



Implementation of Data Communication System Between OPC Server and WCF Client using NetTcpBinding

Young-Kyu Oh¹, Jae-Woong Kim², Jeong-Bo Lee³

¹*Computer Science and Engineering, Kongju National University*

²*Computer Science and Engineering, Kongju National University*

³*Sayinfo Corporation*

ABSTRACT

Recent smart factories use OPC as their common communication mode to collect data from the manufacturing facilities. Data collected from OPC Server can be used for monitoring facilities, remote control, data sharing between facilities and equipment using OPC Client. However, the OPC client of OPC Classic is based on COM/DCOM, and has difficulty in firewall setup and security problems when communicating with remote clients. Additionally, the expansion of OPC server data to other platforms is required by the emergence of smart phones, but COM/DCOM is dependent on Windows platform. In this article, we have designed and implemented a remote data communication server/client using WCF as a way to demonstrate functions similar to the OPC UA communication system in the early OPC DA communication system. We have developed a system that can transfer collected data from PLC to remote WCF client, which data gets collected through OPC communication that is for automatic control and monitoring of industrial plant using WCF server. This study is the implementation of a model that can integrate and control PLC data from distributed production sites. This study introduces a model that can be applied to a system that has a limitation to introduce OPC UA, and it implements a method to collect and manage centralized PLC data of distributed production sites.

© 2018 KKITS All rights reserved

KEYWORDS: OPC server/client, WCF, Factory automation, MES, POP, Smart-factory

ARTICLE INFO: Received 6 November 2018, Revised 5 December 2018, Accepted 7 December 2018.

*Corresponding author is with the Computer Science & Engineering, Kongju National University, 1223-24 Cheonan-daero, Seobuk-gu, Cheonan-si, Chungcheongnam-do,

31080, KOREA

E-mail address: jwkim@kongju.ac.kr

1. 서론

미래 제조업의 성장 동력으로 스마트 팩토리, 4차산업혁명(Industry 4.0)의 중요성이 화두가 되면서 제조업 현장에서 생산되는 데이터의 수집, 분석, 모니터링이 강조되고 있다. 생산 설비에서의 데이터 수집 방법으로 OPC(OLE for Process Control)통신을 이용한 데이터 수집 방식이 현장에서 일반화되어 운영 되고 있다[1].

OPC통신은 원격지 클라이언트와 데이터 송수신시 RPC(Remote Procedure Call)통신의 보안 문제로 인해 OPC Foundation에서는 OPC UA(Unified Architecture)가 제안 되어 이러한 단점들이 일정부분 해결 되었다. 다만, OPC Classic기반에서 개발되어 운영 중인 기존시스템을 OPC UA로 변경하기에는 복잡도 증가, 시스템의 불안정을 야기 할 수 있다[2].

본 논문에서는 OPC Classic기반으로 만들어진 시스템에서 보안문제, 플랫폼 종속 문제를 해결하고 원격지 클라이언트들 사이의 데이터를 효율적으로 전달 할 수 있는 인터페이스 방법을 연구 개발 하고자 한다.

본 논문의 구성은 2장에서 OPC 와 WCF 관련 연구를 분석하며, 3장에서는 WCF를 이용해 원격지와 데이터 통신을 위한 서버/클라이언트 설계 및 기법을 제시한다. 4장에서는 설계에 따라 개발된 프로그램으로 데이터 측정 및 결과를 나타내고, 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 관련 연구

2.1 OPC(OLE for Process Control)

OPC는 산업자동화 태스크 포스에 의해 1996년 개발된 표준이다. 이러한 표준은 서로 다른 제조업체의 컨트롤 디바이스 사이에서 실시간 플랜트 데이터 통신을 명시하고 있다. 표준은 OPC Foundation에 의해 관리되며, OPC Data Access Standard로 명명하였다. Windows 기반 어플리케이션을 연결하고 컨트롤 하드웨어 및 소프트웨어 어플리케이션을 처리하기 위해 고안된 것으로 공정 설비로부터 데이터 접근을 위한 연속적인 방법을 허용하는 개방표준이다[3][4].

