

Web Service를 이용한 배전계통 애플리케이션의 통합

정남준^o 조선구 유인협 김선익 고종민
한전 전력연구원

njjung@kepri.re.kr, csk9306@kepri.re.k, ihyu@kepri.re.kr, sikim@kepri.re.kr, kojim@kepri.re.kr

Integration of Distribution System Applications Using Web Service

Nam-Joon Jung^o, Seon-Ku Cho*, In-Hyeob Yu*, Jong-Min Ko*, Sun-Ik Kim*
Korea Electric Power Research Institute

요 약

배전계통은 전력계통에서 전력을 소비자에게 공급하는 전반적인 분야를 말한다. 이와 관련된 애플리케이션으로는 고객정보관리시스템(CIS), 정전관리프로그램(OMS), 계량정보시스템(AMR), 작업관리프로그램(WM) 등이 있다. 이들은 사용자정보와 전력사용에 관한 정보를 활용하여 배전 시스템의 안정적 운영에 필요한 기능을 수행하고 있다. 하지만 현재 운영 중인 시스템들은 각 시스템이나 애플리케이션마다 각각 사용자정보와 전력사용 이력정보를 관리하고 있어서 데이터가 중복되어 관리되고 있거나, 서로 정보를 교환하는데 제약을 가지기도 한다. 본 논문에서는 이와 같은 배전계통 애플리케이션에 SOA(Service Oriented Architecture)를 적용하여 시스템간 정보의 흐름이 원활하게 이루어 질 수 있도록 애플리케이션들을 통합하는 방법에 대해 기술한다.

1. 서 론

컴퓨터를 이용한 애플리케이션 개발이 보편화되고 네트워크의 급속한 발전에 따라 일련의 작업을 위한 애플리케이션 간 직접적인 통신 요구가 증가되었다. 하지만 애플리케이션마다 구현된 시스템 플랫폼이 다르고 운영되는 통신 프로토콜이 다르기 때문에 서로 정보교환에 어려움이 있었다이를 극복하기 위한 노력의 하나로 탄생한 것이 웹 서비스(Web Service)이다. 웹 서비스는 데이터를 표현하는데 있어서 XML문서를 이용하고 이 XML문서를 SOAP(Simple Object Application Protocol)을 사용하여 전달한다 XML문서를 사용함으로써 구현 언어에 상관없이 데이터 정보를 표현하고 쉽게 인식 가능하며, SOAP이 HTTP(Hyper Text Transport Protocol) 기반의 프로토콜이므로 방화벽을 통과할 수 있는 장점 또한 가지고 있다 이러한 여러 장점으로 인하여 웹 서비스는 최근 소프트웨어 산업에서 각광을 받고 있다 [1][2][3].

배전계통은 전력계통에서 전력을 사용자에게 공급하는 분야로서, 배전계통 애플리케이션에서는 그 기능에 따라 전력 설비 정보(정전, 휴전 등)와 사용자정보(계약정보 등) 그리고 사용자의 전력 사용 정보가 취급 관리된다. 배전계통 애플리케이션은 그 주요 기능에 따라 크게 고객정보시스템(CIS: Customer Information System), 자동계량정보시스템(AMR

: Automated Meter Reading), 정전관리 시스템(OMS : Outage Management System), 작업관리 시스템(WM : Work Management System)으로 구분할 수 있다.

현재 운영중인 배전계통 애플리케이션들은 각 애플리케이션마다 독립된 데이터베이스를 이용하여 단독으로 동작하고 있다. 하지만 최근 업무 효율성의 향상과 고객의 새로운 부가서비스에 대한 요구로 애플리케이션간의 통합이 필요하게 되었다. 특히 최근 IEC TC57에서 전력계통에 대한 공통정보모델인 CIM(Common Information Model)이 제정됨에 따라 애플리케이션간 통합을 할 수 있는 가능성이 증가되었다[4][5][6].

