ICROS-SICE International Joint Conference 2009
August 18-21, 2009, Fukuoka International Congress Center, Japan

Implementation of Web Services Based on IEC 61400-25 for Wind Power Plants

Min-Jae Seo, Tae-o Kim, Hong-Hee Lee[1]

[1] Department of Electrical and Electronics Engineering, Ulsan University, Ulsan, Korea
(Tel : +82-52-259-1478; E-mail: seominjae@gmail.com, hhlee@mail.ulsan.ac.kr)

Abstract: The SOAP (Simple Object Access Protocol) based on web services is one of mapping to communication profiles, specified in part4 of IEC 61400-25. The SOAP is a lightweight protocol for exchange of information in a distributed environment and Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. This paper presents web services to monitor and control the wind power plants by using SOAP according to IEC 61400-25 protocol.

Keywords: IEC 61400-25, Web service, SOAP

1. INTRODUCTION

Wind power has steadily gained a bigger and more dominant position in the power generation industry through the years. Now wind power has continued the worldwide successful story as the most dynamically growing energy source again in the year 2008. Since 2005, the global wind installations become more than doubled. They have reached 121,188 MW after 59,024 MW in 2005, 74,151 MW in 2006, and 93,927 MW in 2007. The turnover of the wind sector worldwide reached 40 billion € in the year 2008. [1]

The international communication standard IEC 61400-25(Communications for monitoring and control of wind power plants) of the IEC TC 88 is developed in order to provide uniform information exchange for monitoring and controlling of wind power plants. It will eliminate the issue of proprietary communication systems utilizing a wide variety of protocols, labels, semantics, etc.. It enables components from different vendors to easily communicate with other components at any location and at any time. Object-oriented data structures make the engineering and the handling of huge amounts of information provided by wind power plants less time-consuming and more efficient. Scalability, connectivity, and interoperability can be maximized to reduce cost and needed manpower.

This paper focuses on IEC 61400-25 standard. That working group is working hard trying to provide a set of communication protocols able to fulfill the different requirements of the system and using the best set of available communication protocols. These protocols include the following mapping methods:

- SOAP-based web services
- OPC/XML-DA
- IEC 61850-8-1 MMS
- IEC 60870-5-104
- DNP3

Depending on the mapping used, the data interchanged in the network have a different format, but the information exchanged by the same services in all mappings methods contains the same meaning and content. [2]

This experiment investigates web services to monitor and control the wind power plants by using SOAP according to IEC 61400-25.

2. IEC 61400-25 STANDARD

The IEC 61400-25 standard is a basis for simplifying the roles that the wind turbine and SCADA systems have to play. The crucial part of the wind power plants information, information exchange methods, and communication stacks are standardized by IEC 61400-25.

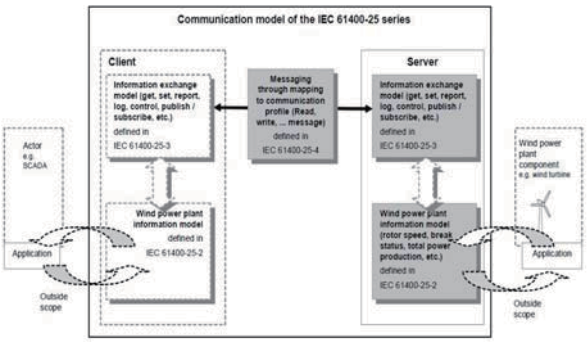

Fig. 1 Communication model IEC 61400-25.

Fig. 1 shows communication model IEC 61400-25. IEC 61400-25 defines all details required to connect the components of wind power plants in a multi-vendor environment and to exchange the information available by a component. This is done by definitions in the document or by reference to other commonly used standards, such as IEC 61850. IEC 61400-25 consists of six parts:

Part 25-1 Overall description of principles and models
Part 25-2 Information models
Part 25-3 Information exchange models

Part 25-4 Mapping to communication profiles
Part 25-5 Conformance testing
Part25-6 LN classes and Data classes for Condition Monitoring

2.1 Information model

The approach of the standard is to decompose the application functions into the smallest entities, which are used to exchange information. These entities are called logical node (LN). A LN consists of a gathering of related data defined as Data Classes (DC). All the information in a logical node is contained in the respective DC. The structures of all LN is similar and has a standardized form where different types of LN can be constructed through the combination of different optional DC.[3][4]