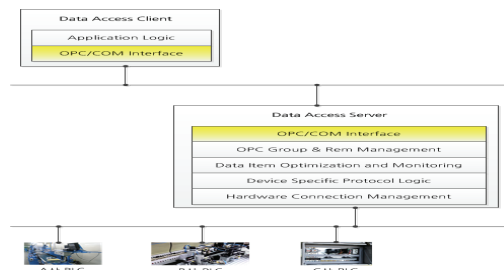


그림 1. OPC 데이터 액세스 구조
Figure 1. OPC Data Access Architecture

2.2 OPC DA(OPC Data Access) 와 OPC UA(OPC Unified Architecture)

OPC DA는 생산자동화 분야에서 필드버스와 상위 버스간의 호환을 위하여 마이크로소프트 COM/DCOM 기술을 기반으로 한 데이터 전달 인터페이스의 집합이다.

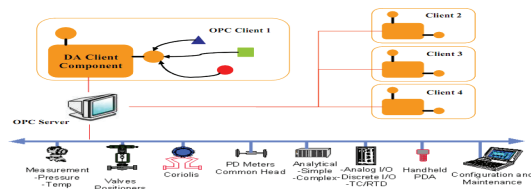


그림 2. OPC DA
Figure 2. OPC DA

OPC Classic은 COM과 DCOM을 사용하여 운영되므로 윈도우 플랫폼에 국한된다. 보안이 취약하고 플랫폼에 종속된다[5].

OPC UA는 정보 상호운용을 위해 개발된 M2M(machine to machine)통신 프로토콜이다[6]. OPC UA는 기존의 OPC Classic 즉 OPC DA, OPC A&E, OPC HDA[7]의 기능을 계승하고 있다 [8][9][10].

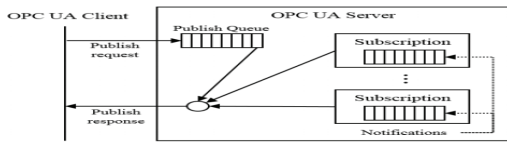


그림 3. OPC UA
Figure 3. OPC UA

2.3 WCF(Window Communication Foundation)

2.3.1 WCF의 정의

WCF는 서비스 지향 응용 프로그램을 만드는데 사용되는 프레임워크이다. 웹서비스 기반의 통신 프레임워크인 WCF를 사용하면 서로 통신하고자 하는 두 프로그램이 WCF를 이용하여 통신을 할 수 있다. <그림4>와 같이 어떤 기능을 제공하는 프로그램으로서 호출을 받는 피 호출자(callee)를 서비스(service)라 하고 기능을 제공 받기 위해 호출을 수행하는 호출자(caller)를 클라이언트라고 한다[11].

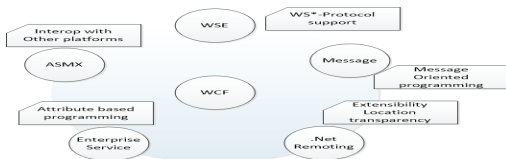


그림 4. WCF의 사용성
Figure 4. Usability of WCF

2.3.2 WCF의 특징

WCF는 서비스 지향 어플리케이션을 신속하게 개발하기 위한 통합된 서비스 프로그래밍이라고 정의할 수 있다. WCF의 특징은 다음과 같이 요약할 수 있다.

○ 보안성: 정보 보호를 위해 메시지를 암호화할 수 있으며, 사용자가 인증을 해야 메시지를 받을 수 있도록 지정할 수 있다.

○ 서비스 지향성: WS(Web Service)표준을 사용하여 WCF를 통해 서비스 지향 응용 프로그램을 만들 수 있다.