본 논문에서는 전력계통에 대한 공통정보모델로서 각광을 받고 있는 CIM에 대해 간단히 소개하고 이를 활용한 XML 문서의 작성, 그리고 Web Service를 이용하여 애플리케이션 간 데이터를 교환하는 방법에 대해 기술한다

2. 기존 배전계통 애플리케이션의 구성

앞에서 언급한 바와 같이 배전계통 애플리케이션은 정전 관리시스템(OMS), 고객정보시스템(CIS), 계량정보 관리(AMR), 작업관리(WM), 계통운영(NO) 애플리케이션 등으로 구성되어 있다. 더구나 이들 애플리케이션은 각기 다른 벤더에 의해 개발되었기 때문에 별도의 데이터모델을 사용한 데이터베이스를 사용하고 있다 따라서 어떤 애플리케이션

이 다른 애플리케이션에서 관리하는 데이터베이스에 접근하기 위해서는 해당 데이터베이스의 데이터모델에 대한 분석과 이를 위한 인터페이스를 별도로 제작하여야 했다는 통합을 위한 기간과 비용을 증가시키는 원인이 된다또한 이미 개발된 인터페이스를 통해 다른 통합이 이루어졌더라도, 만약 데이터구조가 수정되었을 경우 이와 연관된 다른 애플리케이션의 인터페이스까지도 수정해야 하는 상황이 초래된다. 예를 들어, CIS의 사용자정보 데이터베이스가 수정되었을 경우, 이와 연관된OMS, AMR, WM, NO의 인터페이스까지도 수정해야 한다 그림 1은 현재 배전계통 애플리케이션의 구성을 나타낸 것이다 각 애플리케이션에서 측정 정보데이터에 접근하기 위해 별도의 인터페이스를 사용하는 것을 알 수 있다.

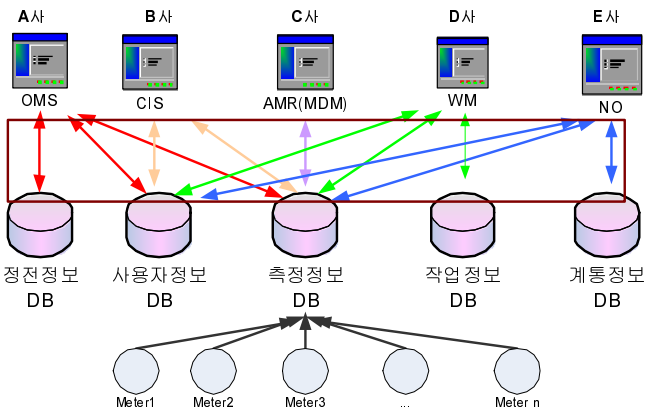


그림 1. 이전의 배전계통 애플리케이션 구성

하지만 이들 애플리케이션간에 공통된 데이터모델을 사용하면, 통합비용을 훨씬 줄일 수 있게 된다 즉 정전정보, 사용자정보, 측정정보, 작업정보, 계통정보에 대해서 데이터구조를 통일한다면 통합을 위한 데이터분석 및 별도의 인터페이스 개발이 생략된다. 일반적으로 통합과정에서 데이터분석이 차지하는 비중이 40% 이상이 된다는 점은 주지할 내용이다. 배전계통 애플리케이션의 통합에 있어서 표준화된 데이터모델을 사용하여XML기반의 메시지를 이용한다면 아래 그림과 같은 애플리케이션 통합이 가능해질 것이다