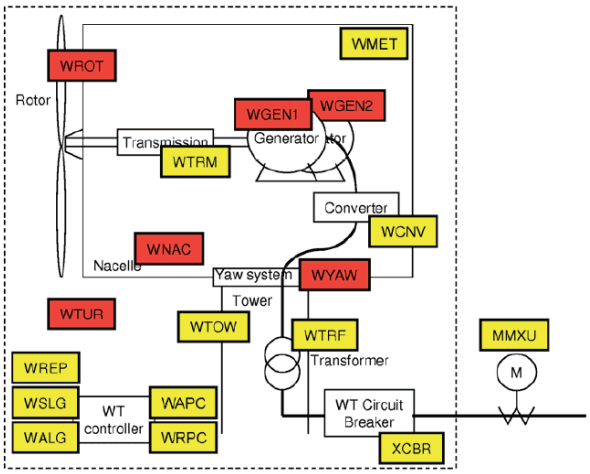

Fig. 2 Information Models & Logical node

All of the LN used in modeling the wind power plants inherit their structure from the abstract logical node class defined in IEC 61850-7-2. From the implementation point of view, these different LN are similar since the structure is based on the common definition and follows the common pattern. The hierarchical structure of a logical device (LD) residing in a server together with LN and DC is depicted in the Fig.2.

Table 1 Logical node class.

WGEN class			
Attribute name	**Attribute type**	**Explanation**	**M/O**
		LN shall inherit all Mandatory Data from Wind Power Plant Common Logical Node Class	M
Data			
Common Information			
OpTmRs	TMS	Generator operation time	O
Status Information			
GnOpMod	STV	Operation mode of generator	O
ClSt	STV	Status of generator cooling system	O
Analogue Information			
Spd	MV	Generator speed	M
W	WYE	Generator active power	O
VAr	WYE	Generator reactive power	O
GnTmpSta	MV	Temperature measurements for generator stator	O

2.2 Information exchange model

IEC 61400-25-2 explains the information exchange model which is implemented on the server enabling client's systems to access and modify data in the information model. Each information model instance has a service interface describing the operations available on that particular instance. Each information model object has a specific set of services making it possible to read or write. [5]

2.3 Mapping to web services

IEC 61400-25-4 presents the mapping of the information model and the information exchange model to a specific protocol stack. Five actual mappings are presented in IEC 61400-25-4 and the developers are free to choose the mapping they prefer. In this paper, we use the SOAP based web services mapping. In order to use the web services mapping, it is worth considering a suitable environment for mapping (implementing) the system. The Windows Communication Foundation (WCF) has been found ideal for this purpose. [5]

3. IMPREMENTATION OF WEB SERVICES

3.1 SOAP (Simple Object Access Protocol)

SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. Fig. 3 and Fig. 4 show communication procedure and structure of SOAP, respectively.

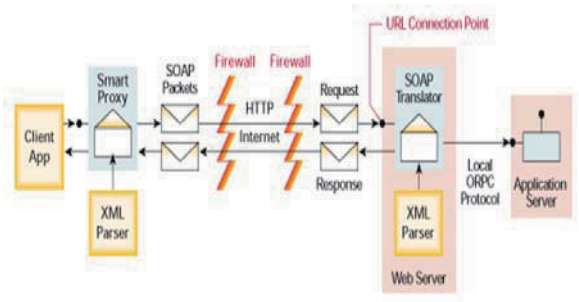

Fig. 3 Communication procedure of SOAP.

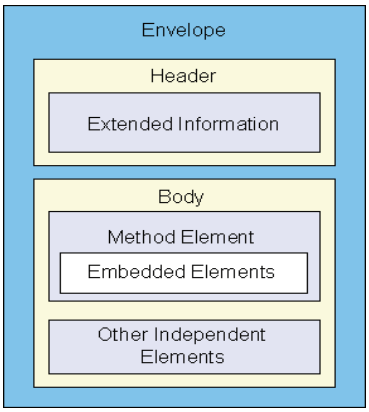

Fig. 4 Structure of SOAP.