○ 상호 운용성: WCF는 웹서비스의 확장 스펙(WS-*)을 모두 지원하고 있다. 자바 플랫폼, MS의 분산 기술들과도 쉽게 서비스를 구현 할 수 있다 [12].

2.3.3 WCF의 요소기술

○ WCF의 역할

WCF는 SOAP메시지 기반의 웹 서비스를 XML 웹 서비스 요소로 바꾸어 준다.

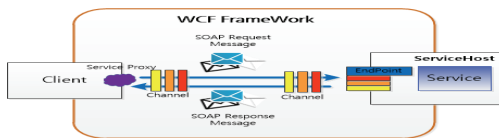


그림 5. WCF의 역할
Figure 5. Role of WCF

<그림5>에서와 같이 WCF서비스와 WCF클라이언트가 WCF런타임과 클래스 라이브러리를 이용하여 서비스와 클라이언트 프록시를 구성한다. 클라이언트가 프록시를 통해 서비스를 호출하면 서비스 호출은 여러 채널을 통과하면서 XML기반의

SOAP 메시지로 변환되고 이 메시지가 서비스에 도착하게 된다. 서비스 측의 채널들은 수신된 SOAP 메시지를 처리하여 다시 서비스 메서드 호출로 변환해 준다. 서비스 메서드 수행 후 서비스의 수행 결과는 다시 채널을 통과하면서 XML기반의 SOAP 결과 메시지로 변환된다. 클라이언트 측의 채널들은 결과 XML을 처리하여 프로키의 리턴값(return value)으로 변환하여 최종적으로 클라이언트에게 전달된다[13].

○ 바인딩(Binding)

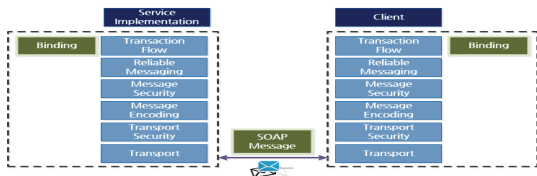


그림 6. 바인딩의 개념
Figure 6. Concept of Binding

WCF에서 바인딩은 <그림6>과 같이 서비스와 클라이언트가 네트워크를 통해 통신하기 위해 필요한 트랜스포트 종류, 프로토콜 설정, 보안과 같은 특정 기능의 사용 여부 등을 명세 하는 통신 설명자이다[13].

○ 다중 클라이언트 지원

WCF서비스가 공용인터넷을 통해 서비스를 제공할 수 있기 때문에 바인딩이 어떤 트랜스포트를 사용하는가에 따라 달라진다. 트랜스포트란 XML SOAP메시지를 네트워크 상에 전송 해주는 매개체 프로토콜을 말한다.

<표1>은 WCF에서 기본으로 제공하는 바인딩과 이들이 사용하는 트랜스포트를 보여준다. 이처럼 서비스 주소는 서비스의 종점이 사용하는 바인딩에 의존적으로 어떤 스킴(scheme)의 URI를 사용할

것인가 가 결정되며 그 주소 설정 방법이 결정된다[14].

표 1. WCF 트랜스포트
Table 1. WCF Transports Type

트랜스포트	바인딩	Scheme
HTTP(S)	BasicHttpBinding, WSHttpBinding, WSDualHttpBinding, WSFederationBinding	http혹은 https
TCP	NetTcpBinding	Net.tcp
Named Pipe	NetNamedPipeBinding	Net.pipe
MSMQ	NetMsmqBinding, MsmqIntegrationBinding	Net.msmq
P2P	NetPeerTcpBinding	Net.p2p

3. WCF를 이용한 원격지 데이터 통신 서버/클라이언트 설계

본 논문에서는 PLC(Programmable Logic Controller)와 연결된 OPC Server에서 실시간으로 수집 되는 PLC Tag Data를 WCF ServerService의 Behavior로 WCFRuntime을이용 하여 실행하고 WCF클라이언트 측에 OPC Server의 데이터를 보내도록 구성한다. WCF하부통신 시스템인 NetTcpBinding 으로 WCF서버 서비스를 시작하고, 원격지 WCF 클라이언트는 WCF서버 에서 정의된 정보를 이용하여 Endpoint를 만들어 Channel 및 바인딩 등으로 연결하도록 설계한다.