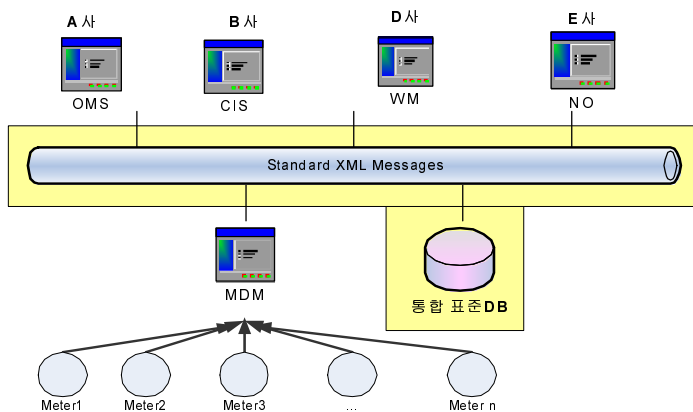


그림 2. 공통정보모델을 활용한 애플리케이션 통합

3. CIM

CIM(Common Information Model)은 IEC TC57에서 제정된 전력계통 구성요소에 대한 데이터모델이다전력계통과 관련된 애플리케이션간에 정보를 교환하기 위해 사용되는 데이터의 모델과 그 연관관계를 상세하게UML을 사용하여 정의하고 있다. 전력계통과 관련된 애플리케이션에 사용되는 모든 요소에 대해 모델링이 가능한 것은 전력계통의 특성상 그 구성요소가 제한되어 있고 그 구성요소의 속성이 일정하기 때문이다 아래 그림 3은 CIM을 구성하는 패키지 와 연관관계를 나타낸 것이다 이 중 배전계통에 해당하는 패키지는 Assets, Documentation, Consumer, Core2, ERP Support, OAG Messages등이고, 나머지 패키지들과도 연관 관계를 갖는다[4].

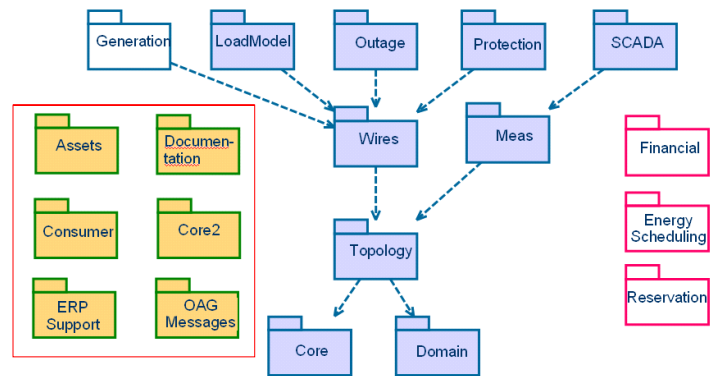


그림 3. CIM 패키지 구성도

이들은 UML의 클래스 다이어그램을 통해 기술되어 있으며, 각 클래스간의 연관관계도 포함되어 있다

또한 IEC 워킹그룹에서는 데이터를 전달하기 위한XML 메시지의 구조를 표준으로 정의하고 있다이것은 표준 메시지 형식을 사용함으로써 애플리케이션간 호환성을 증대시키기 위한 것으로, 메시지의 구조는 그림 4와 같이 크게 MessageHeader와 MessagePayload 두 부분으로 나누어지는데 MessageHeader는 메시지의 이름과 동작을 나타내고 MessagePayload는 실제 메시지 전달하고자 하는 내용이 포함된다[2].

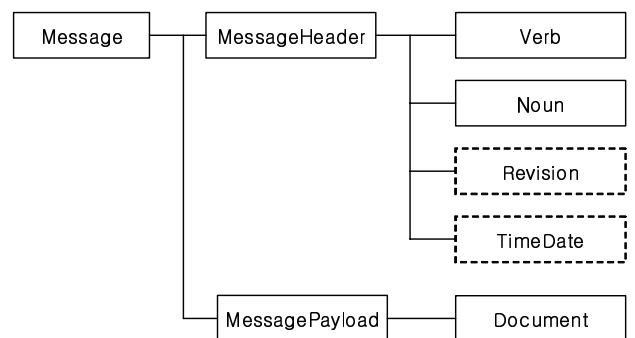


그림 4. XML 메시지 구조 표준

위 그림에서 실선으로 된 부분은 필수 요소이고점선으로

표기된 부분은 선택적인 사항을 나타낸다 즉, MessageHeader, Verb, Noun, MessagePayload, Document 는 모든 메시지가 반드시 포함하고 있어야 하는 항목임을 나타낸다. MessageHeader는 메시지의 동작을 나타내는Verb와 메시지의 명칭인 Noun이 포함하는데, Verb는 'CREATE', 'CHANGE', 'CANCEL', 'CLOSE', 'DELETE' 등과 같은 한정된 것만을 사용하도록 제한하고 있다