It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses.[6]

3.2 Web services

Web services provide a standard means of interoperating between different software applications running on a variety of platforms and/or frameworks. Web services system supports models of request and response. Web services & WSDL are shown in Fig. 5.

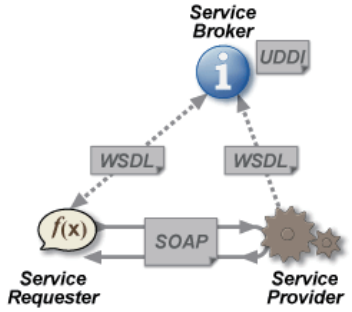

Fig. 5 Web services & WSDL.

The Web Services Description Language (WSDL) is an XML-based language used to describe the services a business offers and to provide a way for individuals and other businesses to access those services electronically. WSDL is the cornerstone of the Universal Description, Discovery, and Integration (UDDI) initiative spearheaded by Microsoft, IBM, and Ariba. UDDI is an XML-based registry for businesses worldwide, which enables businesses to list themselves and their services on the Internet. WSDL is the language used to do this. WSDL is derived from Microsoft's Simple Object Access Protocol (SOAP) and IBM's Network Accessible Service Specification Language (NASSL). WSDL replaces both NASSL and SOAP as the means of expressing business services in the UDDI registry.[7]

3.3 J2EE

The Java 2 Platform, Enterprise Edition (J2EE) is a set of coordinated specifications and practices that together enable solutions for developing, deploying, and managing multi-tier server-centric applications. Building on the Java 2 Platform, Standard Edition (J2SE), J2EE platform adds the capabilities necessary to provide a complete, stable, secure, and fast Java platform to the enterprise level. It provides many advantages by significantly reducing the cost and complexity of developing and deploying multi-tier solutions, resulting in services that can be rapidly deployed and easily enhanced.[8]

4. EXPERIMENTAL RESULTS

Five communication profiles are specified in IEC 61400-25-4 and one of the profiles is SOAP, which use http based on TCP/IP. We used SOAP to implement web services environment and the virtual communication is carried out. Web services are realized under J2EE condition.

Fig. 6 explains web service procedure used in this paper. First, the folders are generated and Java bean is prepared. The Java bean made by a rule enables to inquire and change a value of attribute. Using Java bean, we can monitor and control the wind power plants. Fig. 7 shows the block diagram of Java bean method used in this study. The typical Java bean has its own get/set methods for each field. Table2 shows a part of get/set methods defined in IEC 61400-25-4

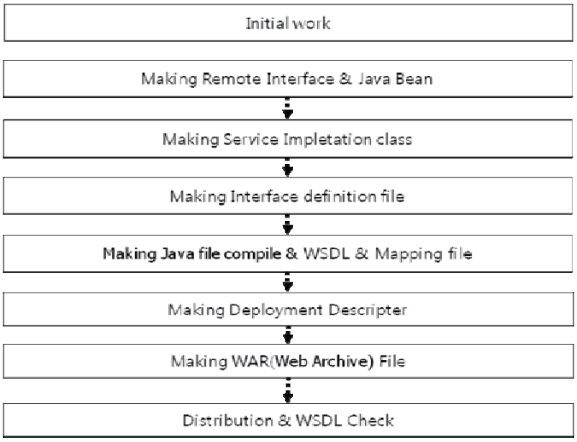

Fig. 6 Procedure of Web services.

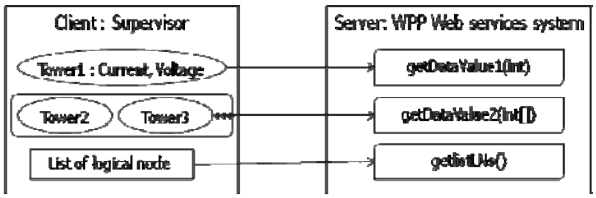

Fig. 7 . Java bean method

Table 2 get/set methods in IEC 61400-24-4

GetServerDirectory	
GetLogicalDeviceDirectory	
GetLogicalNodeDirectory	
GetDataValues	SetDataValues
GetDataDirectory	
GetDataDefinition	
GetDataSetValues	SetDataSetValues
GetBRCBValues	
GetLogStatusValues	

The get/set DataValue is used in our experiment. Service class to register to web server is made as Fig. 8, and registered class offers service to clients.

```
Package IEC61400;

import java.util.*;

public class IEC61400Impl implements IEC61400IF {
   private HashMap LNList;

   public IEC61400Impl() {
      LNList = new HashMap();
   }

   public String getDataValue1(int value) {
      LNList.put(value);
      String result = "T    T";
      return result;
   }
public String getDataValue2(int value[]) {
```

Fig. 8 Service class.