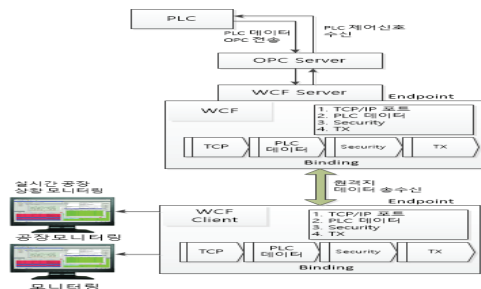


그림 7. 시스템의 구성도
Figure 7. System Configuration Diagram

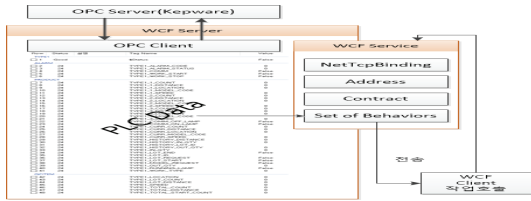


그림 8. WCF Server 구성
Figure 8. WCF Server Configuration

<그림7, 8>에서와 같이 PLC로 부터 수신 하는 OPC Server의 데이터를 WCF를 이용하여 원격지 클라이언트로 전송하는 구조이다. OPC Client를 통해 OPC Server의 데이터를 먼저 수집하고, OPC Client는 WCF서버와 연결을 설정 한다. 이러한 연결은 WCF에서 제공되는 절차에 따른다.

3.1 서비스 계약(Service Contract)

서비스 계약단계에서는 Endpoint가 WCF클라이언트에 연결을 Open하고 클라이언트가 호출할 수 있는 작업, 메시지 형식, 매개 변수, 데이터 형식, 클라이언트가 예상할 수 있는 처리 또는 응답 메시지 형식을 정의한다.

```
[ServiceContract(SessionMode = SessionMode.Allowed)]
public interface IServerService
{
    [OperationContract(IsOneWay = false)]
    List<PointEntity> DataGet();
    [OperationContract(IsOneWay = false)]
    void TagWrite(List<PointEntity> points);
}
```

그림 9. 서비스 계약 클래스
Figure 9. Service Contract Class

<그림9>과 같이 DataGet() 함수와 TagWrite()함수를 통해 서버와 클라이언트 간 데이터통신에 대한 서비스 계약을 맺는다.

3.2 바인딩 및 주소

<그림10>과 같이 NetTcpBinding을 사용하여 런

타임 통신 스택을 생성하며 전송 보안, 메시지 전달을 위한 TCP 및 이진 메시지 인코딩을 사용한다. 이러한 NetTcpBinding은 WCF보안기능을 사용하며 SOAP통신이 아닌 TCP로 직접 연결 된다.

```
var uri = new Uri(String.Format("net.tcp://{0}:{1}"
    "192.168.100.100", "4001"));
var netTcpSecurity = new NetTcpSecurity
{
    Mode = SecurityMode.None, Transport =
    { ClientCredentialType = TcpClientCredentialType.None } };
try { host = new ServiceHost(typeof(ServerService), uri);
    host.AddServiceEndpoint(typeof(IServerService),
    new NetTcpBinding { MaxReceivedMessageSize = int.MaxValue,
    Security = netTcpSecurity}, "ServerService");
    host.Open(); }
```