4. XML을 이용한 애플리케이션간 메시지 전달

IEC 표준에서는 각 배전 애플리케이션 별로 사용될 메시지의 종류와 구조를 정의하고 있다아래 그림은 AMR 애플리케이션 인터페이스를 위한CustomerDataSet 메시지의 구조를 나타낸 것이다.

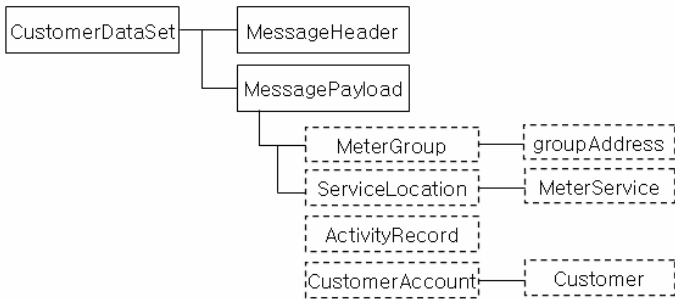


그림 5. CustomerDataSet 메시지 구조

위와 같이 권고된 메시지는XML메시지의 형식을 정의하는 XSD(XML Schema Definition)를 이용하여 정의되어 있다. 따라서 XSD에 정의된 형태로 메시지를 만들기 위해 메시지를 나타내는 클래스를 생성해야 한다 IEC 표준에 의한 배전계통 시스템의XML 메시지와 메시지 통신 프로토콜인 Web Service 간의 연관 관계를 나타내면 그림6과 같다.

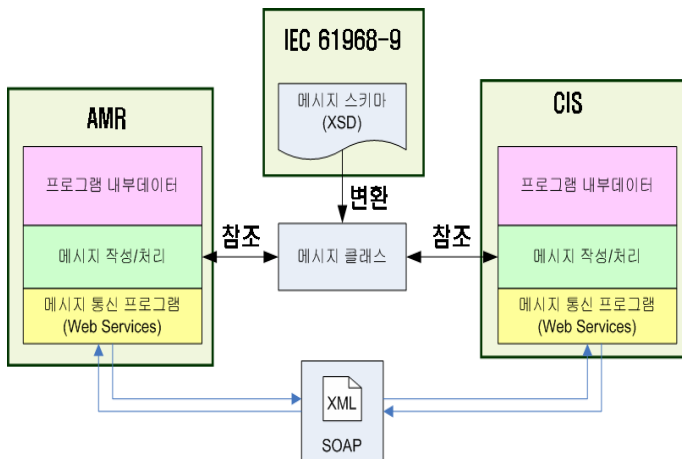


그림 6. IEC 61968 애플리케이션간 메시지 전달 구조

본 연구에서는 프로그래밍 개발 언어로C#을 이용하였으며, C#은 .NET Framework를 이용하여 XML 처리를 위한 다양한 라이브러리를 제공한다

