB는 꼭 필요한 도구가 될 것이다.

### [참 고 문 헌]

[1] SISCO, "UIB White Paper", 2001  
 [2] IEC, "IEC 61970", TS 61970-401, Oct 2005  
 [3] 전력연구원, "국내 전력계통 특성을 고려한 특성정보체계 도출 및 입력프로그램 개발", Dec 2005  
 [4] 김진중, 신만철, 최장흠, 박현경, "전력 시스템의 XML 웹 서비스", 전기학회논문지A,B,C,D, pp.12-14, 2003.07, 대한전기학회