그림 10. 바인딩 및 주소
Figure 10. Binding and Address

3.3 ServerService

서비스 계약에 의해 정의된 DataGet(), TagWrite() 함수는 ServerService에서 실행에 대한 상세 내용이 작성된다. DataGet(), TagWrite() 함수는 WCF 클라이언트 요청에 따라 데이터를 Global로 정의된 PointEntity 클래스의 속성 값으로 전달하여 Service를 구성 한다.

```
[ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Multiple,
    InstanceContextMode = InstanceContextMode.Single)]
[CallbackBehavior(IncludeExceptionDetailInFaults = true)]
public static List<PointEntity> DataGet()
{
    try { return Global.DataEntity; }
}
public void TagWrite(List<PointEntity> points) {
    try{
        foreach (PointEntity point in points) {
            (Global.FormList(Global.CurrentWorkCenterForm) as
            FRM_BASE).TagWriteManual(point.TagName, point.TagValue);
        }
    }
}
```

그림 11. Service Behavior
Figure 11. Service Behavior

<그림11>과 같이 WCF ServerService에 대한 내용을 설계하였고, <그림12>에서는 PLC 데이터를 수신하는 PointEntity 클래스를 정의하였다.

```
public class PointEntity
{
    public PointEntity();
    public int Index { get; set; }
    public string Status { get; set; }
    public string TagName { get; set; }
    public string TagValue { get; set; }
}
```

그림 12. OPC Client 데이터 저장 클래스
Figure 12. OPC Client Data Storage Class

```

PointEntity point = new PointEntity();
point.Index = (Global._TagListViewItem.GetValue(ItemIndex) as
TagListViewItem).Index;
point.Status = (Global._TagListViewItem.GetValue(ItemIndex) as
TagListViewItem).ItemBox.Status.ToString();
point.TagName = (Global._TagListViewItem.GetValue(ItemIndex) as
TagListViewItem).TagShortName;
point.TagValue = (Global._TagListViewItem.GetValue(ItemIndex) as
TagListViewItem).ItemBox.Value.ToString();
Global.DataEntity.Add(point);
    
```

그림 13. OPC 데이터 전달
Figure 13. Forwarding OPC Data

<그림13>는 OPC Client에서 수집된 OPC Server 데이터를 WCF ServerService에 전달하여 WCF클라이언트에 전송 되도록 구성한다.

3.4 WCF 클라이언트 설계

WCF서버는 항상 WCF클라이언트의 요청을 기다리도록 구성하였으며 NetTcpBinding을 통해 이벤트 발생마다 연결이 이루어지면 서비스 계약에 따라 ServerService를 실행하도록 설계 하였다.

```

private readonly ChannelFactory<T> _serverServiceFactory
public T Service { get; private set; }
public ServerProxy(string uri = "") {
var netTcpSecurity = new NetTcpSecurity{
Mode = SecurityMode.None,
Transport = { ClientCredentialType = TcpClientCredentialType.None
} };
_serverServiceFactory = new ChannelFactory<T>(new
NetTcpBinding {
OpenTimeout = TimeSpan.FromSeconds(100),
SendTimeout = TimeSpan.FromSeconds(100),
MaxReceivedMessageSize = int.MaxValue,
Security = netTcpSecurity,
new EndpointAddress(uri.Length == 0 ?
Global.Variable.ServerAddress : uri));
Service = _serverServiceFactory.CreateChannel();
}
    
```

그림 14. WCF 구현
Figure 14. WCF Implementation

<그림14>에서는 WCF클라이언트가 WCF 서버에 접속하여 OPC Server가 보유한 PLC Data를 OPC Client의 Global 변수를 통해 PointEntity 클래스로 담아 WCF클라이언트에 전달되도록 구성하였다.