그림 5의 CustomerDataSet 메시지 스키마를C#의 클래스로 변환하는 방법은 크게 두 가지가 있다. 첫째는 개발자가 직접 코딩으로 작성하는 방법이며 두 번째는 변환 툴을 이용한 방법이다. 첫째 방법은 시간이 많이 걸리고 실수에 의한 에러가 발생할 소지가 있다. 그래서 데이터의 구조를 표시 방법으로 XSD 파일의 사용이 많다 보니 이를 자동으로 변환해주는 툴이 제공되고 있다. 그 대표적인 툴로서는 Microsoft에서 제공하는 'Sample Code Generator' 프로그램 (XSObjectGen)을 들 수 있겠다[10]. 이 프로그램은 XSD 파일을 입력받아 동일한 자료구조를 갖는C#으로 제작된 클래스 파일로 변환시켜준다. CustomerDataSet 메시지 스키마는 'cimBase2.xsd', 'mdiMessage.xsd'등 다른 스키마 파일들을 참조하게 되는데, 참조되는 이 스키마 파일들에는 메시지에 사용되는CIM 데이터 타입과 표준 메시지 구조를 포함하고 있다. 'Sample Code Generator'를 이용하여C# 클래스 파일을 생성하는 과정은 다음 그림과 같다 여기에서 cimBase2.xsd는 메시지를 구성하는 기본형에 대한 데이터타입을 포함한 스키마이고 mdiMessage.xsd는 그림 4에서 나타난 표준 메시지 구조를 나타내는 스키마 파일이며 CustomerDataSet.xsd는 그림 5의 메시지구조를 포함한 파일이다. CustomerDataSet.xsd파일에서 다른 두 개의 스키마를 참조한다. 클래스 자동생성기'XSObjectGen'는 이 세 개의 XSD 스키마 파일을 이용하여 하나의CustomerDataSet 클래스를 생성한다.

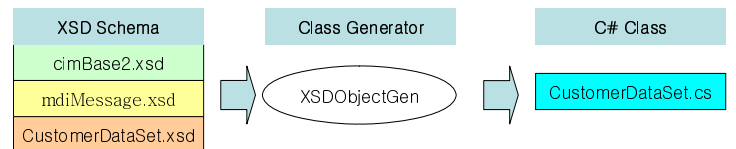


그림 7. 메시지 스키마 파일(xsd)을 클래스로 변환

위와 같은 과정을 통해 자동으로 생성된 메시지 클래스를 이용하여 개발자는 클래스의 멤버데이터 값을 설정하는 것만으로 메시지를 작성할 수 있게 된다 작성된 메시지는 SOAP 프로토콜을 통해 상대 애플리케이션에 전달된다 상대 애플리케이션에서도 동일한 메시지 스키마를 사용하므로 별도의 데이터 변환 등의 작업이 필요하지 않다

5. Web Service를 통한애플리케이션 통합

이제 애플리케이션을 Web Service를 이용하여 통합하는 사례를 들어보자. 애플리케이션 통합 예제로는 AMR(Automated Meter Reading)에서 CIS(Customer Information System)에 고객정보를 요청하는 사례를 들어보겠다. 그전에 AMR과 CIS에 대해 설명하면, AMR은 각 수용가의 전력사용량 계측을 계량기에 요청하여 사용량을 측정하여 MDM에 저장하고, 만일 수용가에 정전 등이 발생하면

OMS에 해당 정보를 전달하는 기능을 가지고 있다 그리고 CIS는 각 고객의 계약정보와 고객정보를 가지고 있으며 전력사용량에 따라 일정주기로 청구서를 발행하는 시스템이다. 아래 그림 8는 앞장에서 설명된 바와 같은 방법으로 작성된 메시지 클래스를 이용하여 CIS에 고객정보를 요청하는 내용의 메시지를 설정하는 코드를 나타낸 것이다

```
private void GetCustomerDataSetToCIS()
{
    CustomerMeterDataSet customerMeterDataSet = new
    CustomerMeterDataSet();
    //Set Message Header
    customerMeterDataSet.MessageHeader.Noun = "CustomerDataSet"
    customerMeterDataSet.MessageHeader.verb =
    mdimsgMay2005.verb.get;

    //Set MessagePayload
    customerMeterDataSet.MessagePayload.CustomerMeterCollection[0].
    CustomerAccountCollection[0].CustomerCollection[0].pucNumber =
    strUserID;

    customerMeterDataSet.MessagePayload.CustomerMeterCollection[0].
    CustomerAccountCollection[0].CustomerCollection[0].pathName =
    receiverUri.ToString();

    SoapEnvelope _message = new SoapEnvelope();
    _message.SetBodyObject(customerMeterDataSet);

    // Create the URI address objects for send and receive
    Uri toCISUri = new Uri(_AMRSettingsForm.strCISsenderTCP);

    // Assign the addressing SOAP message headers
    _message.Context.Addressing.Action = new
    Action("http://www.dk100.co.kr/types/fromAMRtoCIS");
    _message.Context.Addressing.From = new From(receiverUri);
    _message.Context.Addressing.ReplyTo = new ReplyTo(receiverUri);
    soapSender.Destination = toCISUri;

    // Send the SOAP request message
    soapSender.Send(_message);
}
```