The method expressed by Java is compiled to WSDL(Web Services Description Language). The files for web services are deployed like tree in server as shown in Fig. 9.

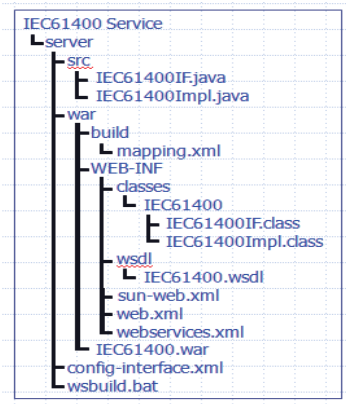

Fig. 9 Web file in server.

In experiment, we use local host in web server to offer web services. Web service for wind power plant is shown in Fig. 10. After client application software as a consumer of web services downloads WSDL at URL and analyzes it, it finally learns how to use web services system.

Fig. 10 Web services screen

SOAP message request for inquiring a current value of tower1 is shown in Fig.11. The current value of tower 1 is assigned the number '200'. As shown in Fig. 11, we can see the SOAP has unique structure such as body and envelope.

```
POST /IEC61400 HTTP/1.1
Host: 203.250.80.242
Content-Tye: text/xml charest=:utf-8"
Content-Lenght : nnn

<?xml version='1.0'?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

    <soap:Body xmlns:m="http://203.250.80.242/IEC61400-25"
        <m:GetTowerInfo>
            <m:Current>Tower1</m:Current>
        </m:GetTowerInfo>
    </soap:Body>
</soap:Envelop>
```

Fig. 11 Request of SOAP message.

After receiving the request of SOAP, web server sends response of SOAP message. We can find the number '200' is sent as tower 1 as shown in Fig. 12.

```
HTTP/1.1 200 OK
Content-Tye: text/xml charest=:utf-8"
Content-Lenght : nnn

<?xml version='1.0'?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

    <soap:Body xmlns:m="http://203.250.80.242/IEC61400-25"
        <m:GetTowerInfoResponse>
            <m:Current>200</m:Current>
        </m:GetTowerInfoResponse>
    </soap:Body>
</soap:Envelop>
```

Fig. 12 Response of SOAP message.

5. CONCLUSION

This paper implements web services to monitor and control the wind power plant by using SOAP according to communication protocol defined in IEC 61400-25. We expect that web services apply real wind power plants

ACKNOWLEDGMENT

The authors would like to thank the Ministry of Knowledge and Economy and Ulsan Metropolitan City which partly supported this research through the Network-based Automation Research Center (NARC) at the University of Ulsan.

REFERENCES

[1] World Wind Energy Association, WWEA, "World Wind Energy Report 2008", page 4

[2] *E. SAN TELMO, I. CANALES, J. L. VILLATE, E.*

ROBLES, S. APIÑANIZ, "THE USE OF IEC 61400-25 STANDARD TO INTEGRATE WIND POWER PLANTS INTO THE CONTROL OFPOWER SYSTEM STABILITY", European Wind Energy Conference & Exhibition. 7 – 10 May 2007, Milan., page. 2

[3] IEC, "wind turbine generator systems - Part 25: Communications for monitoring and control of wind power", WD IEC 61400-25, 2001

[4] IEC 61850"Communication networks and systems in substations. - Part 7-1: Basic communication structure for substation and feeder equipment – Principles and models"

[5] Umut Korkmaz, " Reporting and Logging in compliance with IEC 61400-25", Technical University of Denmark, page 13, 17

[6] World Wide Web Consortium (W3C), "http://www.w3.org/TR/soap12-part1/"

[7] http://www.bitpipe.com/rlist/term/WSDL.html

[8] Sun Microsystems Website, http://java.sun.com/javaee/overview/faq/j2ee.jsp