4. 실험 및 평가

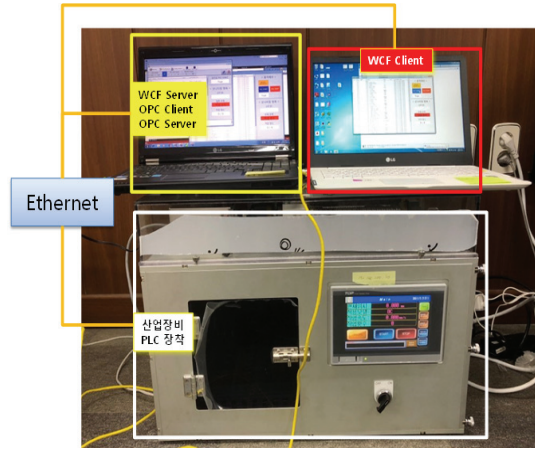


그림 15. 실험환경
Figure 15. Experimental environment

본 논문에서 제안된 방법을 검증하기 위해 3장에서 제시된 설계안에 따라 프로토타입을 개발 하였다. <그림15> 와 같이 생산현장에서 많이 사용하는 Mitsubishi사의 MelsecQ PLC, Takebishi사의 DeviceXPlorer를 OPC Server로 사용하여 시뮬레이션 환경을 구성 하였다.

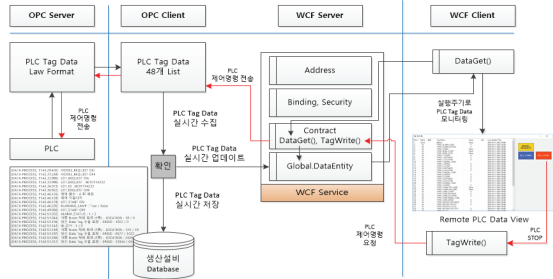


그림 16. 프로토타입 시퀀스 다이어그램
Figure 16. Prototype sequence Diagram

OPC Client, WCF서버/클라이언트 프로그램을 개발 하여 <그림16>의 시퀀스 다이어그램에 따른 데이터 생성, 이동 등에 대해 모니터링, 로그파일 등을 활용하여 비교, 검증 하는 순으로 실험 하였다.

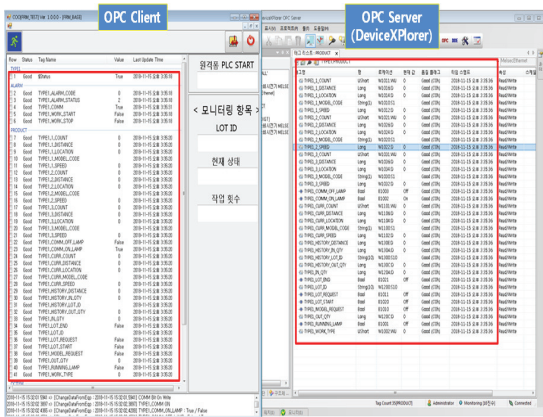


그림 17. OPC Server/Client 데이터 비교
Figure 17. OPC Server/Client Data comparison

OPC Server에 48개의 Tag를 등록하여 PLC에서 생성되는 데이터와 매핑 되게 설정 하였다. 실험 장비를 가동시켜 <그림17>과 같이 OPC Server와 OPC Client에 표시되는 Tag리스트 및 Value가 일치 하는지 육안으로 비교 하였으며, OPC Client에 시간별 Tag Value를 RawData형태의 Log파일로 저장하여 데이터 비교에 활용 하였다.

<그림18>은 OPC Server의 데이터를 OPC Client를 통해 만들어진 RawData이며, Tag의 이벤트 발생 시 마다 Tag별, 시간별로 Value가 저장되어 PLC와 OPC Server/Client 및 WCF클라이언트의 데이터 비교에 활용 하였다.

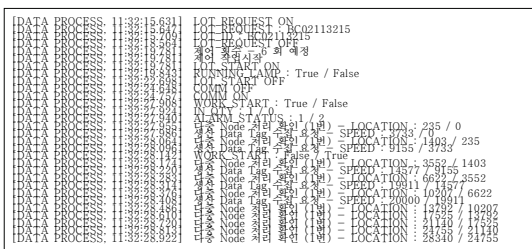


그림 18. PLC RawData 로그
Figure 18. PLC Rawdata Log File

<그림19>과 같이 OPC Client측의 WCF서버 데이터

와 원격지 WCF클라이언트 프로그램의 데이터를 모니터링 했다.