그림 8. 고객정보 요청을 위한 메시지 설정 코드

위 그림 8에서 메시지 클래스를 생성한 후 사용자번호를 설정하고 SOAP Envelop를 이용하여 CIS에 메시지를 전달하는 내용을 알 수 있다. 실제 메시지의 내용을 XML문서로 확인해보면 그림 9와 같다.

```
<?xml version="1.0" encoding="utf-8"?>
<CustomerMeterDataSet
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="CustomerMeterDataSet">
```

```
<MessageHeader>
  <Verb xmlns="mdimsgMay2005">get</Verb>
  <Noun xmlns="mdimsgMay2005">CustomerDataSet</Noun>
</MessageHeader>
<MessagePayload>
  <CustomerMeter>
    <CustomerAccount>
      <Customer>
        <pathName
xmlns="cimBase2">soap.tcp://192.168.0.10:1004/AMRReceiver</path
Name>
        <pucNumber xmlns="cimBase2">User1</pucNumber>
      </Customer>
    </CustomerAccount>
  </CustomerMeter>
</MessagePayload>
</CustomerMeterDataSet>
```

그림 9. CIS에 고객정보 요청XML 메시지

그림 9의 XML 메시지를 살펴보면 회신을 위한 AMR의 수신포스트가 PathName 엘리먼트에 기재되어 있고, 고객번호가 pucNumber 엘리먼트에 나타나 있는 것을 확인할 수 있다. 이 때 AMR은 CIS의 데이터모델이나 제조회사에 상관없이 표준 XML 메시지 인터페이스를 통해 통합이 가능한 것이다. 위 고객정보 요청 메시지를 받은 CIS에서는 그림 10과 같은 고객정보를 설정하여 AMR에 응답하게 된다. 이 때에도 CIS는 AMR의 내부 데이터모델에 상관없이 고객정보를 전달할 수 있다는 것이다.

```
private GetCustomerDataSet(CustomerMeterDataSet
_customerMeterDataSet)
{
    _customerMeterDataSet.MessageHeader.Noun =
    "CustomerMeterDataSet";
    _customerMeterDataSet.MessageHeader.verb =
    mdimsgMay2005.verb.reply;

    _customerMeterDataSet.MessagePayload.CustomerMeterCollection[0].
    CustomerAccountCollection[0].CustomerCollection[0].name =
    row["USER_NAME"] as string;

    _customerMeterDataSet.MessagePayload.CustomerMeterCollection[0].
    MeterGroupCollection[0].groupAddress = (string)row["METER_ID"]
    as string;

    obj.serviceType = row["CONTRACT_KIND"] as string;
    ...

    SoapEnvelope response = new SoapEnvelope();
    response.SetBodyObject(_customerMeterDataSet);

    // Send the SOAP request message
    soapSender.Send(response); }
}
```