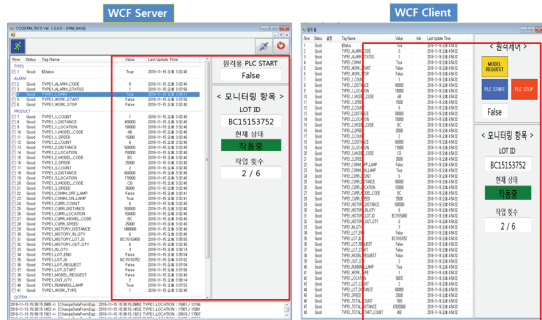


그림 19. WCF Server와 Client간 데이터 비교
Figure 19. Comparing Data between WCF Server and Client

원격지에서 PLC 제어의 검증은 WCF 클라이언트에 PLC START, PLC STOP의 커맨드를 통해 WCF서버의 TagWrite0함수를 호출, OPC Server측에 값이 정상적으로 변경 되는지를 실험 하였다.

WCF서버와 클라이언트간의 데이터 검증은 대표가 되는 Tag의 Value를 프로그램에 표시 하여 서버 데이터 변경 시 클라이언트측 데이터도 변경되는지 육안으로 비교 하였으며, 또한 프로그램 동작에 대한 동영상 촬영을 통해 상세 데이터를 비교 하는 방식으로 재차 검증 하였다. WCF클라이언트에서 WCF서버로의 데이터 요청 시간을 0.5초로 설정 하여 실험을 진행 하였다. 네트워크 환경 등의 이슈로 인해 약간의 지연시간이 발생하였으나, 원격지에서의 데이터 수신은 정확 하였다.

이러한 실험은 OPC Server에서 OPC Client와 WCF서버를 통해 원격지 WCF클라이언트 간 데이터 통신이 정확하게 이루어짐을 증명 하였다.

5. 결론

본 논문에서는 산업플랜트의 자동제어 및 감시에 사용하는 PLC로부터 OPC통신을 통해 수집 한

데이터를 WCF를 통해 원격지 클라이언트와 데이터 통신을 효율적으로 할 수 있는 시스템을 연구 개발 하였다. OPC통신 시스템의 초기 버전인 OPC Classic이 가지고 있던 보안문제, 플랫폼 종속 문제를 해결 할 수 있음을 보여 주었다. OPC UA를 도입하기 어려운 시스템에 OPC UA와 유사한 기능 및 성능을 발휘 할 수 있음을 증명 하였다.

향후 연구 과제로는 원격지 WCF클라이언트에 네트워크로 데이터 전달 시 데이터 전송 효율을 향상시키는 방안이 필요하다.

References

[1] OPC Foundation, *OPC Unified architecture : Interoperability for industrie 4.0 and the internet of things*, <http://www.automationworld.com/industry-40-and-opc-ua>, 2015.

[2] M. J. Lee, *Implementation and evaluation of standalone OPC UA wrapper in embedded system*, Department of Mechanical and Information Engineering The Graduate School University of Seoul, Dec. 2016.

[3] <https://opcfoundation.org/about/opc-technologies/opc-ua/>, Nov. 2008.

[4] <http://itcofe.web.cern.ch/itcofe/services/OPC/generalinformation/Specifications/RelatedDocuments/DASummary/DataAccessOvw.html>, Nov. 1998.

[5] W. Mahnke, S-H. Leitner, and M. Damm, *OPC Unified architecture*, Springer Science & Business Media, 2009.

[6] OPC Foundation, *OPC unified architecture specification*, Parts 1-11, available at www.opcfoundation.org.

[7] OPC Foundation, *OPC data access custom interface standard*, Version 2.04, Sep. 2000.

[8] T. Sauter, and M. Lobashov, *How to access*

factory floor information using internet technologies and wrappers, IEEE- Transactions on Industrial Informatics, Vol. 7, No. 4, pp. 699-712, Nov. 2011.

[9] Mai Son, *Toward an effective OPC-UA discovery mechanism*, School of Computer Engineering and Information Technology University of ULSAN, Nov. 2011.

[10] J. E. Jeong, *OPC UA specification analysis report*. KETI, KETI-TR02-02, V1.2, May 2017.

[11] <http://blog.naver.com/PostView.nhn?blogId=lse805&logNo=50041890988>, Jan. 2009.

[12] hoons.NET, WCF, <http://www.hoons.net>, Jan. 2007.

[13] G. S. Yu, *WCF bible set*, Hanbichmedia, Jan. 2010.

[14] MSDN, <http://msdn.microsoft.com>, Mar. 2017.

[15] V. Kapsalis, K. Charatsis, A. P. Kalogeras, and G. Papadopoulos, *Web gateway: A platform for industry services over Internet*, *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, Vol. 1, pp. 73-77, Jul. 2002.

NetTcpBinding을 이용한 OPC Server와 WCF Client간 데이터 통신 시스템 구현

오영규¹, 김재용², 김달중³

¹ 공주대학교 컴퓨터공학과 석사과정

² 공주대학교 컴퓨터공학과 교수

³ (주)세이정보기술 이사

요 약

최근의 스마트 팩토리는 제조현장의 생산설비로부터 발생하는 데이터를 수집하기 위한 방식으로 OPC 통신 방식을 일반적으로 사용 한다. OPC Server에서

수집된 데이터는 OPC Client를 이용하여 설비의 모니터링, 원격제어, 설비와 설비 간 데이터 공유 등의 기능으로 활용 할 수 있다. 하지만, OPC Classic의 OPC Client는 COM/DCOM 기반이어서 원격지 클라이언트들과 통신 시 방화벽 Setup의 난점[15], 보안문제 등을 안고 있다. 더불어, 스마트폰의 등장으로 OPC Server의 데이터를 타 플랫폼으로 공유하는 확장 기능도 요구되고 있으나, COM/DCOM은 윈도우 플랫폼에 종속적이다. 본 논문에서는 초창기 OPC Classic의 OPC Client 통신 방식으로 구성되어 있는 시스템을 OPC UA통신 방식과 유사한 기능을 발휘 할 수 있는 방안으로 WCF를 이용한 원격지 데이터 통신 서버/클라이언트를 설계하여 실험 하였다. 산업플랜트의 자동제어 및 감시에 사용하는 PLC로부터 OPC통신을 통해 수집한 데이터를 WCF서버와 동기화 시켜 원격지 WCF클라이언트로 전송하는 시스템을 연구 개발 하였다. 이러한 연구는 OPC UA를 도입하기에 한계성이 있는 시스템에 적용 할 수 있는 모델로 분산된 생산현장의 PLC데이터를 수집하여 중앙에서 통합 관리 할 수 있는 방안을 구현 한 것이다.

has been a professor in the Department of Computer Engineering at Kongju National University since 1992. His current research interests include software engineering. He is a life member of the KKITS.

E-mail address: jykim@kongju.ac.kr



Dal-Joong Kim received the bachelor's degree and the M.S. degree in the Department of Computer Engineering from the Jungang University in 1997 and 1999,

respectively. He received the Ph.D degree in the Department of Computer Engineering from the Daejoen University in 2018. He is working in Sayinfo Company. His current research interests include MES, S/W Engineering.

E-mail address: djkim@sayinfo.co.kr



Young-Kyu Oh received the bachelor's degree in the Department of Computer Engineering from the Catholic Kwandong University in 2000. He is

working in IfonSystem Company. His current research interests include MES, POP, OPC S/W Engineering. He is a member of the KKITS.

E-mail address: youngice@hanamil.net



Jae-Woong Kim received the bachelor's degree and the M.S. degree in the Department of Computer Engineering from the Jungang University in 1983

and 1988, respectively. He received the Ph.D. degree in the Department of Computer Engineering from Daejun University in 2000. He