그림 10. 고객정보 응답 코드

마찬가지로 위 내용을 XML문서로 확인해보면 아래 그림 11과 같다. 그림에서 요청된 고객번호와 함께 고객정보 및 계약정보가 설정되어 있는 것을 확인 할 수 있다

```
<?xml version="1.0" encoding="utf-8"?>
<CustomerMeterDataSet
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="CustomerMeterDataSet">
  <MessageHeader>
    <Verb xmlns="mdimsgMay2005">deleted</Verb>
    <Noun xmlns="mdimsgMay2005">Customer</Noun>
  </MessageHeader>
  <MessagePayload>
    <CustomerMeter>
      <MeterGroup>
        <groupAddress xmlns="cimBase2">2</groupAddress>
      </MeterGroup>
      <ServiceLocation>
        <MeterServicePosition />
      </ServiceLocation>
      <ActivityRecord>
        <status xmlns="cimBase2">NotCustomer</status>
        <statusDateTime
xmlns="cimBase2">2007-01-22T00:00:00</statusDateTime>
      </ActivityRecord>
      <ActivityRecord />
      <CustomerAccount>
        <Customer>
          <name xmlns="cimBase2">홍길동</name>
          <pathName
xmlns="cimBase2">soap.tcp://dkbaek79:1004/AMRReceiver</pathName>
          <pucNumber xmlns="cimBase2">User1</pucNumber>
        </Customer>
      </CustomerAccount>
    </CustomerMeter>
  </MessagePayload>
</CustomerMeterDataSet>
```

그림 11. 고객정보 응답XML메시지

앞에서 설명된 바와 같이 애플리케이션간에 데이터를 교환하는에 있어서 Web Service를 이용하였다. 그럼으로써 Web Service의 장점인 프로그램언어에 독립적인 데이터 교환을 달성할 수 있었다. 또한 데이터를 XML문서로 작성하는데 있어서 사용될 데이터의 형태나 구성항목 사용이름 등이 고정된 공통데이터모델을 사용하여 별도의 데이터 변환이 필요치 않았다

이전에도 애플리케이션 통합을 위해 CORBA, COM/DCOM등 다양한 구현기술이 소개되었지만 구현의 복잡성과 운영플랫폼에 종속되는 이유로 널리 보급되지 못했다. 하지만 Web Service는 구현이 비교적 간단하

고 운영플랫폼에 독립적이므로 애플리케이션 통합기술로 널리 받아들여질 것이다.

6. 결론

본 논문을 통해 우리는 Web Service를 이용하여 배전계통 애플리케이션간의 통합을 수행하였다 배전계통 애플리케이션은 그 특성상 공유해야 하는 데이터가 많고 서로 밀접하게 연관된 부분이 많음에도 불구하고 각 애플리케이션마다 개발시기 및 개발업체가 서로 달라서 통합하는데 많은 어려움이 있었다. 본 논문을 통해 애플리케이션의 내부데이터에 상관없이 국제표준인 CIM을 활용하여 정보를 표현하고 교환할 수 있는 기술을 소개하였다 향후에는 본 기술을 바탕으로 실제 운영중인 배전계통 애플리케이션에 적용해봄으로써 실무 활용능력을 개발하고자 한다

참고 문헌

- [1] 최장훈, 김건중, "XML DataSet DB를 연동한 조류계산용 XML Web Service의 개발", 대한전기학회 논문지 제 52권, 제 4호, pp. 571-576, 2003.10
- [2] A. Krowczyk, Z. Greenvoss, et al "Professional C# Web Services : Building .NET Web Services with ASP.NET and .NET Remoting" Wrox, 2001
- [3] 이명진, ".novel XML & XML Web Services, 2006
- [4] IEC TC57 WG 14, "61968 System Interfaces For Distribution Management Part 1 : Interface Architecture and General Requirements"
- [5] IEC TC57 WG 14, "61968 System Interfaces For Distribution Management Part 3 : Interface for network operations"
- [6] Scott Neumann, "Integration of Metering Systems", Autovation 2005. Sept. 18-21, California, US, 2005
- [7] CIM Users Group [Online]. Available: <http://www.cimusers.org/>
- [8] IEC TC57 WG 14, "61968 System Interfaces For Distribution Management Part 9 : Interface Standard for Meter Reading and Control" Draft [Online]. Available: <http://sharepoint.ucausersgroup.org/CIM/default.aspx>
- [9] J. Hasan, M. Duran, "Expert Service-Orientd Architecture in C# 2005", Apress, 2005
- [10] Sample Code Generator [Online]. Available: www.microsoft.com/download/